

Министерство образования и науки
Российской Федерации

ТОМСКИЙ УНИВЕРСИТЕТ СИСТЕМ УПРАВЛЕНИЯ
И РАДИОЭЛЕКТРОНИКИ (ТУСУР)

КАФЕДРА ТЕЛЕКОММУНИКАЦИЙ
И ОСНОВ РАДИОТЕХНИКИ (ТОР)

УТВЕРЖДАЮ
Зав. кафедрой ТОР
_____ Е.П.Ворошилин

**Методические указания
к лабораторным работам по системе SciLab**

по дисциплине «Введение в специальность» (1 семестр)
специальности «Радиосвязь, радиовещание, телевидение» (код 210405)

РАЗРАБОТЧИК
_____ В.И.Белов
доцент каф. ТОР

СОДЕРЖАНИЕ

1. Введение.....	3
2. Лабораторная работа №1. Построение графиков функций.....	3
3. Лабораторная работа №2. Построение усложненных графиков функций.	4
4. Лабораторная работа №3. Построение графиков непериодических и периодических функций (сигналов).....	5
5. Лабораторная работа №4. Нахождение нулей многочленов.....	7
6. Литература.....	8
7. Приложение	8

ВВЕДЕНИЕ

Лабораторные работы по дисциплине «Введение в специальность» предназначены для начального освоения системы математических вычислений Scilab. Эта система похожа на систему MatLab, но в отличие от нее она является открытой и бесплатной. Пакет программ системы Scilab и ее расширения можно найти в Internet. Причем пакет постоянно обновляется и совершенствуется.

Имеются подробные руководства и книги на русском языке по системе Scilab [1], многие из которых можно найти также в сети Internet. Для овладения возможностями всего пакета при математических расчетах потребуется значительное количество времени. В то же время при решении задач в студенческих работах часто необходимо выполнять лишь простые вычисления и построения графиков. Быстро освоить приемы выполнения таких задач (от простых к более сложным) можно изучая примеры программ, которые приведены в каждой лабораторной работе. Примеры содержат подробные пояснения и соответствуют направлению индивидуальных работ по дисциплине «Радиотехнические цепи и сигналы».

Кроме этого желательно дополнительно знакомиться с соответствующими разделами книг по системе Scilab [1].

Результаты выполненных лабораторных работ оформить в виде отчета, содержащего коды программ, графики и пояснения к ним.

ЛАБОРАТОРНАЯ РАБОТА №1. Построение графиков функций.

Цель работы: научиться создавать программы построения графиков функций (от простых к более сложным).

Пример 1. Построить график косинусоиды $y = \cos x$, $0 \leq x \leq 2\pi$.

Ниже даны пояснения, комментарии (курсив) и коды программы, написанные шрифтом Courier New.

// – это символ комментария. Замечание: для разделения целого и десятичной дроби

//используется не запятая, а точка.

x=[0:0.1:2%pi]; //%pi – число π ; x определяется в виде массива точек равномерного*

//разбиения сегмента $[0;2%pi]$ с шагом 0.1 (или вектора с компонентами 0,0.1,0.2,...,6.2).*

//В конце команды ставим «;», иначе произойдет вывод в командном окне всех компонент

//вектора x

`plot(x, sin(x))` *//строим простой график (simple plot)*

Эту программу (две строки) можно набрать в командном окне и запустить на выполнение нажатием клавиши Enter. Появится графическое окно с графиком косинусоиды.

По другому, вначале можно создать специальное окно SciPad для набора, редактирования и запуска программ на исполнение. Для этого нужно войти в меню командного окна и нажать последовательно «Инструменты», «Редактор». После набора программы в окне SciPad ее можно запустить нажатием клавиш «Ctrl+I» или нажать последовательно Execute → Load into Scilab в меню SciPad. Это окно удобно использовать

для сохранения программы в виде отдельного файла с каким-либо именем и расширением .sce.

Пример 2. Построить графики функций $y = \cos x$ и $y = \sin 2x$ на одном рисунке.

```
x=[0:0.1:2*pi];
plot(x, cos(x), x, sin(2*x))//построение двух графиков в одном окне
```

Пример 3. Назначение правого положения оси y

```
x=[0:0.1:2*pi];
plot(x, sin(x))
a=gca(); //присвоение символу «a» роли оператора назначения
a.y_location = "right"; // назначение на графике правого положения оси y
Можно назначить среднее положение оси параметром "middle"
```

Индивидуальные задания по лабораторной работе №1.

1. Построить графики функции $y = y(x)$ при различных положениях оси y , если $y_1 = x^3$, $y_2 = 3x^2 + 1$, $y_3 = \arctg(x)$, $y_4 = \sin(x^2)$, $y_5 = \cos(2x + 3)$, $y_6 = \lg(4 - x)$, $-2 \leq x \leq 3$.

Вариант	1	2	3	4	5	6	7	8	9
y	$y_1 + y_2$	$y_1 + y_3$	$y_1 + y_4$	$y_1 + y_5$	$y_2 + y_3$	$y_2 + y_4$	$y_2 + y_5$	$y_3 + y_4$	$y_3 + y_5$

Вариант	10	11	12	13	14	15	16	17	18
y	$y_1 \cdot y_2$	$y_1 \cdot y_3$	$y_1 \cdot y_4$	$y_1 \cdot y_5$	$y_2 \cdot y_3$	$y_2 \cdot y_4$	$y_2 \cdot y_5$	$y_3 \cdot y_4$	$y_3 \cdot y_5$

Вариант	19	20	21	22	23	24	25	26	27
y	$y_4 + y_5$	$y_4 y_5$	$y_1 + y_6$	$y_2 + y_6$	$y_3 + y_6$	$y_4 + y_6$	$y_5 + y_6$	$y_1 \cdot y_6$	$y_2 \cdot y_6$

ЛАБОРАТОРНАЯ РАБОТА №2. Построение усложненных графиков функций.

Цель работы: научиться создавать программы построения более сложных графиков функций.

Пример 4. Формирование кривой графика функции определенного вида

```
x=1:10; //x представлен вектором с компонентами 1,2,3,...,10
plot(x, x.*x, '*суа--')
```

Здесь $*$ – знак поэлементного способа умножения x на x , в результате получаем вектор x^2 с компонентами 1, 4, 9, ..., 100;

'*суа--' – эта запись относится к области LineSpec и задает вид кривой графика. Первый символ записи * внутри кавычек характеризует тип маркера кривой (описание маркеров находится в соответствующей таблице ПРИЛОЖЕНИЯ). Запись «суа» определяет цвет кривой (цвет выбирается в соответствующей таблице ПРИЛОЖЕНИЯ 1), символ «--» задает тип линии.

Пример 5. Задание толщины всех линий на графике

```
x=[0:0.1:2*pi];
xset("thickness", 2); //установка толщины всех линий графика выбором числа
//(число 2 задает линии толщиной в 2 пикселя)
plot(x, sin(x))
```

Пример 6. Задание ширины и высоты графического окна

```
f4=scf(4); //функция scf создает графическое окно с номером 4 и делает
//его текущим
f4.figure_size = [600, 400]; //установка ширины и высоты графического окна
//с номером 4 в единицах экрана. Если монитор имеет разрешение 1024x768,
//то оно может быть максимальным размером графического окна.
plot() //вывод некоторых графиков по умолчанию
clf(f4); очистка графического окна с номером 4
dim=xget("wdim"); //вывод в командном окне размера текущего графического окна
```

Индивидуальные задания по лабораторной работе №2.

Построить графики трех функций в одном окне, выбирая для них оригинальные маркеры, цвета, типы и толщины линий. Функции брать из лабораторной работы №1. Причем, если $y = y_1 + y_2$, то y_1 – первая функция, y_2 – вторая функция, y – третья функция.

ЛАБОРАТОРНАЯ РАБОТА №3. Построение графиков непериодических и периодических функций (сигналов).

Цель работы: научиться создавать программы построения кусочно-заданных непериодических и периодических сигналов (функций).

Пример 7. Представление сигналов по условию первой задачи индивидуального задания по РТЦиС

```
E = 1; //амплитуда сигнала
Ts = 1; //промежуток времени наблюдения сигнала
//////////Представление данного сигнала в виде кусочно-заданной функции//////////
Ts=1; E=1;
function s = sz(t) //function – endfunction – оператор создания функции
    if ((t >=0) & (t <=2*Ts/3)) then s = -E; //условный оператор «если-то»
    elseif ((t >2*Ts/3) & (t <=Ts)) then s = E; //условный оператор «иначе-если-то»
    end
endfunction
////////// Вывод графика заданного сигнала//////////
t = [0: Ts/100: Ts];
clf(); //очистка графического окна с номером 0 (по умолчанию)
plot(t, sz); //построение графика заданного сигнала
zoom_rect([min(t), -1.1, max(t), 1.1]); //установка границ окна вдоль осей
//координат оператором zoom_rect[xmin, ymin, xmax, ymax]
xtitle('Grafic sz(t)', 't', 'sz(t)', boxed=1); //отображение названий графика, оси
//абсцисс, оси ординат; помещение названий в рамку только при boxed=1
xgrid(); //установка сетки в графическом окне
//////////Формирование периодического сигнала и вывод его графика//////////
t=0:Ts/100:3*Ts;
y = t-(floor(t/Ts))*Ts; //y - линейно изменяющаяся от 0 до Ts периодическая
//переменная с периодом Ts; floor() – операция выделения наибольшего целого,
//меньшего аргумента
for i = 1:length(y); //for-end – оператор цикла, length(y) – определяет
//количество компонент вектора y
    a=sz(y(i)); //дискретные значения периодически продолженного заданного сигнала
    sp(i)=a; //периодически продолженный заданный сигнал
end
scf(1); //функция scf создает графическое окно с номером 1 и делает его текущим
```

```

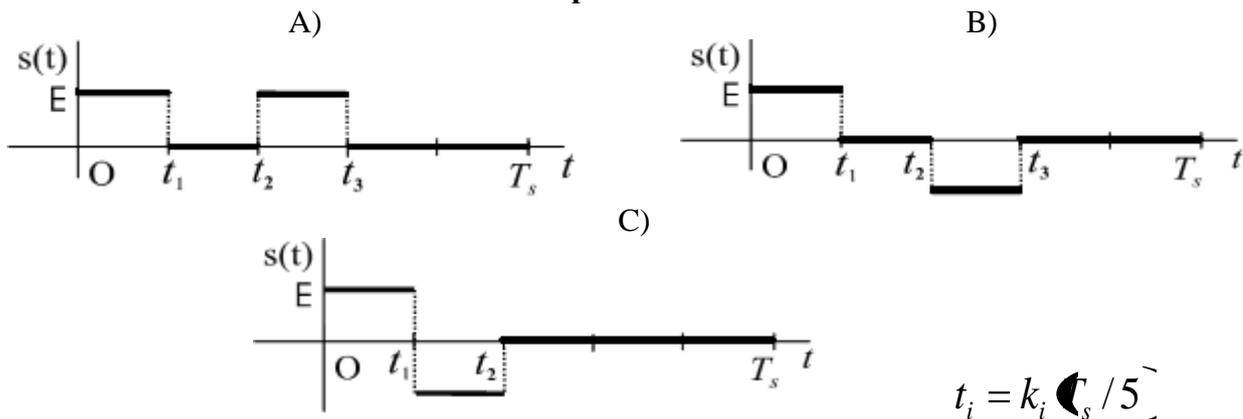
clf(1);
plot(t, sp); //вывод периодически продолженной функции (периодической функции)
zoom_rect([min(t), -1.1, max(t), 1.1]);
xgrid();
//////////Формирование неперидического сигнала и вывод его графика//////////
t=(-2*Ts:Ts/100:3*Ts);
function f = s0(t) //задание неперидической функции (заданной функции,
//продолженной нулями)
    if ((t < 0) | (t > Ts)) then f = 0;
        else f = sz(t);
    end
endfunction
scf(2);
clf(2);
plot(t, s0); //вывод неперидической функции
zoom_rect([min(t), -1.1, max(t), 1.1]);
xgrid();
//////////

```

Индивидуальные задания по лабораторной работе №3.

Построить графики заданной функции, ее неперидического и перидического продолжения

Формы сигналов



$$t_i = k_i \left(\frac{T_s}{5} \right)$$

k – формат сигнала (А, В или С), k_i – коэффициенты

Вариант	Шифр сигнала $k_1 k_2 k_3$	Вариант	Шифр сигнала $k_1 k_2 k_3$
1	A 1 2 3	16	B 1 4 5
2	A 1 2 4	17	B 2 3 4
3	A 1 2 5	18	B 2 3 5
4	A 1 3 4	19	B 2 4 5
5	A 1 3 5	20	B 3 4 5
6	A 1 4 5	21	C 1 2
7	A 2 3 4	22	C 1 3
8	A 2 3 5	23	C 1 4
9	A 2 4 5	24	C 1 5
10	A 3 4 5	25	C 2 3
11	B 1 2 3	26	C 2 4
12	B 1 2 4	27	C 2 5
13	B 1 2 5	28	C 3 4
14	B 1 3 4	29	C 3 5

ЛАБОРАТОРНАЯ РАБОТА №4. Нахождение нулей многочленов.

Цель работы: научиться создавать программы нахождения нулей (корней) многочленов.

Многочлен (или полином – много имен, степенной многочлен) $p(x) = a_0x^n + a_1x^{n-1} + \dots + a_{n-1}x + a_n$ имеет в точке $x = x_0$ нуль (корень), если $p(x_0) = 0$.

Основная теорема алгебры. Многочлен степени n имеет ровно n корней, действительных или комплексных.

Краткие сведения о комплексных числах (поясняются преподавателем).

Комплексное число $z = \overbrace{(x, y)}$,

$(x, 0) = x$ – действительное число,

$(0, y)$ – мнимое число,

$x = \operatorname{Re} z$ – действительная часть комплексного числа,

$y = \operatorname{Im} z$ – мнимая часть комплексного числа,

$\{z\} = \mathbb{R} + \{(x, y)\}$, $y \neq 0$ – множество комплексных чисел. Обозначают $C = \{z\}$.

$z_1 \pm z_2 = (x_1, y_1) \pm (x_2, y_2) = (x_1 \pm x_2, y_1 \pm y_2)$ – правило сложения-вычитания,

$z_1 \cdot z_2 = (x_1, y_1) \cdot (x_2, y_2) = (x_1x_2 - y_1y_2, x_1y_2 + x_2y_1)$ – правило умножения,

$j = (0; 1)$ – мнимая единица,

$j \cdot j = j^2 = (0, 1) (0, 1) = (-1, 0) = -1$, $j = +\sqrt{-1}$

Уравнение $x^2 = -1$ имеет решения $x_1 = -j$, $x_2 = j$.

$z = (x, y) = (x, 0) + (0, y) = (x, 0) + (0, 1)(y, 0) = x + jy$ – алгебраическая форма записи комплексного числа

При сложении и умножении комплексных чисел в такой форме записи, с ними можно обращаться как с обычными двучленами. Действительно,

$$z_1 + z_2 = (x_1 + jy_1) + (x_2 + jy_2) = (x_1 + x_2) + j(y_1 + y_2).$$

$$z_1 \cdot z_2 = (x_1 + jy_1)(x_2 + jy_2) = (x_1x_2 - y_1y_2) + j(x_1y_2 + x_2y_1).$$

Пример 8. Нахождение нулей многочлена.

`p = poly([5, 0, 4, 1], 'x', 'c') // задание полинома $x^3 + 4x^2 + 5 = 5 + 0 \cdot x + 4 \cdot x^2 + 1 \cdot x^3$`

`// В результате p становится именем полинома-символа. Таким образом здесь применяется`

`// символическое исчисление. Полином вызывается по его имени p.`

`// Значение полинома невозможно задать последовательностью чисел или вектором.`

`res=roots(p) // вычисление нулей полинома и вывод их на экран`

`x=-7:0.1:7;`

`y=5+4*x^2+x^3; // задание полинома в виде функции (в виде вектора)`

`plot(x, y) // построение графика полинома для контроля`

Индивидуальные задания по лабораторной работе №4

Найти нули полинома и построить его график для контроля:

$$\Omega \cdot x^4 + 2x^2 + x - \Omega,$$

где Ω – число, равное порядковому номеру студента в групповом журнале.

ЛИТЕРАТУРА

1. Алексеев Е. Р. Решение инженерных и математических задач / Е.Р. Алексеев, О.В.Чеснокова, Е. А.Рудченко. - М. : ALT Linux ; БИНОМ. Лаборатория знаний, 2008.- 260 с. : ил. ; 8 с. цв. вклейки.- (Библиотека ALT Linux).

ПРИЛОЖЕНИЕ

plot

plot (y, <LineStyle>, <GlobalProperty>)

plot (x, y, <LineStyle>, <GlobalProperty>)

plot (x1, y1, <LineStyle1>, x2, y2, <LineStyle2>, ... xN, yN, <LineStyleN>, <GlobalProperty1>, <GlobalProperty2>, .. <GlobalPropertyM>)

plot (<axes_handle>, ...)

x a real matrice or vector

y a real matrice or vector

<LineStyle>

LineStyle:

Specifier	Line Style
-	Solid line (default)
--	Dashed line
:	Dotted line
-. .	Dash-dotted line

Color:

Specifier	Color
r	Red
g	Green
b	Blue
c	Cyan
m	Magenta
y	Yellow
k	Black
w	White

Marker type:

Specifier	Marker Type
+	Plus sign
o	Circle
*	Asterisk
.	Point
x	Cross
'square' or 's'	Square
'diamond' or 'd'	Diamond
^	Upward-pointing triangle
>	Right-pointing triangle
<	Left-pointing triangle
'none'	No marker (default)