Лабораторная работа №4. Модель гармонических колебаний

c/б 1032186063 | НФИбд-01-18

Доборщук Владимир Владимирович

6 марта 2021

Содержание

# Цели и задачи

**Цель:** изучить модель гармонических колебаний и программно реализовать процесс моделирования гармонического осциллятора.

**Задачи:**

* изучить теорию о модели гармонических колебаний
* построить модели гармонического осцилятора (фазовый портрет и его решение) для 3 случаев:
  + без затуханий, без воздейтвия внешних сил
  + с затуханиями, без воздействия внешних сил
  + с затуханиями, с воздействием внешних сил

# Теоретическая справка

Уравнение свободных колебаний гармонического осциллятора имеет следующий вид:

где – переменная, описывающая состояние системы (смещение грузика, заряд конденсатора и т.д.), – параметр, характеризующий потери энергии (трение в механической системе, сопротивление в контуре), – собственная частота колебаний, – время. (Обозначения )

Для решения поставленной нами задачи мы будем использовать именно эту форму уравнения, предварительно перейдя к следующей системе:

где - функция воздействия внешних сил.

# Программная реализация

## Подготовка к моделировнию

Все данные соответствуют варианту 14 = .

**Инициализация библиотек**

import numpy as np  
import matplotlib.pyplot as plt  
from math import sin, cos  
from scipy.integrate import odeint

**Начальные данные и необходимые функции**

Для успешной реализации модели нам потребуется и начальные данные коэффициентов и . Помимо этого, объявим функции для наших систем дифференциальных уравнений (с и без воздействия внешних сил).

w = np.sqrt(6)  
g = 0.00  
  
def f0(t):  
 value = sin(0.00\*t)  
 return value  
  
def f1(t):  
 value = cos(3.50\*t)  
 return value  
  
def dx0(x,t):  
 dx1 = x[1]  
 dx2 = -w\*w\*x[0] - g\*x[1] + f0(t)  
 return [dx1, dx2]  
  
def dx1(x,t):  
 dx1 = x[1]  
 dx2 = -w\*w\*x[0] - g\*x[1] + f1(t)  
 return [dx1, dx2]

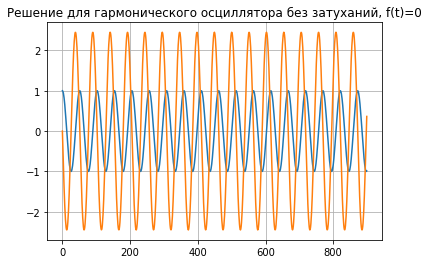
t0 = 0  
x0 = [1,0]  
t = np.arange(t0, 45, 0.05)

Также объявим функции для построения решения и фазового портрета гармонического осциллятора.

def plot\_solution(res, title):  
 plt.grid()  
 plt.title(title)  
 plt.plot(res)  
  
def plot\_portrait(res, title):  
 y1 = res[:,0]  
 y2 = res[:,1]  
   
 plt.grid()  
 plt.title(title)  
 plt.yticks(np.arange(-2,2,0.18))  
 plt.xticks(np.arange(-2,2,0.2))  
 plt.ylabel('y')  
 plt.xlabel('x')  
 plt.plot(y1, y2)

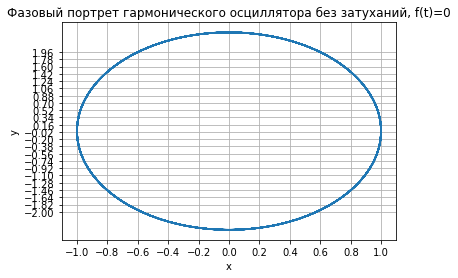
## Гармонический осциллятор без затухания, без воздействия внешних сил

x = odeint(dx0, x0, t)  
plot\_solution(x, 'Решение для гармонического осциллятора без затуханий, f(t)=0')



Решение для ГО без затуханий,

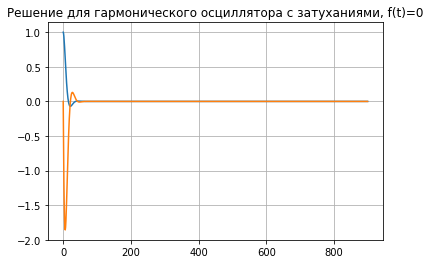
plot\_portrait(x, 'Фазовый портрет гармонического осциллятора без затуханий, f(t)=0')



Фазовый портрет для ГО без затуханий,

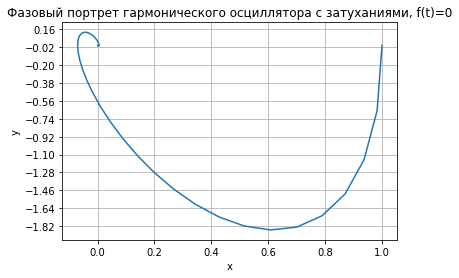
## Гармонический осциллятор с затуханиями, без воздействия внешних сил

w = np.sqrt(15)  
g = 5.00  
  
x = odeint(dx0, x0, t)  
plot\_solution(x, 'Решение для гармонического осциллятора с затуханиями, f(t)=0')



Решение для ГО с затуханиями,

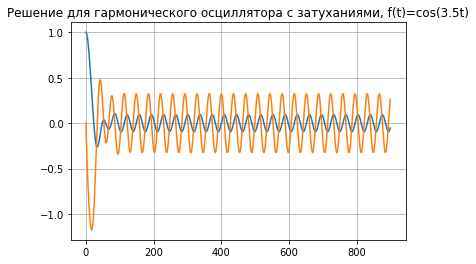
plot\_portrait(x, 'Фазовый портрет гармонического осциллятора с затуханиями, f(t)=0')



Фазовый портрет для ГО с затуханиями,

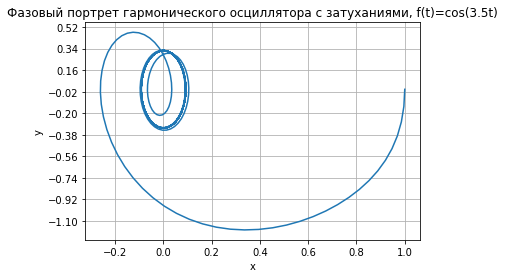
## Гармонический осциллятор с затуханиями, с воздействием внешних сил

w = 2  
g = 2.00  
  
x = odeint(dx1, x0, t)  
plot\_solution(x, 'Решение для гармонического осциллятора с затуханиями, f(t)=cos(3.5t)')



Решение для ГО с затуханиями,

plot\_portrait(x, 'Фазовый портрет гармонического осциллятора с затуханиями, f(t)=cos(3.5t)')



Фазовый портрет для ГО с затуханиями,

# Выводы

Мы изучили теорию о модели гармонических колебаний и программно реализовали процесс моделирования гармонического осциллятора, его фазового портрета и непосредственного решения. Все задачи можно считать выполненными успешно.