

УДК 519.6

ОСОБЕННОСТИ ГРАФИЧЕСКОГО ОФОРМЛЕНИЯ РЕЗУЛЬТАТОВ В СРЕДЕ ТЕХНИЧЕСКИХ РАСЧЕТОВ MATLAB

Рыболовлев А.А., к.т.н., доцент Академии ФСО России, г. Орел, e-mail: rybolovlev@rambler.ru;

Рыболовлев Д.А., к.т.н., научный сотрудник Академии ФСО России, г. Орел, e-mail: dmitrij-rybolovlev@yandex.ru

Ключевые слова: обработка результатов, график, диаграмма, MATLAB, единая система конструкторской документации.

Введение

На завершающем этапе любой исследовательской работы отдельное внимание уделяется представлению полученных результатов в той или иной форме. Существуют различные способы представления информации: вербальная форма (текст, речь), символическая (знаки, формулы), графическая (схемы, диаграммы, графики), предметно-образная (макеты, модели, filmy и др.).

Настоящая работа посвящена рассмотрению вопросов графического оформления результатов, полученных в среде технических расчетов MATLAB. Авторы убеждены, что наглядность (а значит и ясность, понятность, убедительность [1]) представляемых графиков и диаграмм¹ определяется не только владением элементарными способами обработки графического материала, но и умением исследователя грамотно использовать возможности среды MATLAB.

Отдельное внимание в статье уделяется вопросу соблюдения правил выполнения графиков, которые определены в рекомендациях единой системы конструкторской документации (ЕСКД) Р 50-77-88 [3]. Поскольку в других системах подобные сведения не обнаружены, в общем случае при построении графиков целесообразно руководствоваться правилами ЕСКД. Следует подчеркнуть, что при оформлении отчетов о научно-исследовательских работах (согласно ГОСТ 7.32-2001 [4] и ГОСТ 2.105-95 [5]), диссертаций и авторефератов диссертаций (согласно ГОСТ 7.0.11-2011 [6]) графики выполняются также в соответствии с требованиями стандартов ЕСКД.

При рассмотрении указанных вопросов практический интерес представляют замечательная книга Дж. Сквайрса [7] и пособие [8], которые также помогут исследователю в оформлении иллюстративного материала.

Исходные данные

Все примеры подготовлены в среде технических расчетов MATLAB R2012a. Рассматривается построение

Рассмотрены особенности графического оформления результатов, полученных в среде MATLAB, на примере построения графика нескольких функций. Отдельное внимание уделено вопросам наглядности построения и соответствия требованиям единой системы конструкторской документации.

графиков в прямоугольной системе двух координат (особенности выполнения диаграмм в других системах координат подробно описаны в ЕСКД).

В качестве исходных данных для построения кривых используются отсчеты базисных функций (БФ, несущих колебаний) системы связи, основанной на применении технологии VDSL (Very high speed Digital Subscriber Line, сверхскоростная цифровая абонентская линия) [9]. Исходный код скрипта, реализующего построение графика, с подробными комментариями приводится ниже:

```
function result = figure_def()
    n = 4; % количество кривых
    n_samples = 150; % количество опорных точек кривой
    norm_t = 1:n_samples; % нормированное время

    % В качестве примера кривых используются базисные функции (несущие колебания)
    % системы связи по цифровым абонентским линиям (технология VDSL)
    T = 2.3e-4; % период вспомогательного колебания в секундах
    dtDSL = 1.15e-6; % длительность тактового интервала в секундах
    dT = T/n_samples;
    data = zeros(n,n_samples); % инициализация матрицы данных, заполнение нулями

    % Цикл заполнения матрицы данных
    for i = 1:n
        freq = 4312.5*i; % изменение частоты очередного несущего колебания
        data(i,:) =
            cos(2*pi*freq*dT*norm_t).*sqrt(2/T).*(1/sqrt(T)).*sinc(norm_t.*dT./T);
        data(i,:) = data(i,:) ./
            norm(data(i,:)); % нормировка значений
    end

    for i = 1:n
        plot(data(i,:)); % построение очередной кривой
        if i == 1 hold on; end; % режим "наложения" на одном графике
    end;
    hold off;
end
```

¹ Далее термины «график» и «диаграмма» используются как синонимы и обозначают геометрическое изображение функциональной зависимости при помощи линии на плоскости [2].

Построение графика функцией `plot()` по умолчанию приводит к результату, представленному на рис. 1:

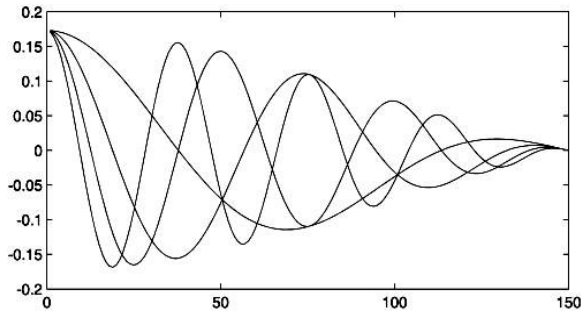


Рис. 1. Построение графика функцией `plot()` по умолчанию

Настройка параметров оформления графиков

Очевидно, что в случаях, когда в одной общей диаграмме изображают две или более функциональные зависимости, их необходимо каким-либо образом различать. Для этого достаточно передать в функцию `plot()` дополнительные параметры, определяющие толщину, тип и цвет линии, а также тип и размер маркеров (характерных точек кривой):

```
markerStyle = {'.'}; % тип маркера
markerSize = {1}; % размер маркера
lineColor = {'k'}; % цвет линии (k -
черный)
lineStyle = {'-', '--', ':', '-.'}; %
тип линии
lineWidth = {1 1 2 1}; % толщина линии

for i = 1:n
    plot(0:n_samples-1, data(i,:), ...
        markerStyle(mod(i-
1,length(markerStyle))+1), ...
        'MarkerSize', markerSize(mod(i-
1,length(markerSize))+1), ...
        'Color', lineColor(mod(i-
1,length(lineColor))+1), ...
        'LineStyle', lineStyle(mod(i-
1,length(lineStyle))+1), ...
        'LineWidth', lineWidth(mod(i-
1,length(lineWidth))+1));
    if i == 1 hold on; end;
end;
hold off;
```

В приведенном исходном коде обращение к элементам структур данных `cell array` производится с помощью «циклической» индексации – с использованием функции `mod()`, реализующей операцию взятия остатка от деления. Таким образом, количество выводимых на график кривых не ограничивается размерностью структур `markerStyle`, `markerSize`, `lineColor`, `lineStyle` и `lineWidth`.

Для повышения наглядности также целесообразно добавить линии координатной сетки и вывести пояснительную часть диаграммы:

```
grid(gca); % добавление координатной
сетки
```

```
% Явное указание шрифта позволяет избе-
жать проблем с кириллическими
% символами в надписях
```

```
set(gca, 'FontName', 'Times New Roman
Cyr');
set(gca, 'FontSize', 9); % размер шрифта
labels = {};
for i = 1:n
    labels = cat(1,labels,['БФ '
num2str(i)]); % БФ - базисная функция
end
legend(labels, 1);
```

Внесенные в код изменения приводят к результату, представленному на рис. 2. Согласно требованиям ЕСКД допускается изображать две или более функциональные зависимости в одной диаграмме линиями различной толщины и различных типов, если этим обеспечивается удобство пользования диаграммой (пункт 3.6, здесь и далее указание *пункта* относится к рекомендациям Р 50-77-88 [3]). Однако полученный график не позволяет четко различить кривые из-за особенностей экспорта фигур среды MATLAB в высоком разрешении (300 точек на дюйм и выше, как это требуется обычно при подготовке к печати): отличия пунктирной, штриховой и штрихпунктирной линий просматриваются только при увеличенном масштабе.

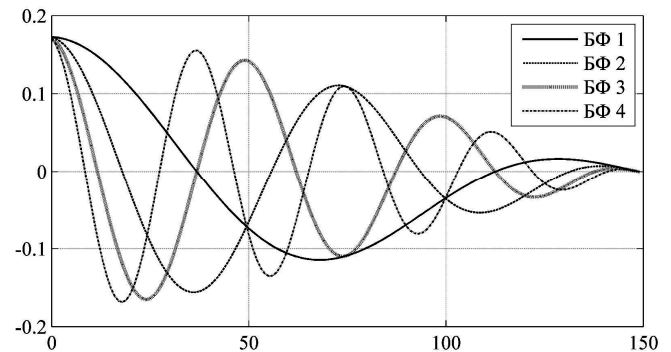


Рис. 2. Построение графика функцией `plot()` с дополнительными параметрами линий

Поскольку цветовое кодирование кривых не всегда подходит из-за ограничений цветности печатающих устройств, наиболее простым выходом из этой ситуации может служить использование различных маркеров для обозначения узловых точек кривых. Согласно пункту 3.10 рекомендаций ЕСКД точки диаграммы, полученные путем расчетов, допускается обозначать графически, например, кружком, крестиком и т.п. Обозначения точек при этом должны быть разъяснены в пояснительной части диаграммы.

Для использования различных маркеров необходимо внести следующие изменения в исходный код:

```
markerStyle = {'.', '.', 'o', '^', 's',
'*'}; % тип маркера
markerSize = {1 8 5 5 8 5}; % размер
маркера
lineColor = {'k'}; % цвет линии (k -
черный)
lineStyle = {'-'}; % тип линии
lineWidth = {0.5}; % толщина линии
```

Соответствующий график представлен на рис. 3.

Также необходимо обозначить на графике переменные величины и единицы их измерения. Требования

ЕСКД предписывают обозначать переменные величины одним из следующих способов (пункт 4.1):

- символом;
- наименованием;
- наименованием и символом;
- математическим выражением функциональной зависимости.

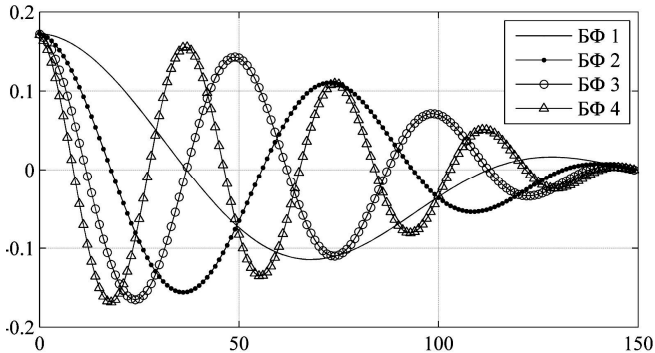


Рис. 3. Построение графика функций `plot()` с дополнительными параметрами маркеров опорных точек

Единицы измерения следует наносить одним из следующих способов (пункт 5.1):

- в конце шкалы между последним и предпоследними числами шкалы (при недостатке места допускается не наносить предпоследнее число);
- вместе с наименованием переменной величины после запятой;
- в конце шкалы после последнего числа вместе с обозначением переменной величины в виде дроби, в числителе которой – обозначение переменной величины, а в знаменателе – обозначение единицы измерения.

Для обозначения переменной величины в примере используются способ «символом» и «наименованием», а единицы измерения наносятся вместе с наименованием переменной величины после запятой.

Для обозначения переменной величины в примере используются способ «символом» и «наименованием», а единицы измерения наносятся вместе с наименованием переменной величины после запятой.

Кроме того, поскольку пункт 1.3 рекомендаций требует, чтобы в диаграммах со шкалами оси координат заканчивались стрелками за пределами шкал или обозначались самостоятельными стрелками после обозначения величины параллельно оси координат, после указания единицы измерения следует добавить стрелку. Форма стрелки и примерное соотношение её элементов приведены в ГОСТ 2.307-2011 [10], указанные требования носят рекомендательный характер, в связи с чем допускается для отображения стрелок использовать предопределенный символ "rightarrow" среды MATLAB.

Соответствующие изменения в исходном коде приведены ниже, а результаты построения графика представлены на рис. 4:

```
% Подписи к осям, название графика
xlabel('t, \muc \rightarrow');
ylabel('Нормированное значение \rightarrow');
title('Временное представление базисных функций системы связи VDSL');
```

```
xTickStep = 0.2e-6; % шаг координатной сетки по горизонтали = 0.2 мкс
len = length(data(i,:)); % количество
```

```
откладываемых отсчетов по оси абсцисс
xTickLabels = {'0'}; % подписи к делительным штрихам
for i = 1:(dtDSL/xTickStep)
    xTickLabels =
cat(1,xTickLabels,[num2str(i*xTickStep*1e6,3)
]);
end
xTickLabels =
cat(1,xTickLabels,[num2str(dtDSL*1e6,3)]); % подпись последнего штриха
set(gca, 'XTick',
[1:round(xTickStep*len/dtDSL):len len-1],
'XTickLabel', xTickLabels, 'Xlim', [1 len-1]); % изменение параметров
```

Приведенный исходный код также осуществляет вывод подписей делительных штрихов горизонтальной оси координат в микросекундах (а не в номерах отсчетов по умолчанию) – в единицах измерения переменной величины времени.

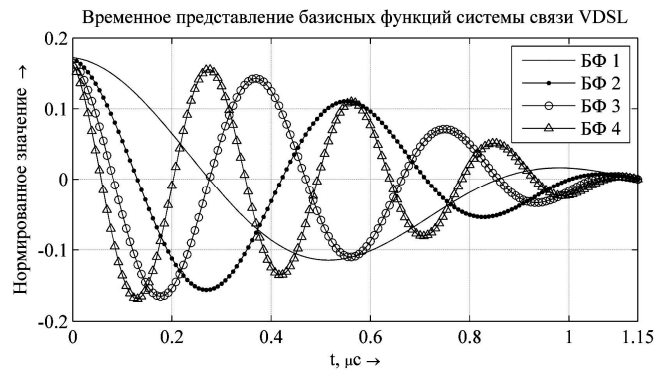


Рис. 4. Построение графика с обозначением величин и единиц измерения

График, представленный на рис. 4, соответствует требованиям ЕСКД, за исключением расхождения с пунктом 3.3 рекомендаций, предписывающим выполнять линии координатной сетки и делительные штрихи сплошной тонкой линией. При построении графика по умолчанию в среде MATLAB линии координатной сетки выполняются пунктирной линией, для изменения типа линии достаточно изменить свойство `GridLineStyle` координатных осей (результат представлен на рис. 5):

```
set(gca, 'GridLineStyle', '-'); % gca - дескриптор координатных осей текущего графика
grid(gca); % добавление координатной сетки
```

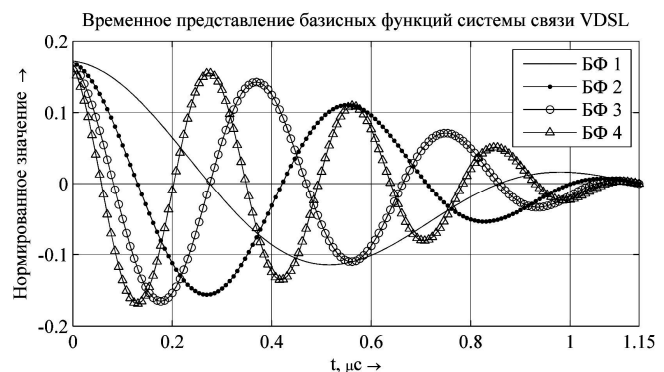


Рис. 5. Построение графика в соответствии с требованиями ЕСКД



Окончательный код скрипта, выполняющего построение графика, приведен ниже:

```
function result = figure_eskd()
    n = 4; % количество кривых
    n_samples = 150; % количество опорных точек кривой
    norm_t = 1:n_samples; % нормированное время

    % В качестве примера кривых используются базисные функции (несущие колебания)
    % системы связи по цифровым абонентским линиям (технология VDSL)
    T = 2.3e-4; % период вспомогательного колебания в секундах
    dtDSL = 1.15e-6; % длительность тактового интервала в секундах
    dT = T/n_samples;
    data = zeros(n,n_samples); % инициализация матрицы данных, заполнение нулями

    % Цикл заполнения матрицы данных
    for i = 1:n
        freq = 4312.5*i; % изменение частоты очередного несущего колебания
        data(i,:) =
            cos(2*pi*freq*dT*norm_t).*sqrt(2/T).*(1/sqrt(T)).*sinc(norm_t.*dT./T);
        data(i,:) = data(i,:) ./
            norm(data(i,:)); % нормировка значений
    end

    markerStyle = {'.', '.', 'o', '^', 's', '*'}; % тип маркера
    markerSize = {1 8 5 5 8 5}; % размер маркера
    lineColor = {'k'}; % цвет линии (k - черный)
    lineStyle = {'-'}; % тип линии
    lineWidth = {0.5}; % толщина линии

    for i = 1:n
        plot(0:n_samples-1, data(i,:), ...
            markerStyle{mod(i-1,length(markerStyle))+1}, ...
            'MarkerSize', markerSize{mod(i-1,length(markerSize))+1}, ...
            'Color', lineColor{mod(i-1,length(lineColor))+1}, ...
            'LineStyle', lineStyle{mod(i-1,length(lineStyle))+1}, ...
            'LineWidth', lineWidth{mod(i-1,length(lineWidth))+1});
        if i == 1 hold on; end;
    end;
    hold off;

    set(gca, 'GridLineStyle', '-'); % gca - дескриптор координатных осей текущего графика
    grid(gca); % добавление координатной сетки

    % Явное указание шрифта позволяет избежать проблем с кириллическими символами в надписях
    set(gca, 'FontName', 'Times New Roman Cyr');
    set(gca, 'FontSize', 9); % размер шрифта
```

```
labels = {};
for i = 1:n
    labels = cat(1,labels,['BФ ' num2str(i)]); % BФ - базисная функция
end
legend(labels, 1);

% Подписи к осям, название графика
xlabel('t, \muc \rightarrow');
ylabel('Нормированное значение \rightarrow');
title('Временное представление базисных функций системы связи VDSL');

xTickStep = 0.2e-6; % шаг координатной сетки по горизонтали = 0.2 мкс
len = length(data(i,:)); % количество откладываемых отсчетов по оси абсцисс
xTickLabels = {'0'}; % подписи к делительным штрихам
for i = 1:(dtDSL/xTickStep)
    xTickLabels =
        cat(1,xTickLabels,[num2str(i*xTickStep*1e6,3) '']);
end
xTickLabels =
    cat(1,xTickLabels,[num2str(dtDSL*1e6,3)]); % подпись последнего штриха
set(gca, 'XTick',
[1:round(xTickStep*len/dtDSL):len len-1],
'XTickLabel', xTickLabels, 'Xlim', [1 len-1]); % изменение параметров
end
```

Указанный исходный код выполняет построение четырех кривых на одном графике. В случае, когда требуется одновременно изображать большее количество функциональных зависимостей в одной диаграмме, следует подобрать набор элементов структур данных *markerStyle*, *markerSize*, *lineColor*, *lineStyle* и *lineWidth* таким образом, чтобы обеспечить максимальное удобство пользования диаграммой.

Анализ способов выполнения графиков в отечественных академических изданиях показал, что большинство требований ЕСКД выполняется, однако допускаются незначительные расхождения, не влияющие на наглядность и удобство пользования диаграммами. Так, линии координатной сетки часто выполняются пунктирной линией, а стрелки, указывающие направления возрастания значений величин, отсутствуют. Учитывая рекомендательный статус документа Р 50-77-88 [3], такие расхождения допустимы.

Заключение

Графики представляют результаты измерений в наиболее наглядной форме, выдавая максимум информации на минимальном пространстве [8]. Правильное построение графика позволяет выявить характерные особенности изучаемых зависимостей: области возрастания и убывания, максимумы и минимумы, области наибольшей и наименьшей скорости изменения, периодичность и т.д. Кроме того, графики позволяют быстро проверить соответствие теории и экспериментальных данных.

В работе подробно рассмотрен пример построения

графика нескольких функций в среде MATLAB. Отдельное внимание уделено вопросам наглядности построения и соответствия требованиям ЕСКД.

Не рассматриваются вопросы выбора интервалов изменения переменных, выбора масштабов и другие, что обусловлено несколькими причинами. Во-первых, приведенный исходный код позволяет решить указанные вопросы настройкой параметров (выбор интервалов изменения переменных определяется первыми двумя параметрами функции *plot()*, масштаб задается параметрами «*Xlim*» и «*Ylim*» и т.д.) следовательно, не требуется внесение дополнительных инструкций. Во-вторых, подобные аспекты в деталях изложены в пособии [8], с которым будет полезно ознакомиться каждому исследователю.

Не стоит забывать, что графический метод эффективен лишь при его грамотном применении. Освоение рассматриваемого метода поможет исследователю повысить наглядность и удобство восприятия представляемых графиков и диаграмм, и, как ожидаемый результат, получить общую положительную оценку работы со стороны специалистов.

Литература

1. Власов, В.Г. Новый энциклопедический словарь изобразительного искусства : в 10 т., т. VI / В.Г. Власов. – СПб : Азбука-классика, 2007. – 591 с.
2. Большая советская энциклопедия : в 30 т. – М.: Советская энциклопедия, 1969–1978.
3. Рекомендации Р 50-77-88. Единая система конструкторской документации. Правила выполнения диаграмм. – Взамен ГОСТ 2.319–81 ; введ. 1989.01.01. – М. : Изд-во стандартов, 1989. – 9 с.
4. ГОСТ 7.32-2001. Система стандартов по информации, библиотечному и издательскому делу. Отчет о научно-исследовательской работе. Структура и правила

оформления. – Введ. 2002.01.07. – М. : Изд-во стандартов, 2001. – 21 с. – (Межгосударственный стандарт).

5. ГОСТ 2.105-95. Единая система конструкторской документации. Общие требования к текстовым документам. – Введ. 1996.01.07. – М. : Изд-во стандартов, 1996. – 31 с. – (Межгосударственный стандарт).

6. ГОСТ 7.0.11-2011. Система стандартов по информации, библиотечному и издательскому делу. Диссертация и автореферат диссертации. Структура и правила оформления. – Введ. 2011.12.13. – М. : Стандартинформ, 2012. – 12 с. – (Национальный стандарт Российской Федерации).

7. Сквайрс, Дж. Практическая физика : [пер. с англ.] / Дж. Сквайрс ; под ред. Е. М. Лейкина. – М. : Мир, 1971. – 205 с.

8. Светозаров, В.В. Элементарная обработка результатов измерений: учеб. пособие / В.В. Светозаров. – М. : Изд. МИФИ, 1983. – 52 с.

9. G.993.1 ITU-T Recommendation. Very high speed digital subscriber line transceivers (VDSL) [Электронный ресурс]. – Электрон. дан. – Режим доступа: <http://www.itu.int/rec/T-REC-G.993.1/en/>.

10. ГОСТ 2.307-2011. Единая система конструкторской документации. Нанесение размеров и предельных отклонений. – Введ. 2012.01.01. – М.: Стандартинформ, 2012. – 37 с. – (Межгосударственный стандарт).

THE FEATURES OF DATA PLOTTING IN MATLAB

Rybolovlev A.A., Rybolovlev D.A.

The specifics of plotting in MATLAB environment is discussed. The example of drawing multiple functions on the same graph is given. Special attention is paid to plot clarity and compliance with the requirements of Russian national technical standards.

Уважаемые авторы!

Редакция научно-технического журнала «Цифровая обработка сигналов» просит Вас соблюдать следующие требования к материалам, направляемым на публикацию:

Требования к текстовым материалам и сопроводительным документам:

1. Текст - текстовый редактор *Microsoft Word*.
2. Таблицы и рисунки должны быть пронумерованы. На все рисунки, таблицы и библиографические данные указываются ссылки в тексте статьи.
3. Объем статьи до 12 стр. (шрифт 12). Для заказных обзорных работ объем может быть увеличен до 20 стр.
4. Название статьи на русском и английском языках.
5. Рукопись статьи сопровождается:
 - краткой аннотацией на русском и английском языках;
 - номером УДК;
 - сведениями об авторах (Ф.И.О., организация, должность, ученая степень, телефоны, электронная почта);
 - ключевыми словами;
 - актом экспертизы (при наличии в вашей организации экспертной комиссии).

Требования к иллюстрациям:

- Векторные (схемы, графики) - желательно использование графических редакторов *Adobe Illustrator* или *Corel DRAW*.
- Растровые (фотографии, рисунки) - М 1:1, разрешение не менее 300dpi, формат *tiff, jpg*.