УДК 621.31

В.А. Сытник, ассистент кафедры ОЭ (КузГТУ)
г. Кемерово

РАСЧЕТ ПЕРЕХОДНОГО ПРОЦЕССА С ПОМОЩЬЮ PYTHON

Процесс перехода от одного установившегося режима работы электрической цепи, при изменении значений параметров схемы или коммутации, к другому, отличающегося от предыдущего энергетическим состоянием называют переходным процессом.

Изучение переходных процессов позволяет оценить превышения токов или напряжений на отдельных участках цепи, устойчивость электромеханической системы.

В данной статье будет рассмотрен электромагнитный переходный процесс на примере электрической цепи, изображенной на рис. 1.

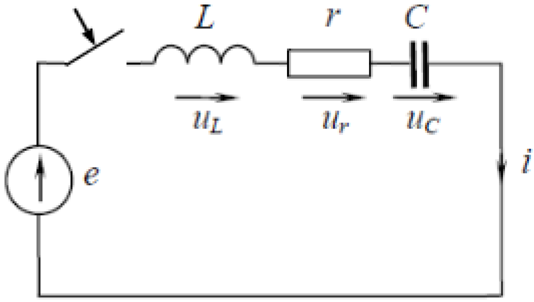


Рис. 1. Электрическая цепь до замыкания ключа

Параметры элементов электрической цепи на рис. 1.: *e(t)*=*E*=200 *В*, *r*=40 Ом, *L*=0,05 Гн, *C*=10^{-5} Ф, конденсатор заряжен до 100 В.

Независимые начальные условия $i_L(0_+)$ *и* $u_C(0_+)$ определим из режима до коммутации – ключ разомкнут, цепь отключена от источника энергии. Так как цепь разомкнута ток через катушку индуктивности не протекал $\mathrm{i_L}(0_-) = 0$ А. При условии, что конденсатор заряжен до 100 В имеем $\mathrm{u_C}(0_-) = 100$ В.

$$\mathrm{i_L}(0_+) = \mathrm{i_L}(0_-) = 0$$
$$u_C(0_+) = u_C(0_-) = 100$$

При замыкании ключа цепь можно представить следующим образом:

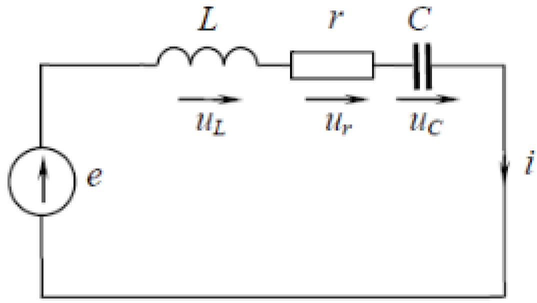


Рис. 2. Электрическая цепь после замыкания ключа.

Запишем уравнение электрического состояния – II закон Кирхгофа для цепи на рис. 2.:

$$u_L(t) + u_r(t) + u_C(t) = e(t) \qquad (1)$$

или в дифференциальной форме:

$$L\frac{di(t)}{dt} + ri(t) + \frac{1}{C}\int i(t)dt = e(t) \qquad (2)$$

Выразим ток в цепи через напряжение на конденсаторе $i = C\frac{du_C(t)}{dt}$:

$$LC\frac{d^2u_C(t)}{dt} + rC\frac{du_C(t)}{dt} + u_C(t) = e(t) \qquad (3)$$

Для численного решения данного уравнения воспользуемся функцией odeint() модуля scipy.integrate языка программирования Python.

Функцию odeint() можно импортировать из модуля scipy.integrate. Функция служит для поиска решений системы дифференциальных уравнений в одной точке, т.е. для решения задачи Коши.

Для одного дифференциального уравнения n-го порядка, задача Коши состоит в нахождении функции, удовлетворяющей равенству:

$$y^{(n)} = f\left(t, y, y', \dots, y^{(n-1)}\right) \qquad (4)$$

и начальным условиям

$$y(t_0) = y_1^0, y'(t_0) = y_2^0, \dots, y^{(n-1)}(t_0) = y_n^0 \qquad (5)$$

Для решения уравнение (4) должно быть переписано в виде системы дифференциальных уравнений (СДУ):

$$\begin{aligned} \frac{dy_1}{dt} &= f_1(y_1, y_2, \dots, y_n, t) \\ \frac{dy_1}{dt} &= f_2(y_1, y_2, \dots, y_n, t) \\ &\dots\dots. \\ \frac{dy_n}{dt} &= f_n(y_1, y_2, \dots, y_n, t) \end{aligned} \quad (6)$$

с начальными условиями

$$y_1(t_0) = y_1^0, y_2(t_0) = y_2^0, \dots, y_n(t_0) = y_n^0 \quad (7)$$

Функция odeint() имеет следующий формат odeint(f(y, t), y_0, t), где f(y, t), y_0, t – три обязательных аргумента.

Аргумент f – это имя функции двух переменных, первой из которых является записанная в виде списка $y = [y_1, y_2, \dots, y_n]$ система уравнений, которую необходимо рассчитать, а второй – имя независимой переменной, в данном случае времени *t*. Функция f возвращает список из n значений функций $f_i(y_1, \dots, y_n, t)$ при заданном значении t. Фактически функция f(y,t) реализует вычисление правых частей системы (6).

Аргумент y_0 функции odeint() является массивом (или списком) начальных значений $y_1^0, y_2^0, \dots, y_n^0$ при $t=t_0$.

Аргумент *t* является массивом моментов времени, в которые мы хотим получить решение системы дифференциальных уравнений. Первым элементом этого массива является t_0 [1].

Для решения выражения (3) при помощи функции odeint() запишем его в виде системы дифференциальных уравнений в форме Коши, для этого выполним замену $y_1 = u_c(t), y_2 = y_1' = \frac{du_c(t)}{dt}$:

$$\begin{aligned} \frac{dy_1}{dt} &= \frac{dy_1}{dt} \\ \frac{dy_2}{dt} &= \frac{E}{LC} - \frac{r}{L} y_2 - \frac{y_1}{LC} \end{aligned} \quad (8)$$

Ниже приведен текст программы для расчета переходного процесса в электрической цепи на рис. 1,2:

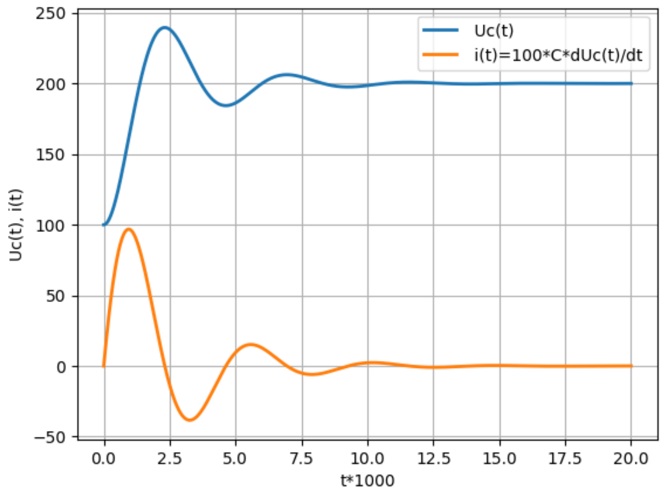
```
import numpy as np
from scipy.integrate import odeint
E=200; r=40; L=0.05; C=0.00001; #параметры электрической цепи
tm=0.02 # время моделирования
```

```
def f(y, t): #СДУ переходного процесса.
        y1,y2 = y
        return [y2, E/(L*C)-(r/L)*y2-(1/(L*C))*y1]
t = np.linspace(0,tm,1000)
#Возвращает равномерно распределенные числа через указанный интервал,
#т.е. в интервале от 0 до значения tm=0.02 разбивает на 1000 шагов

y0 = [100,0] #начальные условия
z = odeint(f, y0, t) #решения системы дифференциальных уравнений
y1=z[:,0] #вектор значений решения
y2=100*C*z[:,1] #вектор значений производной.
```

Обратите внимание, что в языке программирования Python отступы от левого края имеют значение. Графическое отображение результатов получено при помощи инструмента pyplot библиотеки matplotlib и в тексте программы не приводится.

Результаты моделирования в графическом виде представлены на рис. 3:



Напряжение на конденсаторе стремится к значению принужденной составляющей $u_{C\text{пр}}(t) = E$, а сила тока в цепи стремится к $i_{\text{пр}}(t) = 0$. Колебательный характер переходного процесса обусловлен обменом энергией между электрическим и магнитным полем.

Таким образом используя функцию odeint() можно найти решения системы дифференциальных уравнений в конкретных точках, в том числе рассчитать параметры переходного процесса электрической цепи в любой момент времени.

Список литературы:

1. Численное решение математических моделей объектов заданных системами дифференциальных уравнений // [Электронный ресурс]: Режим доступа: https://habr.com/ru/post/418139/ (дата обращения 17.10.2022).

Информация об авторах:
Сытник Владимир Анатольевич, ассистент кафедры ОЭ, КузГТУ, 650000, г. Кемерово, ул. Весенняя, д. 28, sytnikva@kuzstu.ru