Лабораторная работа № 4

Ли Тимофей Александрович, НФИбд-01-18

Цель работы

Цель работы

- Изучить модель гармонический колебаний, построить фазовый портрет гармонического осциллятора и решение уравнения гармонического осциллятора для трех данных случаев.

Задачи

Задачи

- изучить теорию о модели гармонических колебаний
- реализовать программный код для 32 варианта

Ход работы

Описание решения

Уравнение свободных колебаний гармонического осциллятора имеет следующий вид:

$x'' + 2jx' + w_0^2 x = 0$ (1)

где x – переменная, описывающая состояние системы (смещение грузика, заряд конденсатора и т.д.), j – параметр, характеризующий потери энергии (трение в механической системе, сопротивление в контуре), w_0 – собственная частота колебаний, t – время. Уравнение (1) есть линейное однородное дифференциальное уравнение второго порядка, и оно является примером линейной динамической системы.

1. При отсутствии потерь в системе ($j = 0$) вместо уравнения (1) получаем уравнение консервативного осциллятора энергия колебания которого сохраняется во времени: $x'' + w_0^2 x =$ В моем варианте, уравнение выглядит следующим образом: $x'' + 5.2x = 0$, где $w_0^2 = 5.2$.
2. Во втором случае учитываются потери в системе, поэтому j = 14, в таком случае уравнение (1) принимает вид: $x'' + 14x' + 0.5x = 0$, где $w_0^2 = 0.5$.
3. Поскольку в третьем случае учитываются действия внешних сил, находящихся вне системы, то уравнение (1) приравнивается к функции $f(t) = 0.8sin(9t)$. Получим: $x'' + 13x' + 0.3x = 0.8\sin(9t)$, где $j = 13$, $w_0^2 = 0.3$.

Первый случай

```
In [60]: w=sqrt(5.2)
         g=0
         def f(t):
             f=sin(0.00*t)
             return f
         def dx(x,t):
             dx1=x[1]
             dx2=-w*w*x[0]-g*x[1]-f(t)
             return [dx1,dx2]

In [61]: t0=0
         t=arange(t0,59,0.05)
         x0=[0.5,-1.5]

In [62]: x=odeint(dx,x0,t)
         y=[[elem[0] for elem in x],[elem[1] for elem in x]]

In [63]: plt.grid()
         plt.title('Решение 1')
         plt.plot(t,x)

In [64]: def phaze(der, title): #функция получения и вывода фазового портрета
             x = odeint(der,x0,t)
             y1 = x[:,0]
             y2 = x[:,1]

             plt.grid()
             plt.title(title)
             plt.yticks(arange(-2,2,0.2))
             plt.xticks(arange(-2,2,0.2))
             plt.ylabel('y')
             plt.xlabel('x')
             plt.plot(y1, y2)

In [65]: phaze(dx, 'Фазовый портрет 1')
```

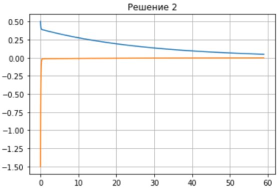
Рис. 2: Программный код для первого случая

Второй случай

```
In [66]: w=sqrt(0.5)
         g=14
         x=odeint(dx,x0,t)
         y=[[elem[0] for elem in x],[elem[1] for elem in x]]
```

```
In [67]: plt.grid()
         plt.title('Решение 2')
         plt.plot(t,x)
```

```
Out[67]: [<matplotlib.lines.Line2D at 0x62ced78>,
          <matplotlib.lines.Line2D at 0x62ceb50>]
```




```
In [68]: phaze(dx, 'Фазовый портрет 2')
```

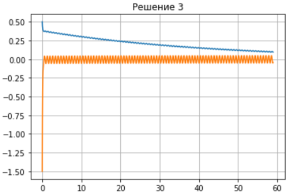
Рис. 3: Программный код для второго случая

Третий случай

```
In [69]: w=sqrt(0.3)
         g=13
         def f(t):
             f=0.8*sin(9*t)
             return f
         x=odeint(dx,x0,t)
         y=[[elem[0] for elem in x],[elem[1] for elem in x]]
```

```
In [70]: plt.grid()
         plt.title('Решение 3')
         plt.plot(t,x)
```

```
Out[70]: [<matplotlib.lines.Line2D at 0x1092bfe8>,
          <matplotlib.lines.Line2D at 0x1092b898>]
```




```
In [71]: phaze(dx, 'Фазовый портрет 3')
```

Рис. 4: Программный код для третьего случая

Результат

Решение и фазовый портрет для первого случая

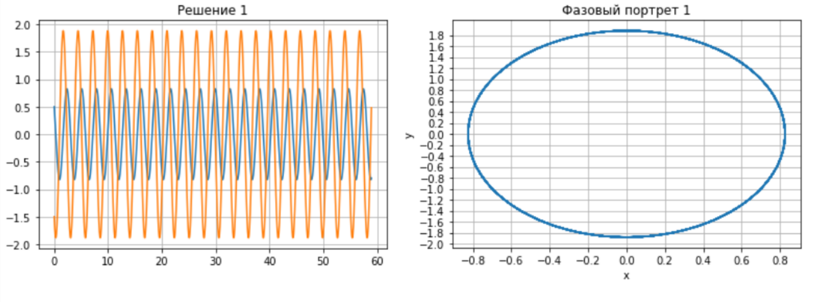


Рис. 5: Решение и фазовый портрет для первого случая

Решение и фазовый портрет для второго случая

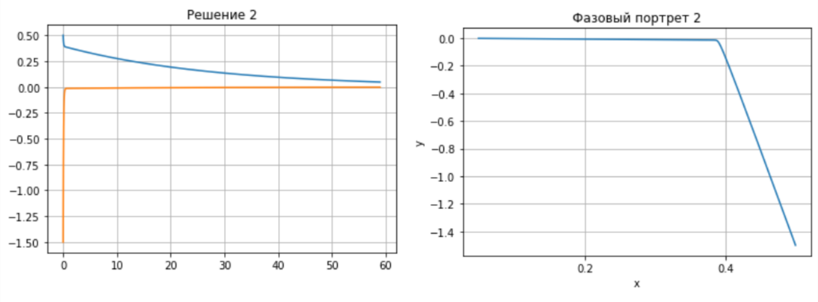


Рис. 6: Решение и фазовый портрет для второго случая

Решение и фазовый портрет для третьего случая

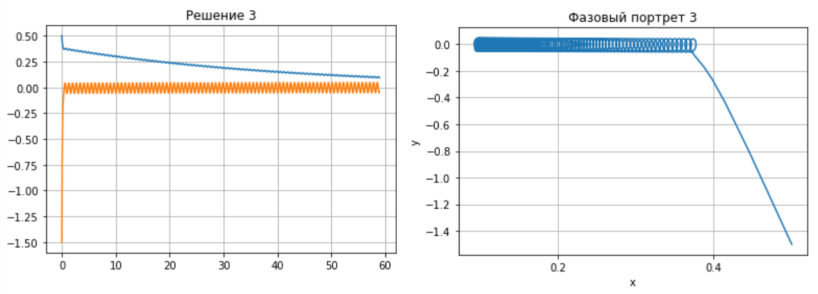


Рис. 7: Решение и фазовый портрет для третьего случая

Выводы

Выводы

- Изучил различные модели гармонических колебаний
- Реализовал программный код для поставленной задачи