

Лабораторная работа 5

Круглов Иван 1/2

1) Математическая модель

Колебательному контуру, содержащему индуктивность и емкость, можно сопоставить пружинный маятник. При этом заряд конденсатора аналогичен смещению груза, а ток в контуре - скорости движения груза. Упругая пружина является аналогом конденсатора, а движущаяся масса - аналогом катушки индуктивности.

Кроме катушки индуктивности и конденсатора в рассмотренной цепи имеется источник питания. Благодаря ему в цепи могут возникать колебания даже в том случае, когда в начальный момент и заряд конденсатора, и ток в цепи равны нулю.

Направим ось Ox вертикально вниз и будем отсчитывать смещение груза x от начального положения, в котором пружина не деформирована. Проекция силы, действующей на груз со стороны пружины, равна $-kx$.

На груз действует еще и сила тяжести.
Уравнение второго закона Ньютона имеет вид:

$$ma = mg - kx$$

Ускорение - это вторая производная смещения по времени по времени (x''). Введем обозначение $\omega_0^2 = k / m$ и перепишем уравнение в виде:
$$x'' + \omega_0^2 x = g$$

Процессы в механической и электрической системе описываются одинаковыми уравнениями. Одинаковыми будут и начальные условия: отсутствию заряда конденсатора и тока в начальный момент времени в электрической цепи соответствуют равные нулю смещение груза и его скорость в момент выдергивания подставки:

$$x(0) = 0, v(0) = 0$$

Таким образом, рассматриваемая механическая система действительно представляет собой аналог электрической цепи, и все сопоставляемые в них друг другу величины изменяются со временем по одинаковому закону. Поэтому смещение груза $x(t)$ дается формулой, в которую только вместо величины \mathcal{E} нужно подставить ее аналог в механической системе.

Следует заменить:

1) величину \mathcal{E} / L на g ,

2) величину $C\varepsilon = \varepsilon / L\omega_0^2$ на $g / \omega_0^2 = mg / k$.

Получим уравнение для механической системы

$$x(t) = mg / k (1 - \cos(\omega_0 t))$$

2) Код программы:

```
1. from numpy import*
2. from matplotlib.pyplot import*
3.
4. C = 0.1
5. L = 1
6. E = 10
7. g = 9.8
8. m = 0.3 # произвольная величина близкая к реальной
9. k = 0.15 # произвольная величина близкая к реальной
10.
11. w0 = sqrt(1 / L * C)
12.
13. def getX(t):
14.     return m * g / k * (1 - cos(w0 * t))
15.
16. T = 10**2
17.
18. xt = array([getX(i) for i in range(0, T)])
19. time = array([i for i in range(0, T)])
20.
21. figure()
22. title("Зависимость x(t)")
23. xlabel("t")
24. ylabel("x(t)")
25. plot(time, xt)
26. plot([-0.1 * T, 1.1 * T], [0, 0])
27. savefig("x(t) graphic.jpg")
```

3) Результат программы:

