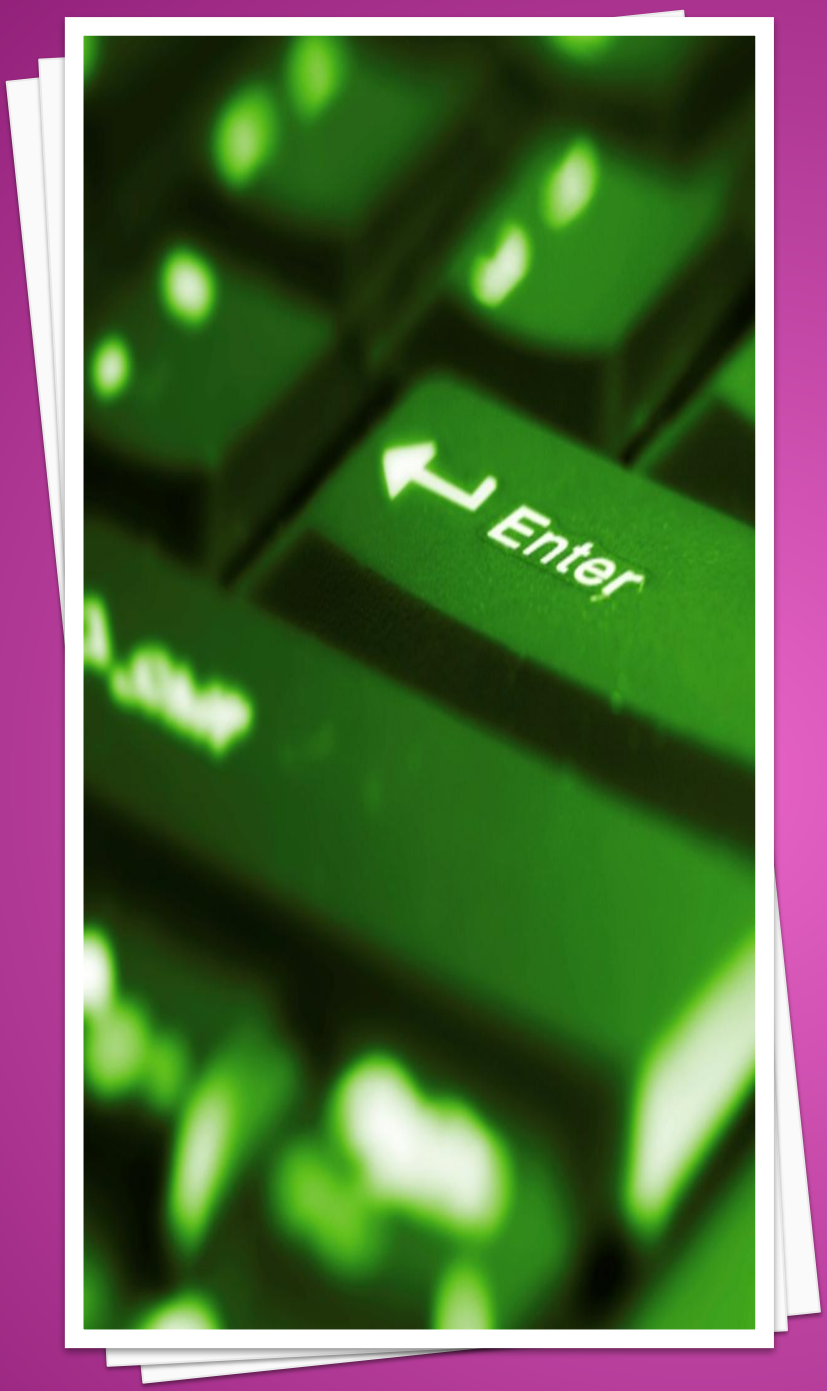

ГРАФИЧЕСКОЕ ПРОГРАММИРОВАНИЕ В СРЕДЕ MATLAB

Построение простейших графиков

1.5.1. Процедура plot

Вывод графиков в системе MatLAB - настолько простая и удобная операция, что ее можно использовать даже в режиме калькулятора.

Основной функцией, обеспечивающей построение графиков на экране дисплея, является функция plot. Общая форма обращения к ней такова:

plot(x1, y1, s1, x2, y2, s2,...).

Здесь x1, y1 - заданные векторы, элементами которых являются массивы значений аргумента (x1) и функции (y1), отвечающие первой кривой графика; x2, y2 - массивы значений аргумента и функции второй кривой и т.д. При этом предполагается, что значения аргумента откладываются вдоль горизонтальной оси графика, а значения функции - вдоль вертикальной оси. Переменные s1, s2,... являются символьными (их указание не является обязательным). Любая из них может содержать до трех специальных символов, определяющих соответственно: а) тип линии, которая соединяет отдельные точки графика; б) тип точки графика; в) цвет линии. Если переменные s не указаны, то тип линии по умолчанию - отрезок прямой, тип точки - пиксел, а цвет устанавливается (в версии 5.3) в такой очередности: - синий, зеленый, красный, голубой, фиолетовый, желтый, черный и белый - в зависимости от того, какая по очереди линия выводится на график.

Например, обращение вида plot(x1,y1,x2,y2,...) приведет к построению графика, в котором первая кривая будет линией из отрезков прямых синего цвета, вторая кривая - такого же типа зеленой линией и т.д.

Графики в MatLAB всегда выводятся в отдельное графическое окно, кото-
рое называют фигурой.
Приведем пример. Пусть нужно вывести график функции
y = 3sin(x +π /3)
на промежутке от -3π до +3π с шагом π /100.
Сначала надо сформировать массив значений аргумента x:
x = -3*pi : pi/100 : 3*pi,
потом вычислить массив соответствующих значений функции:
y = 3*sin(x+pi/3)
и, наконец, построить график зависимости y(x).
В командном окне последовательность операций будет выглядеть так:

```
» x = -3*pi:pi/100:3*pi;
» y = 3*sin(x+pi/3);
» plot(x,y)
```

Если вектор аргумента при обращении к функции plot не указан явно, то
система по умолчанию принимает в качестве аргумента номера элементов век-
тора функции.
Недостаток устраняется с помощью функции grid. Если к этой
функции обратиться сразу после обращения к функции plot:

```
» x = -3*pi:pi/100:3*pi;
» y = 3*sin(x+pi/3);
» plot(x,y), grid,
```

то график будет снабжен координатной сеткой.
Ценной особенностью графиков, построенных в системе MatLAB, является
то, что сетка координат всегда соответствует "целым" шагам изменения, что
делает графики "читабельными", т. е. такими, что по графику можно отсчитывать
значение функции при любом заданном значении аргумента и наоборот.
Заголовок графика выводится с помощью процедуры title. Если после об-
ращения к процедуре plot вызвать title таким образом:
title(‘<текст>’),

то над графиком появится текст, записанный между апострофами в скобках. При
этом следует помнить, что текст всегда должен помещаться в апострофы.
Аналогично можно вывести объяснения к графику, которые размещаются
вдоль горизонтальной оси (функция xlabel) и вдоль вертикальной оси (функция
ylabel).
Например, совокупность операторов

```
» x = -3*pi : pi/100 : 3*pi;
» y = 3*sin(x+pi/3);
» plot(x,y), grid
» title('Функция y = 3*sin(x+pi/3)');
» xlabel('x');   ylabel('y');
```

Не сложнее вывод в среде MatLAB графиков функций, заданных парамет-
рически. Пусть, например, необходимо построить график функции y(x), заданной
формулами:
x = 4 e -0,05 t sin t; y = 0,2 e -0,1 t sin 2t.
Выберем диапазон изменения параметра t от 0 до 50 с шагом 0.1. Тогда, на-
бирая совокупность операторов

```
t = 0:0.1:50;
x = 4*exp(-0.05*t).*sin(t);
y = 0.2*exp(-0.1*t).*sin(2*t);
plot(x,y)
set(gcf,'color','white')
title('Параметрическая функция  x=4*exp(-0.05t)*sin(t);
y= 0.2*exp(-0.1t)*sin(2t) ')
```

Большим удобством, предоставляемым системой MatLAB, является указанная ранее возможность не указывать аргумент функции при построении ее графика. В этом случае в качестве аргумента система принимает номер элемента вектора, график которого строится. Пользуясь этим, например, можно построить "график вектора":

```
» x = [ 1 3 2 9 6 8 4 6];
» plot (x)
» grid
» title('График вектора X')
» ylabel('Значение элементов')
» xlabel(' Номер элемента').
```

Результат представлен на рис. 1.32.

Еще более наглядным является представление вектора в виде столбцовой диаграммы с помощью функции bar (см. рис. 1.33):

```
» bar(x)
» title('График  вектора X')
» xlabel(' Номер элемента')
» ylabel('Значение элементов')
```

Если функция задана своими значениями при дискретных значениях аргумента, и неизвестно, как она может изменяться в промежутках между значениями аргумента, удобнее представлять график ее в виде отдельных вертикальных линий для любого из заданных значений аргумента. Это можно сделать, применяя процедуру stem, обращение к которой целиком аналогично обращению к процедуре plot:

```
 x = [ 1 3 2 9 6 8 4 6];
```

```
stem(x,'k')
 grid
 set(gca,'FontName','Arial','FontSize',14),
 title('График векторы X')
 ylabel('Значение элементов')
 xlabel(' Номер элемента')
```

Другой пример - построение графика функции в виде столбцовой диаграммы

y e x = − 2

```
» x = - 2.9 : 0.2 : 2.9;
» bar(x, exp(-x . * x))
» title('Стовпцева диаграмма функции y = exp(-x2)')
» xlabel (' Аргумент x')
» ylabel (' Значение функции y')
```

Еще одна полезная инженеру функция - hist (построение графика гисто-граммы заданного вектора). Стандартное обращение к ней таково:

hist(y, x),

где y - вектор, гистограмму которого нужно построить; x - вектор, элементы его определяют интервалы изменения первого вектора, внутри которых подсчитывается количество элементов вектора “y”.

Эта функция осуществляет две операции:

- подсчитывает количество элементов вектора “y”, значения которых попадают внутрь соответствующего диапазона, указанного вектором “x”;
- строит столбцовую диаграмму подсчитанных чисел элементов вектора “y” как функцию диапазонов, указанных вектором “x”.

В качестве примера рассмотрим построение гистограммы случайных величин, которые формируются встроенной функцией randn. Примем общее количество элементов вектора этих случайных величин 10 000. Построим гистограмму для диапазона изменения этих величин от -2,9 до +2,9. Интервалы изменения пусть будут равны 0,1.

Тогда график гистограммы можно построить с помощью совокупности таких операторов:

```
x = -2.9:0.1:2.9;
y = randn(10000,1);
hist(y,x)
set(gca,'fontsize',14)
ylabel('Количество из 10000')
xlabel('Аргумент')
title('Гистограмма нормального распределения')
```

В частности, вытекает, что встроенная функция randn достаточно верно отображает нормальный гауссовый закон распределения случайной величины.

Процедура comet(x,y) ("комета") строит график зависимости y(x) постепенно во времени в виде траектории кометы. При этом изображающая точка на графике имеет вид маленькой кометы (с головкой и хвостиком), которая плавно перемещается от одной точки к другой. Например, если ввести совокупность операторов:

```
» t = 0:0.1:50;
» x = 4 * exp(-0.05*t) .* sin(t);
» y = 0.2 * exp(-0. 1*t) . * sin(2*t);
» comet(x,y),
```

то график, приведенный на рис. 1.31, будет построен как траектория последовательного движения кометы. Это обстоятельство может быть полезным при построении пространственных траекторий для выявления характера изменения траектории с течением времени.

MatLAB имеет несколько функций, которые позволяют строить графики в логарифмическом масштабе. К примеру, функция logspace с обращением x = logspace(d1, d2, n) формирует вектор-строку "x", содержащую "n" равноотстоящих в логарифмическом масштабе друг от друга значений, которые покрывают диапазон от 10 d1 до 10 d2 .

Функция loglog полностью аналогична функции plot, но графики по обеим
осям строятся в логарифмическом масштабе.
Для построения графиков, которые используют логарифмический масштаб
только по одной из координатных осей, пользуются процедурами semilogx и
semilogy. Первая процедура строит графики с логарифмическим масштабом вдоль
горизонтальной оси, вторая - вдоль вертикальной оси. Обращение к последним
трем процедурам аналогично обращению к функции plot.
В качестве примера рассмотрим построение графиков амплитудно-
частотной и фазочастотной характеристик звена, описываемого передаточной
функцией:
W p
p
p p
( ) =
+
+ · +
4
4 100 2
.
Для этого нужно, во-первых, создать полином числителя Pc = [1 4] и знаме-
нателя передаточной функции Pz = [1 4 100]. Во-вторых, определить корни этих
двух полиномов:

```
» P1 = [1 4];   P2 = [1 4 100];
» roots(P1)
ans =    -4
» roots(P2)
ans =
-2. 0000e+000 +9. 7980e+000i
-2. 0000e+000 -9. 7980e+000i
```

В-третьих, задать диапазон измения частоты так, чтобы он охватывал все
найденные корни:
om0 = 1e-2; omk = 1e2.

Теперь нужно задаться количеством точек будущего графика (например, n = 41), и сформировать массив точек по частоте в логарифмическом масштабе

OM = logspace(-2,2,41),

где значения -2 и +2 отвечают десятичным порядкам начального om0 и конечного omk значений частоты.

Пользуясь функцией polyval, можно вычислить сначала вектор "ch" комплексных значений числителя частотной передаточной функции, отвечающих заданной передаточной функции по Лапласу, если в качестве аргумента функции polyval использовать сформированный вектор частот OM, элементы которого умножены на мнимую единицу (см. определение Частотной Передаточной Функции). Аналогично вычисляется комплекснозначний вектор "zn" знаменателя ЧПФ.

Вектор значений АЧХ (амплитудно-частотной характеристики) можно найти, расчитывая модули векторов числителя и знаменателя ЧПФ и поэлементно деля полученные векторы. Чтобы найти вектор значений ФЧХ (фазочастотной характеристики), нужно разделить поэлементно комплекснозначные векторы числителя и знаменателя ЧПФ и определить вектор аргументов элементов полученного вектора. Для того чтобы фазу представить в градусах, полученные результаты следует домножить на 180 и разделить на π .

Наконец, для построения графика АЧХ в логарифмическом масштабе, достаточно применить функцию loglog, а для построения ФЧХ удобнее воспользоваться функцией semilogx.

В целом последовательность действий может быть такой:

```
P1 = [1 4];  P2 = [1 4 100];
OM = logspace(-2,2,40);
ch = polyval(P1,i*OM);   zn = polyval(P2,i*OM);
ACH = abs(ch)./abs(zn);
loglog(OM,ACH);grid;    set(gca,'FontSize',12)
title('График Амплитудно-Частотной Характеристики')
xlabel('Частота  (рад/с)'); ylabel('Отношение  амплитуд')
```

```
FCH = angle(ch./zn)*180/pi;
semilogx(OM,FCH);  grid,
title('Фазо-Частотная Характеристика'),
xlabel('Частота  (рад/с)'), ylabel('Фаза  (градусы)')
```

Обычно графики, получаемые с помощью процедур plot, loglog, semilogx и semilogy, автоматически строятся в таких масштабах по осям, чтобы в поле графика поместились все вычисленные точки графика, включая максимальные и минимальные значения аргумента и функции. Тем не менее, MatLAB имеет возможности установления и других режимов масштабирования. Это достигается за счет использования процедуры axis.

Команда axis([xmin xmax ymin ymax]) устанавливает жесткие границы поля графика в единицах величин, которые откладываются по осям.

Команда axis(‘auto') возвращает масштабы по осям к их штатному значению (принятому по умолчанию).

Команда axis(‘ij') перемещает начало отсчета в левый верхний угол и реализует отсчет от него (матричная система координат).

Команда axis(‘xy') возвращает декартову систему координат с началом отсчета в левом нижнем углу графика.

Команда axis(‘square') устанавливает одинаковый диапазон изменения переменных по осям графика.

Команда axis(‘equal') обеспечивает одинаковый масштаб по обоих осям графика.

В одном графическом окне, но на отдельных графических полях можно по-
строить несколько графиков, используя процедуру subplot. Обращение к этой
процедуре должно предшествовать обращению к процедурам plot, loglog, semilogx
и semilogy и иметь такой вид:
subplot(m,n,p).
Здесь m - указывает, на сколько частей разделяется графическое окно по
вертикали, n - по горизонтали, а p - номер подокна, в котором будет строиться
график. При этом подокна нумеруются слева направо построчно сверху вниз (так,
как по строкам читается текст книги).
Например, два предшествующих графика можно поместить в одно графиче-
ское окно следующим образом:

```
subplot(2,1,1);  loglog(OM,ACH,'k'); grid;
set(gca,'FontName','Arial','FontSize',12),
title('Амплитудно-Частотная Характеристика'),
ylabel('Амплитуда'),
subplot(2,1,2);  semilogx(OM,FCH,'k'); grid
title('Фазо-Частотная Характеристика')
xlabel('Частота (рад/с)'),  ylabel('Фаза (гр.)')
```

Команда text(x, y, ‘<текст>’) позволяет расположить указанный текст на по-
ле графика, при этом начало текста помещается в точку с координатами x и y.
Значения указанных координат должны быть представлены в единицах величин,
откладываемых по осям графика, и находиться внутри диапазона изменения этих
величин. Часто это неудобно, так как требует предварительного знания этого
диапазона, что не всегда возможно.
Более удобно для размещения текста внутри поля графика использовать
команду gtext(‘<текст>’), которая высвечивает в активном графическом окне пе-
рекрестие, перемещение которого с помощью мыши позволяет указать место на-
чала вывода указанного текста. После этого нажатием левой клавиши мыши или

любой клавиши текст вводится в указанное место:
» gtext(' Ч Х')
» subplot(2,1,1);
» gtext(' Ч Х')

Чтобы создать несколько графических окон, в любом из которых располо-
жены соответствующие графики, можно воспользоваться командой figure, кото-
рая создаст такое графическое окно, оставляя предшествующие.
Наконец, для того, чтобы несколько последовательно вычисленных графи-
ков были изображены в одном графическом окне в одном стиле, можно использо-
вать команду hold on, тогда каждый такой график будет строиться в том же пред-
варительно открытом графическом окне, т. е. каждая новая линия будет добав-
ляться к прежде построенным. Команда hold off выключает режим сохранения
графического окна, установленного предшествующей командой

Вывод графиков на печать

Чтобы вывести график из графического окна (фигуры) на печать, т. е. на
лист бумаги, следует воспользоваться командами меню, расположенного в верх-
ней части окна фигуры. В меню File выберите команду Print. Подготовьте прин-
тер к работе и нажмите кнопку <Ок> в окошке печати, - принтер распечатает со-
держимое графического окна на отдельном листе бумаги.
Для предварительной настройки на определенный тип принтера и установ-
ления вида печати используйте в том же меню File команду Print Setup.