

信号与系统实验教学大纲

“信号与系统仿真实验”是与“信号与系统”课程理论教学相配套而开设的计算机仿真实验课程，其目的在于实现在可视化的交互式实验环境中，以计算机为辅教学手段，以科技应用软件 MATLAB 为实验平台，辅助学生完成“信号与系统”课程中的数值分析、可视化建模及仿真调试，同时将“信号与系统”课程教学中难点、重点及部分课后练习，通过计算机来进行可视化的设计、调试和分析，从而将学生从繁杂的手工运算中解脱出来，把更多的时间和精力用于对信号与系统基本分析方法和原理的理解和应用上，培养学生主动获取知识和独立解决问题的能力，为学习后继专业课打下坚实的基础。

实验教学基本要求：

- 1、熟悉 MATLAB 的运行环境及基本操作命令，根据实验要求，认真完成基本数值算法的设计、编程、上机调试，分析运行结果，书写实验报告。
- 2、掌握用 MATLAB 对连续与离散信号进行可视化表示的方法，信号的时域运算、变换及 MATLAB 实现方法，学会应用 MATLAB 对常用信号进行时域特性分析及波形绘制。
- 3、掌握用 MATLAB 对线性系统的时域特性进行分析的基本方法。
- 4、掌握利用 MATLAB 对周期信号进行频谱分析的实现方法，重点掌握周期信号的频谱与信号周期及其时域宽度的变化规律。
- 5、掌握利用 MATLAB 对连续信号进行频域特性分析的基本方法，重点掌握傅里叶变换的符号实现、傅里叶变换的数值近似、傅里叶变换性质以及信号频谱分析的 MATLAB 实现方法。
- 6、掌握应用 MATLAB 进行连续系统频域分析的基本实现方法，重点掌握系统频率响应、幅频响应、相频响应曲线的绘制，系统的频率特性分析的 MATLAB 实现方法。
- 7、掌握应用 MATLAB 对连续系统进行复频域分析的基本方法，重点掌握拉普拉斯变换的三维可视化表现、连续系统的零极点图的绘制及拉普拉斯逆变换的 MATLAB 实现方法。

实验一 MATLAB 程序入门和基础应用

一、实验名称

MATLAB 程序入门和基础应用

二、实验目的

- 1.学习 Matlab软件的基本使用方法；
- 2.了解 Matlab的数值计算，符号运算，可视化功能；
3. Matlab程序设计入门

三、实验原理

MATLAB 如今已经被广泛地应用于各个领域，是当今世界上最优秀的数值计算软件。它广为流传的原因不仅在于在它的计算功能强大，图形功能丰富、方便，还在于它的编程效率高，扩充能力强；语句简单，易学易用，而不会像其他的那些高级语言一样距人于千里之外。在这里我们就对 MATLAB 做一下简单的介绍。

1、MATLAB 简介

在科学技术飞速发展的今天，计算机正扮演着越来越重要的角色。在进行科学研究与工程一用的过程中，科技人员往往会遇到大量繁重的数学运算和数值分析，传统的高级语言 BASIC、FORTRAN 及 C 语言等虽然能在一定程度上减轻计算量，但它们均要求应用人员具有较强的编程能力和对算法有深入的研究 [7]。

另外，在运用这些高级语言进行计算结果的可视化分析及图形处理方面，对非计算机专业的普通用户来说，仍存在着一定的难度。MATLAB 正式在这一应用要求背景下产生的数学类科技应用软件，它具有的顶尖的数值计算功能。强大的图形可视化功能及简介医学的“科学便笺式”工作环境和编程语言，从根本上满足了科技人员对工程数学计算的要求，并将科技人员从繁重的数学运算中解放出来，因而，越来越受到广大科技工作者的普遍欢迎。

MATLAB 是 matrix和 laborator前三个字母的缩写，意思是“矩阵实验室”，是 MathWorks 公司推出的数学类可以应用软件。其 DOS 版本 (MATLAB1.0) 发行于 1984 年，到现在已经到了 MATLAB7.X 。经过 20 多年的不断发展与完善，MATLAB 已发展成为由 MATLAB 语言、MATLAB 工作环境、MATLAB 图形处

理系统、MATLAB 数学函数库和 MATLAB 应用程序接口五大部分组成的集数值
计算、图形处理、程序开发为一体的功能强大的系统。MATLAB 由

“主包”和三十多个扩展功能和应用学科性的工具箱 (Toolboxes) 组成。

MATLAB 具有一下基本功能：

数值计算功能

符号计算功能

图形处理集可视化功能

可视化建模集动态仿真功能

MATLAB 语言是以矩阵计算为基础的程序设计语言，语法规则简单易学，用户不用花太多时间即可掌握其编程技巧。其指令格式与教科书中的数学表达式非常相近，用 MATLAB 编写程序犹如在便笺上列写公式和求解，因而被称为“便笺式”的编程语言。另外，MATLAB 还具有功能丰富和完备的数学函数库及工具箱，大量繁杂的数学运算和分析可通过调用 MATLAB 函数直接求解，大大提高了编程效率，其程序编译和执行速度远远超过了传统的 C 和 FORTRAN 语言，因而用 MATLAB 编写程序，往往可以达到事半功倍的效果。在图形处理方面，MATLAB 可以给数据以二维、三维乃至四维的直观表现，并在图形色彩、视角、品性等方面具有较强的渲染和控制能力，使科技人员对大量原始数据的分析变得轻松和得心应手。

正是由于 MATLAB 在数值计算及符号计算等方面的强大功能，使 MATLAB 一路领先，成为数学类科技应用软件中的佼佼者。目前，MATLAB 已成为国际上公认的最优秀的科技应用软件。MATLAB 的上述特点，使它深受工程技术人员及科技专家的欢迎，并很快成为应用学科计算机辅助分析、设计、仿真、教学等领域不可缺少的基础软件。目前，在国外高等院校，MATLAB 已成为本科生、研究生必须掌握的基础软件，国内很多理工院校也已经把 MATLAB 作为学生必须掌握的一种软件，“教育部全国计算机专业课程知道委员会”已将 MATLAB 语言列为推荐课程。

2、MATLAB 的安装

如今我们最常用的 MATLAB 版本是 MATLAB6.5 ，虽然它不是最高最新的版本，但是它已经把 MATLAB 的功能很好的汇总在一起。

MATLAB 既可在 PC 单机环境下亦可在网络环境下安装进行，在这里介绍 MATLAB6.5 在 PC 机单机环境使用 Microsoft Windows 2000 或者 Windows XP 操作系统进行安装的情况。

MATLAB6.5 对系统的基本要求为：

Microsoft Windows 2000 或者 Windows XP

奔腾处理器 (1.0G 上)

256MB 以上内存

16 位以上显卡

在系统满足上述要求的情况下，即可进行 MATLAB6.5 的安装，安装过程^[8]如下：

- 1) 可以到本校 E383 软件园下载 MATLAB6.5 安装程序，上传的文件一般为压缩格式，所以首先我们得把它解压缩后再点击 setup 安装图标开始安装，接着显示如图 1.1 所示的“欢迎进入 Mathworks 的 Release 13 的安装工具，开始启动 MATLAB6.5 的安装程序。

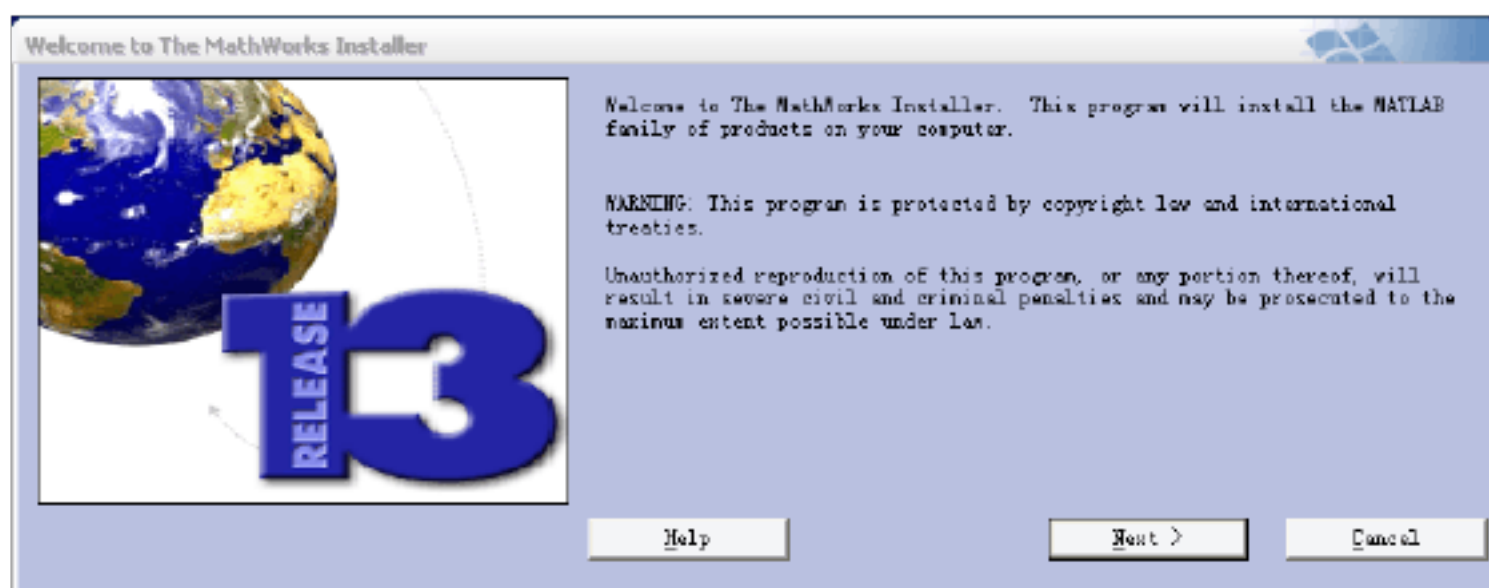


图 1.1 Mathworks 公司的软件安装工具界面

- 2) 安装程序自动打开 Mathworks 公司多的安装工具后，按下【Next】按钮即继续安装。
- 3) 安装程序自动进入注册对话框，用户在响应的编辑框内输入产品注册码（可以在解压缩后的文件夹中找到），然后按下【Next】按钮继续安装。
- 4) 用户认可 Mathworks 公司的软件协议后，继续在用户名称和公司名称的编辑框内输入相关信息。按下【Next】按钮继续安装。

- 5) 安装程序会自动打开 MATLAB6.5 安装内容选择对话框(图 1.2), MATLAB 的组件、安装目录路径以及安装所需要的磁盘空间等信息均显示在图中。在项目 1 下的编辑框内输入安装的路径与目录,并可点击【Browse】按钮,以浏览、选择并改变 MATLAB 的安装的路径与子目录。对于项目 2 下的 3 个单选钮可以选择仅安装系统或仅安装文件或系统文件都安装。在项目 3 下 2 个单选钮可以选择安装文件的语种:英语或英语与日语。对项目 4 下多个复选框可以勾选要安装的组件。

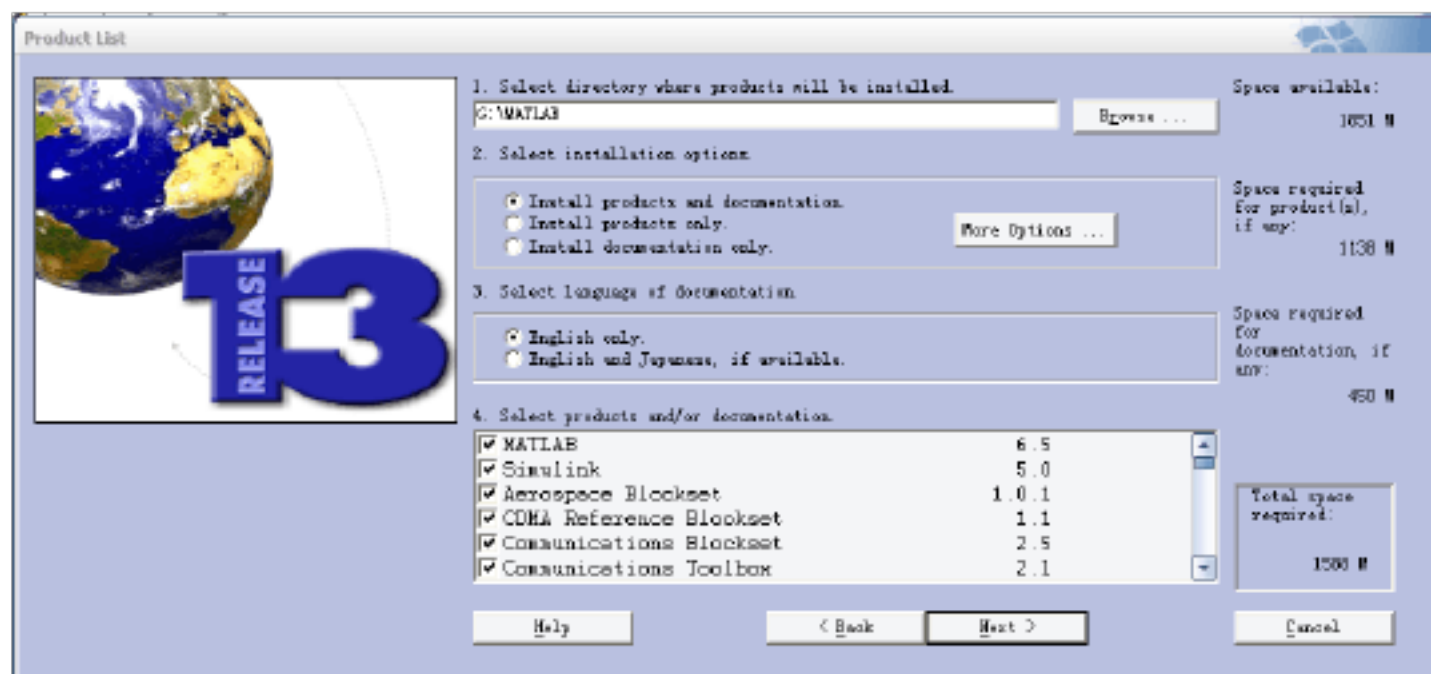


图 1.2 MATLAB6.5 安装目录路径与组件选择界面

在图 1.2中我们可以看到,安装全部 MATLAB6.5 时所需要的磁盘空间(约 1.6G),按下【Help】按钮即可获得相关的帮助;按下【Back】按钮可返回到上一步;按下【Cancel】按钮即可退出安装;按下【Next】按钮即可继续安装。

- 6) 安装完成时,有 MATLAB6.5 安装成功的提示界面如图 1.3 所示。用单选钮“立即重新启动计算机”或“步立即重新启动计算机”,然后再按下【Finish】按钮,则安装过程即告结束,同时在 Windows 的操作桌面上生成 MATLAB6.5 快捷图标。

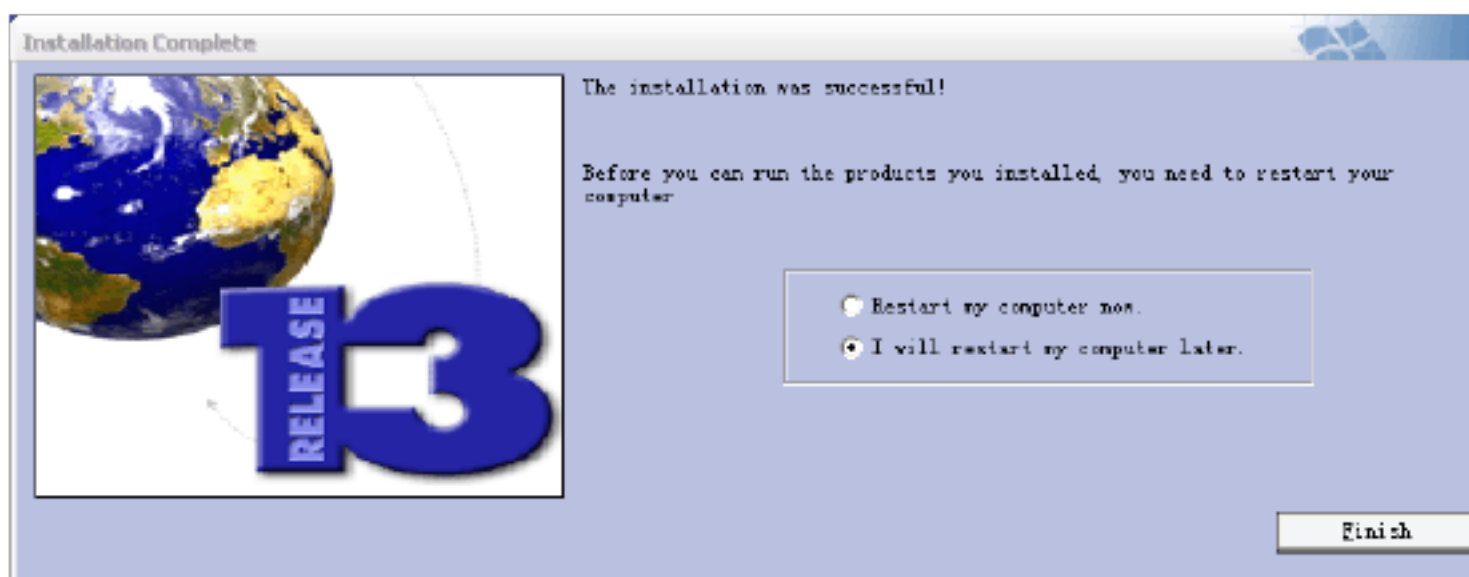


图 1.3 安装完成的提示界面

3、MATLAB 的启动与退出

MATLAB 的启动有如下两种方式：方式一：单击【开始】菜单，一次指向【程序】→【MATLAB 6.5】→【MATLAB6.5】，如图 1.4 所示，即可启动并打开 MATLAB 命令窗口。

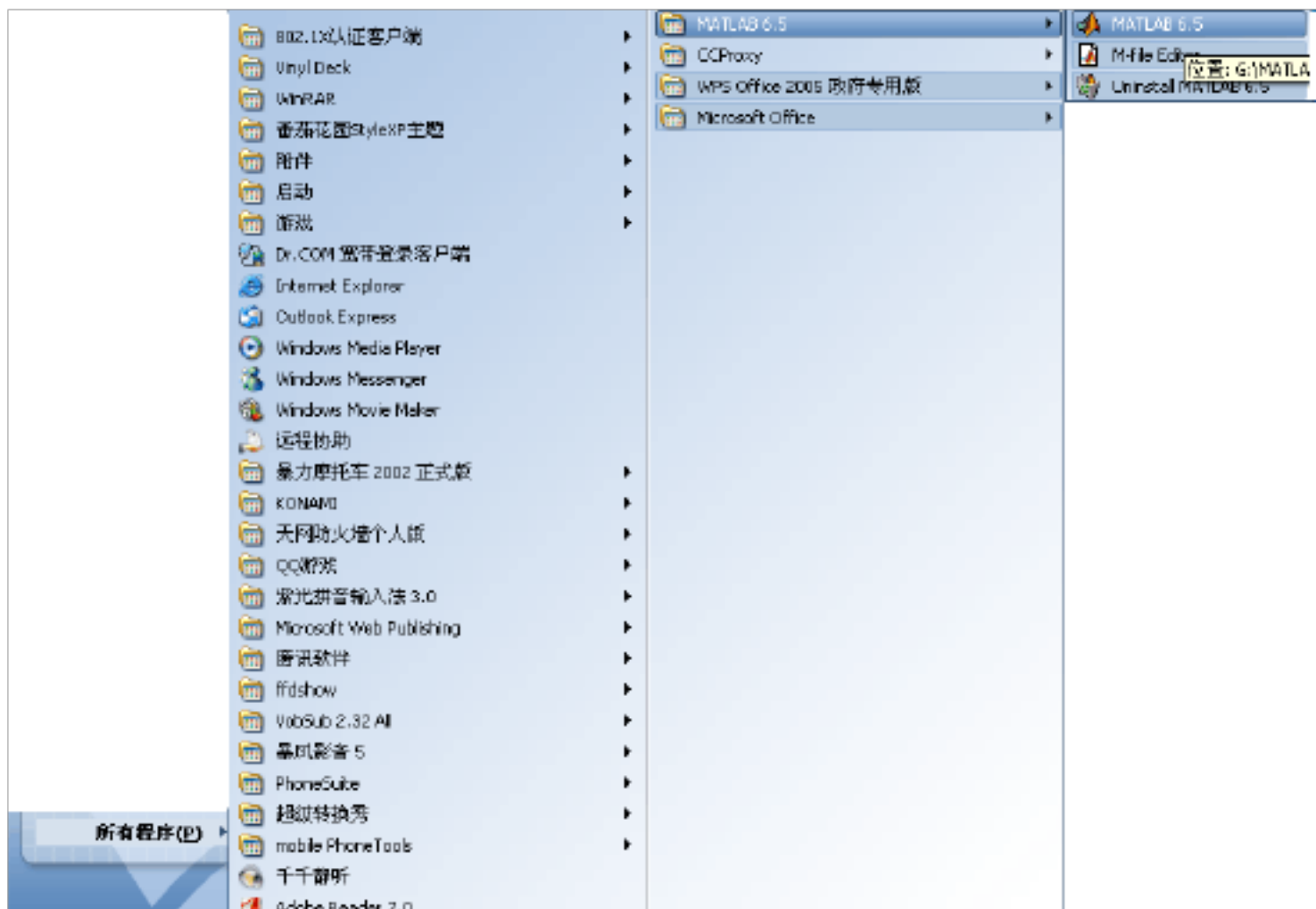


图 1.4 从开始菜单打开 MATLAB6.5

方式二：双击 Windows 2000 或 XP 操作系统桌面上的 MATLAB 快捷方式，即可启动并打开 MATLAB 命令窗口。

退出 MATLAB 非常简单，只需在 MATLAB 命令窗口内键入命令 quit 或单击命令窗口的【关闭】按钮即可。

4、MATLAB 的系统界面

MATLAB 既是一种高级计算机语言，又是一个编程环境。MATLAB 的系统界面，通常是指这个软件所具有的各种界面以及这些界面中的诸多菜单命令、工具栏按钮与对话框。通过对其操作，可以运行并管理系统：生成编辑与运行程序，管理变量与工作空间，输入输出数据与相关信息以及生成与管理 M 文件等。本节要介绍 MATLAB6.5 的系统界面、系统菜单项命令、系统工具按钮、系统界面的窗口、Star 开始按钮等。

MATLAB6.5 的系统界面如图 1.5 所示。在图中，最上面有“MATLAB”标题，标题栏的右边从左到右依次为窗口最小化、缩放和关闭按钮。标题栏下是条形主菜单，主菜单下是工具栏按钮与设置当前目录的弹出式菜单框及其右侧的查看目录树的按钮（【Browse for Folder】）。在工具栏下的大窗口就是 MATLAB 的主窗口，在大窗口里设置右 4 个小窗口（这是桌面平台的默认设置）：“Workspace”、“Current Directory”、“Command History”、“Command Windows”。在最下方是“Start”开始按钮。

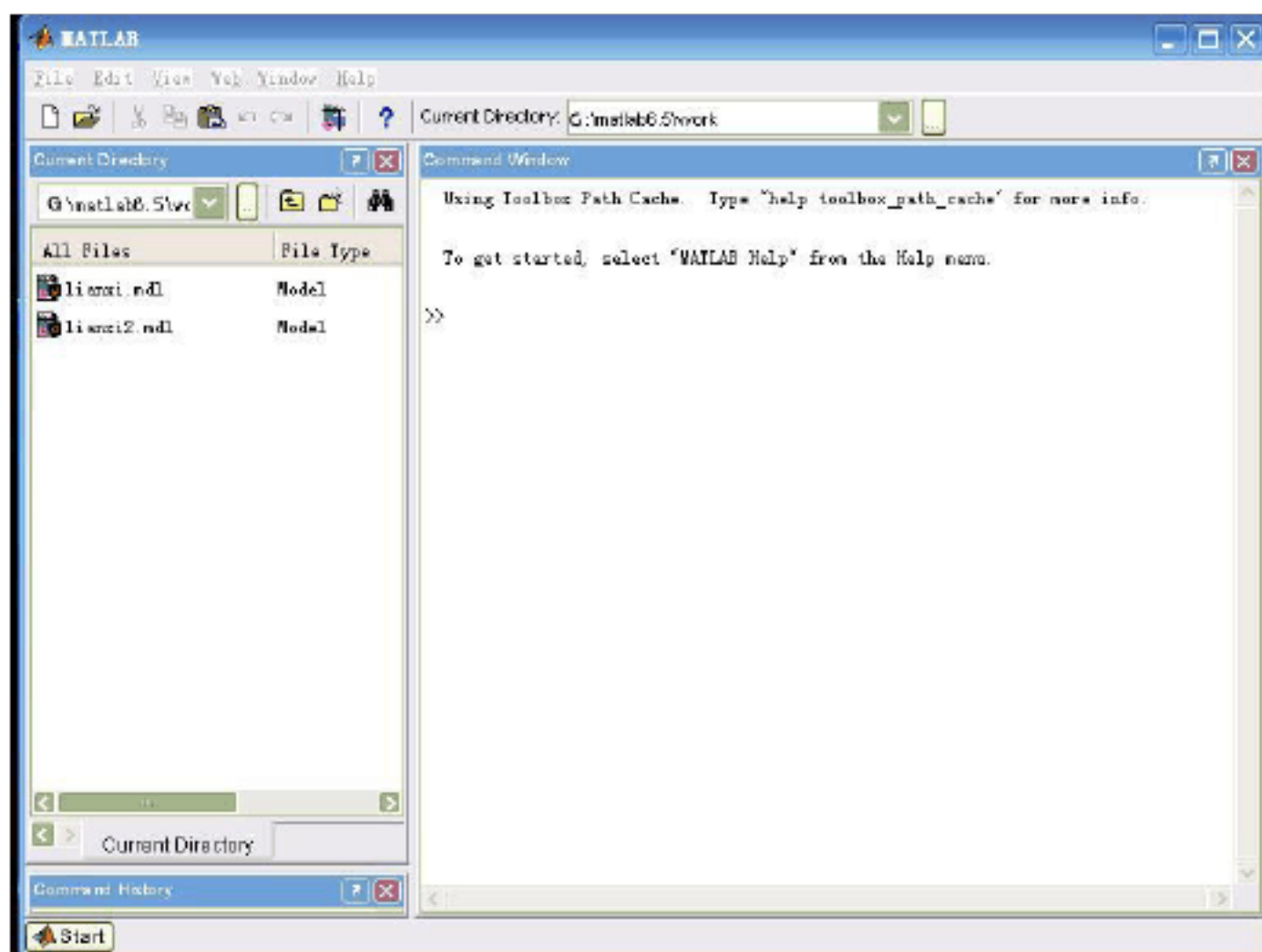


图 1.5 MATLAB 命令窗口

5、MATLAB 6.5 菜单项命令

条形组菜单有【文件(File)】、【编辑(Edit)】、【查看(View)】、【网络(Web)】、【窗口(Window)】、【帮助(Help)】等 6 个菜单项。

通过【文件(File)】我们可以根据目的的不同来新建和打开各种不同的文件，如 M 文件，Simulink 仿真库等等，而且可以把已经编辑好的文件进行存档。

【编辑(Edit)】菜单我们可以把对文件的操作进行撤销或者重复操作，以及进行复制、粘贴、剪切和删除等等。

【查看(View)】菜单的功能式用于查看各种界面。

【网络(Web)】菜单，我们可以在联网的情况下点击，可以看到一些 Mathworks 公司网站等等一些信息。

【窗口(Window)】菜单项下拉后仅有【关闭所有窗口】的子项。

【帮助(Help)】是一些使用 MATLAB 的帮助命令。

MATLAB 的工具栏以图标方式为用户提供了 MATLAB 的常用命令及操作。工具栏图标及对应功能如图 1.6 所示。

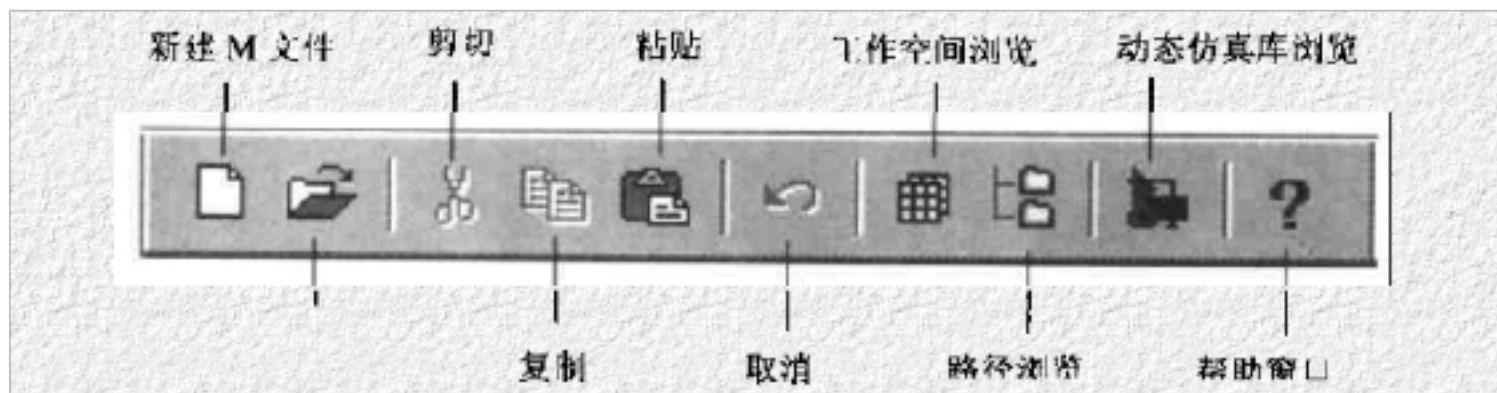


图 1.6 工具栏

命令窗口的工作区是用户使用 MATLAB 的重要空间，MATLAB 在这里为用户提供了交互式的工作环境，即用户可随时输入命令，计算机即时给出元素结果。用户只需输入简单易学的 MATLAB 命令，即可进行诸如数值计算、符号运算和运算结果的可视化等复杂的分析和处理。但要注意，每一条命令或命令行键入后都要按【Enter】(回车)键，命令才会被执行。例如，在命令窗口的工作区直接输入如下字符：`a=ones(3,3)`

然后按回车键，即可创建一个 3×3 且元素值为 1 的矩阵，并显示如下运行结果：

```
a=
     1     1     1
     1     1     1
     1     1     1
```

6、MATLAB 软件的使用

刚启动 MATLAB 进入到如图 1.5 所示的界面，但是这只是文件引索和命令执行窗口。我们可以通过点击【File】→【New】→【M-file】，进入到如图 1.7 所示的界面，也就是 M 程序编辑窗口，或者按下 Ctrl+n 键，或者直接点击新建空白 M 文件，这就是我们在接下来的实验中要用到的界面。

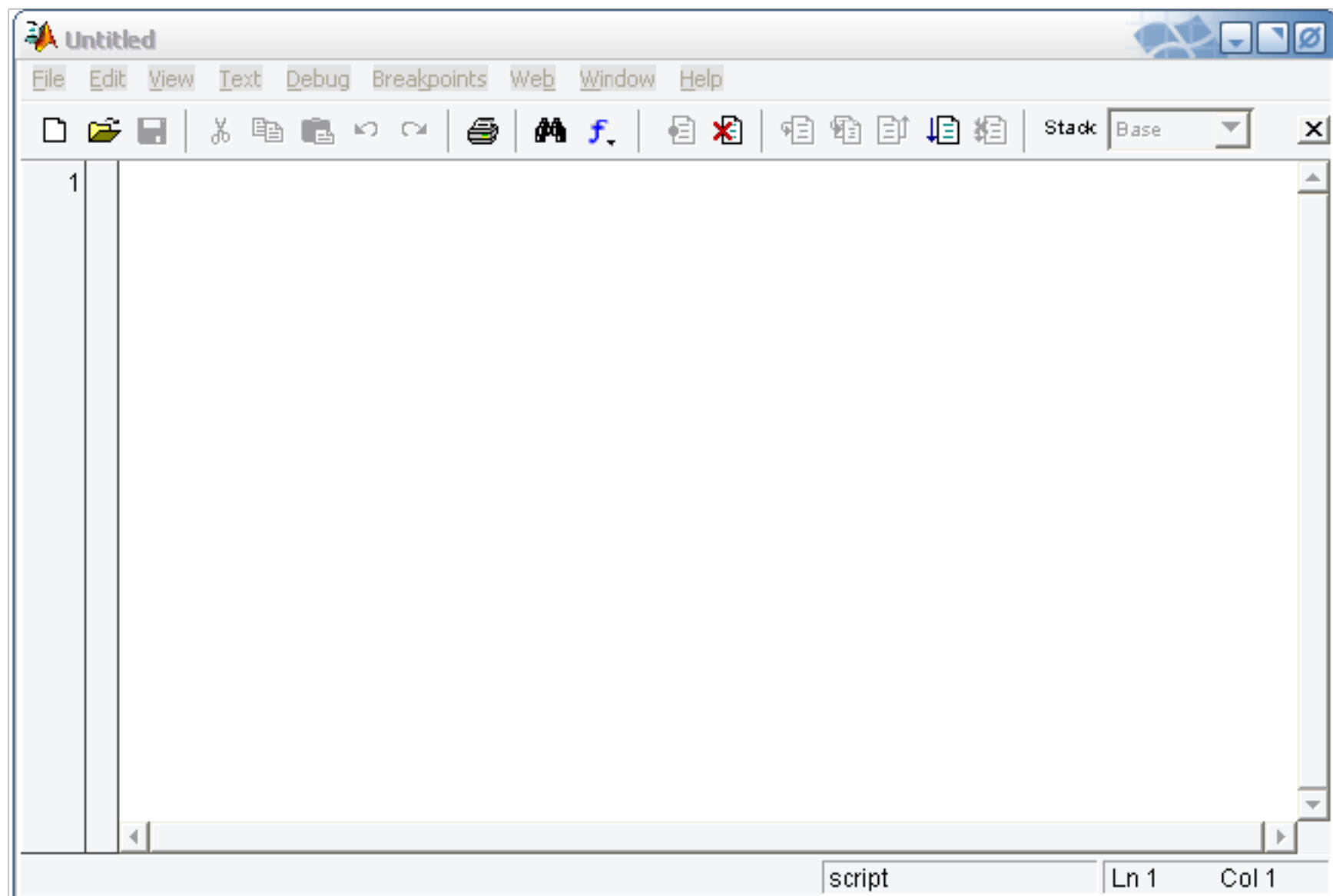


图 1.7 M程序编辑窗口

进入到上面的界面以后，就开始进行编写程序。我们把编写好的程序键入，然后再把程序保存起来，因为 MATLAB 不会编译没有保存的文件。点击【File】→【Save As...】，给文件输入一个容易辨别的名字，系统会以 .m 为后缀保存到 G:\MATLAB\work 的 work 文件夹中。保存之后，我们要对程序进行编译，点击【Debug】→【Run】，或者直接按 F5 就可以进行编译。编译成功，会听到“嘀”的一声，随之弹出程序运行结果。如果编译不成功，则会在图 1.5 所示窗口显示出哪一行出现了错误。对指出的错误进行修改，再次编译运行。

7、实验中要用的指令介绍

在整个信号与系统实验过程中，我们要利用 MATLAB 语言来编程实现对信号描述，时域变换，计算卷积等等一些过程。所以，对 MATLAB 中的简单常用指令必须有一个比较好的了解。下面对一些常用指令做简单的介绍。

`function` 在 MATLAB 中不是它的自带函数就可以完成所有功能，更多的时候是自己编写程序来实现我们要的功能，这时就要用到此命令，调用格式为：
`function ****` (括号外面为函数名称，括号中为函数中要用到的变量。

`plot`命令：`plot`命令是 MATLAB 中用来绘制用向量表示法(在下一章中讲到)表示的连续信号的波形。它的功能是将向量点用直线依次连接起来。调用格式：

`plot(k,f)`其中 k 和 f 是向量。

ezplot命令: ezplot命令是用来绘制用符号运算表示法表示的连续信号的波形。调用格式: ezplot(f,[t1,t2])其中[t1,t2]为一时间范围, f为以 t为变量的函数。

title命令: 在绘图命令中, 我们可以用此命令来对绘制出来的波形做一些注释, 以便后期我们做图形处理。调用格式为: title('..... ') 中间部分可以任意对图形进行注释的文字。

xlabel ylabel命令: 这两个也是来对绘制出来的波形做标注用的, 可以标注出两个坐标轴的未知数的意义, 增加图形中的信息量。调用格式: xlabel('..... '), ylabel('..... ') 中间可以是对坐标轴做注释的文字或字母。

axis命令: 此命令可以来定义绘制波形中坐标的范围。调用格式为: axis([k1,k2,g1,g2])其中 k1, k2 表示横坐标的范围, g1, g2 表示纵坐标的范围。

syms 命令: 在符号表示法中, 可以用此命令来定义变量。调用格式为: syms t 意思是定义一个变量 t。

sym 命令: 是符号表示法中的调用系统自带函数的命令。调用格式为: f=sym('.....') 中间为系统能识别的常用信号, 如正弦信号, e^{-nt} 等。

stem命令: 此命令专门用来绘制离散序列的波形。调用格式为: stem(k, f) 调用此命令可以绘制出离散序列的点状图。

subs命令: 此可以将连续信号中的时间变量 t用 t-t0, at等等来替换, 从而可以完成信号在时域范围内的变换。调用格式为: subs(f, t, t-t0) 通过调用此函数可以把信号做移位, 伸展等等变换。

flip命令: 此函数用来将向量以零时刻为基准点进行反折。调用格式为: f=flipr(f) 这样 f就是向量 f1反折后的函数。

min、max 命令: 这两个命令可以用来比较算出一个向量中的最小值和最大值, 或者比较得出两个值中的较小值。调用格式为: min(k), max(k), min(k1, k2) max(k1, k2)

length命令：此函数可以计算出向量的长度。调用格式为：length(f)

ones函数：这是 MATLAB 中一个常用的函数，它产生元素全部为 1 的矩阵，本文中用此函数来表示离散阶跃序列，或者定义连续的门信号，调用格式为： $n=0:5$ ；ones(1,n)表示长度为整数 n 的阶跃序列。 $k=1:p$ ；ones(1,length)表示长度为 k 的门信号。

subplot命令：在 MATLAB 绘图过程中，有时候为了便于观测图形的变化，需要在一幅波形显示窗口显示多个信号的波形，这时可以调用 subplot命令。调用格式为：subplot(n1,n2,k)定义一个 subplot(2,2,1)就可以在显示窗口中显示 $k=2 \times 2$ 个波形。

conv函数：这个函数是用来计算两个序列的卷积和，调用此函数，可以将两个给定的序列计算出卷积和，调用格式为 $f=\text{conv}(f1,f2)$

四、实验设备

计算机 MATLAB 软件

五、实验的预习

- 1、了解 MATLAB 的相关知识，包括它的功能，简单的函数用法等等；
- 2、学习 MATLAB 的常用矩阵运算函数，关系运算和逻辑运算，MATLAB 描述信号方法。了解 matlab 语言的绘图功能
- 3、预习 MATLAB 连续信号和离散信号的时域变换。

六、实验内容及具体步骤

- 1、打开 MATLAB 的系统界面，对其功能做一个大致了解；
- 2、学习变量的描述方法，掌握几个固定变量：I, j, pi, inf的使用。
注意，变量描述以字母开头，可以由字母、数字和下划线混合组成，区分字母大，小写字符长度不超过 31 个。
- 3、学习数值，矩阵，运算符，向量的矩阵运算，数组运算的描述方法。

(1) 用一个简单命令求解线性系统

$$3x_1 + x_2 - x_3 = 3.6$$

$$x_1 + 2x_2 + 4x_3 = 2.1$$

$$-x_1 + 4x_2 + 5x_3 = -1.4$$

$$A=[3 \ 1 \ -1;1 \ 2 \ 4;-1 \ 4 \ 5];b=[3.6;2.1;-1.4];$$

```
x=A\b
```

```
x = 1.4818 -0.4606 0.3848
```

(2) 用简短命令计算并绘制在 $0 \leq x \leq 6$ 范围内的 $\sin(2x)$ $\sin x^2$ $\sin^2 x$

```
x=linspace(0,6)
```

```
y1=sin(2*x), y2=sin(x.^2), y3=(sin(x)).^2;
```

```
plot(x, y1, x, y2, x, y3)
```

4、Matlab符号运算功能

(1) 符号运算的过程

在符号运算的整个过程中，所有的运算均是以符号进行的，即使以数字形式出现的量也是字符。做一个对 $\sin(x/2)$ 求导的过程。

在命令窗口中输入如下符号表达式按回车：

```
f='sin(x/2)';
```

```
dfdx=diff(f)
```

显示结果如下： $dfdx = 1/2*\cos(1/2*x)$

整个求导的过程都是由符号变量和符号表达式完成，没有涉及到具体的数值运算，其中 $1/2$ 也被当作是字符。

注意：符号变量前要先进行定义，定义语句是：`sym` 或 `syms` 变量名列表。

前者定义一个单一的符号变量，后者可以一次定义多个符号变量。如：

```
sym a    定义 a 为符号变量
```

```
sym a b c 定义 a, b, c 均为符号变量
```

符号表达式是由符号变量组成的一个表达式，符号方程是将一个符号表达式通过等号给一个符号变量。凡是用到 `sym` 命令的时候，表达式和方程式对空格都是敏感的，因此不要随意添加空格符到式中

(2) 符号表达式的创建：用 `sym` 命令直接创建符号表达式，这种创建方式不需要在前面有任何说明，使用快捷方便。例如：

```
f=sym('a+b+c')
```

在命令窗口中输入上述语句按回车键，出现以下结果： $f = a+b+c$ 说明已成功将符号表达式 $a+b+c$ 赋给变量 f

5、matlab语言的绘图功能

(1) plot——最基本的二维图形指令

用 plot 命令自动打开一个图形窗口 Figure 用直线连接相邻两数据点来绘制图形，根据图形坐标大小自动缩扩坐标轴，将数据标尺及单位标注自动加到两个坐标轴上，可自定坐标轴，可把 x, y 轴用对数坐标表示。任意设定曲线颜色和线型，给图形加坐标网线和图形加注功能。

分别完成 plot(x) plot(x, y) plot(x1, y1, x2, y2) 三种格式的调用，得出结论。

调用 plot(x, y, 's') 注意开关量字符串 s 设定的曲线颜色和绘图方式。

(2) 曲线绘图

观察 以下各段语句的绘图结果：

A: `x=[0, 0.48, 0.84, 1, 0.91, 0.6, 0.14]`

`[x1, x2,x3, x4, x5, x6, x7,]`

`plot (x)`

B: `t=0:pi/100:2*pi;`

`y=sin(t);y1=sin(t+0.25);y2=sin(t+0.5);`

`plot (t, y, t, y1, t, y2)`

C: `subplot(1, 3, 1); plot(t, y)`

`subplot(1, 3, 2); plot(t, y3)`

`subplot(1, 3, 3); plot(t, y2)`

D: `t=0:pi/100:2*pi;`

`y=sin(t);y1=sin(t+0.25);y2=sin(t+0.5);`

`plot (t, y, t, y1, t, y2)`

E: `subplot(3, 1, 1);`

`plot(t, y)`

`subplot(3, 1, 2);`

`plot(t, y3)`

`subplot(3, 1, 3);`

`plot(t, y2)`

F: `ezplot('sin(x)')`

`ezplot('sin(x)', 'cos(y)', [-4*pi 4*pi], figure(2))`

思考题:

1、说明产生一个 matlab 行向量的两种方法，其中行向量从 0 开始，并在 π 结束，共有 13 个等间隔点。

2、假定行向量 $x=[3\ 6\ 9]$, $y=[5\ 3\ 10]$ 列 matlab 运算的结果是什么?

A $x+y$ b, $x.*y$

c, $x.^y$ d, $x./y$

实验二 MATLAB 描述常用信号

一、实验名称：

MATLAB 描述常用信号

二、实验目的：

1. 在了解 MATLAB 这个软件的基本应用之后，学习用 MATLAB 描述常用信号的方法
2. 掌握连续时间信号和离散时间信号的描述
3. 学会用向量和符号表示法
4. 掌握信号的时域变换方法

三、实验原理：

MATLAB 强大的图形处理功能及符号运算功能，为我们实现信号的可视化提供了强有力的工具。在 MATLAB 中通常有两种方法来表示信号，一种是用向量来表示信号，另一种则是用符号运算的方法来表示信号。用适当的 MATLAB 语句表示出信号后，我们就可以利用 MATLAB 的绘图命令绘制出直观的信号波形。

1、连续时间信号

所谓连续时间信号，是指自变量的取值范围是连续的，且对于一切自变量的取值，除了有若干不连续点以外，信号都有确定的值与之对应的信号。从严格意义上来讲，MATLAB 并不能处理连续信号，在 MATLAB 中，是用连续信号在等时间间隔点的样值来近似地表示连续信号的，当取样时间间隔足够小时，这些离散的样值就能较好地近似出连续信号。在 MATLAB 中连续信号可用向量或符

号运算功能来表示。

(1) 向量表示法

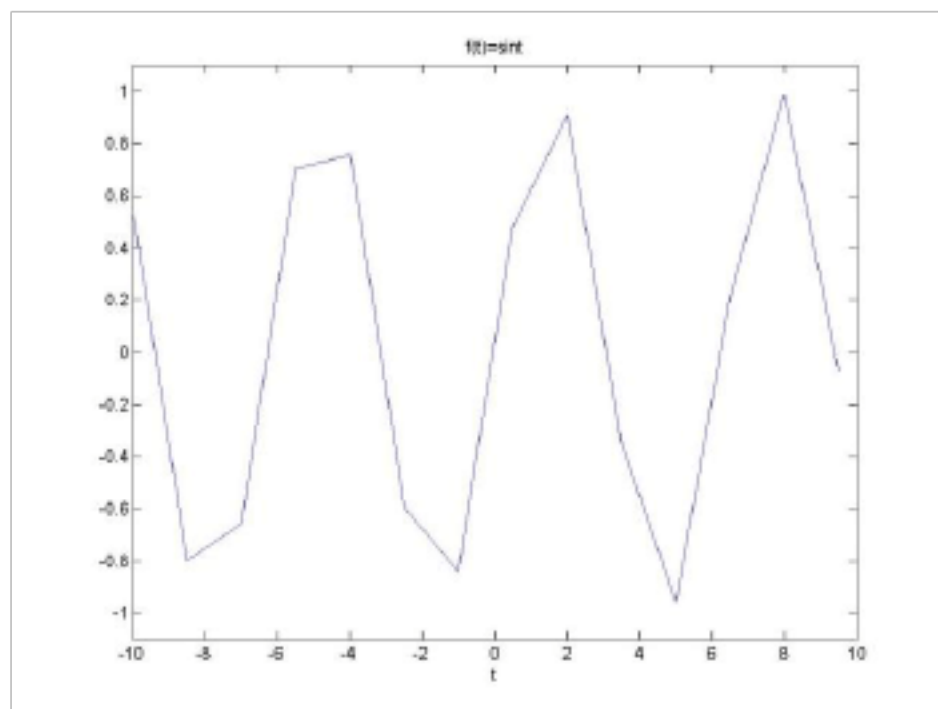
对于连续时间信号 $f(t)$ ，我们可以用两个行向量 f 和 t 来表示，其中向量 t 是行如 $t=t_1:p:t_2$ 的 MATLAB 命令定义的时间范围向量， t_1 为信号起始时间， t_2 为中止时间， p 为时间间隔。向量 f 为连续信号 $f(t)$ 在向量 t 所定义的时间点上的样值。例如对于连续信号 $f(t)=\sin t$ ，我们可以用如下两个向量来表示：

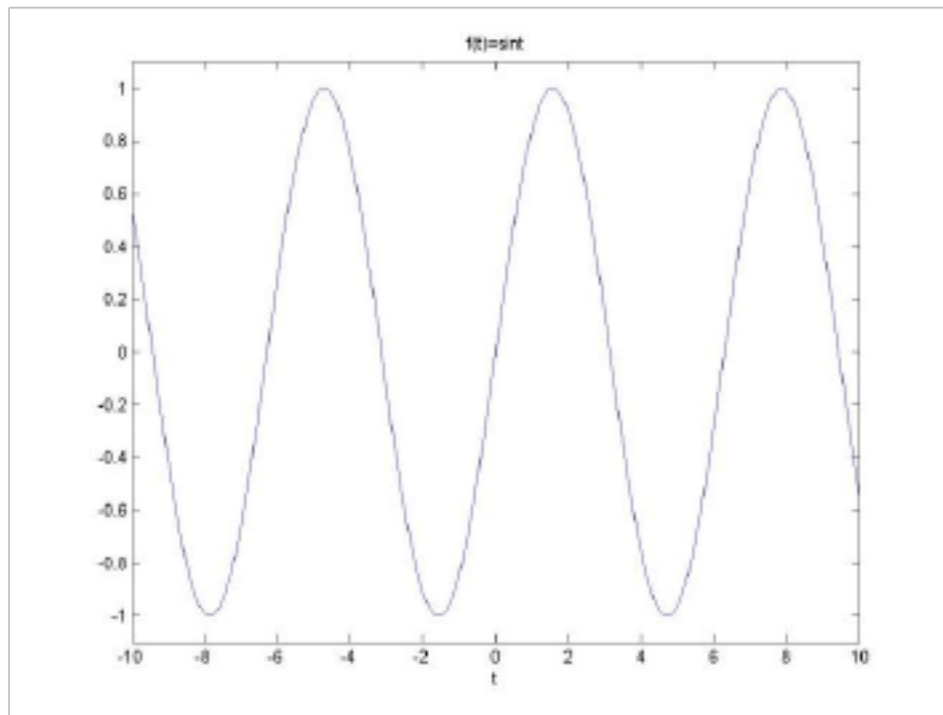
```
t=-10:1.5:10 f=sin(t)
```

用上述向量对连续信号表示后，就可以用 `plot` 命令来绘出该信号的时域波形。`Plot` 命令可将点与点间用直线连接，当点与点间的距离很小时，绘出的图形就成了光滑的曲线。命令如下：

```
plot(t, f)
title('f(t)=sin t')
xlabel('t')
axis([-10, 10, -1.1, 1.1])
```

绘制的信号波形如图 3.1 所示，当把时间间隔 p 取得更小 (如 0.01) 时，就可得到 \sin 较好的近似波形，如图 3.2 所示。



图 3.1 $p=1.$ 的 \sin 近似波形图 3.2 $p=0.0$ 的 \sin 近似波形

(2) 符号运算表示法

如果信号可以用一个符号表达式来表示它，则我们可用 `ezplot`命令绘制出信号的波形。例如对于连续信号 $f(t) = e^{-t/2}$ ，我们可以用符号表达式表示为：

```
syms t
```

```
f=sym('exp(-t/2)')
```

```
f= exp(-t/2)
```

然后用 `ezplot`命令绘制其波形：`ezplot(f, [-6, 6])`

该命令绘制的信号波形如图 3.3 所示

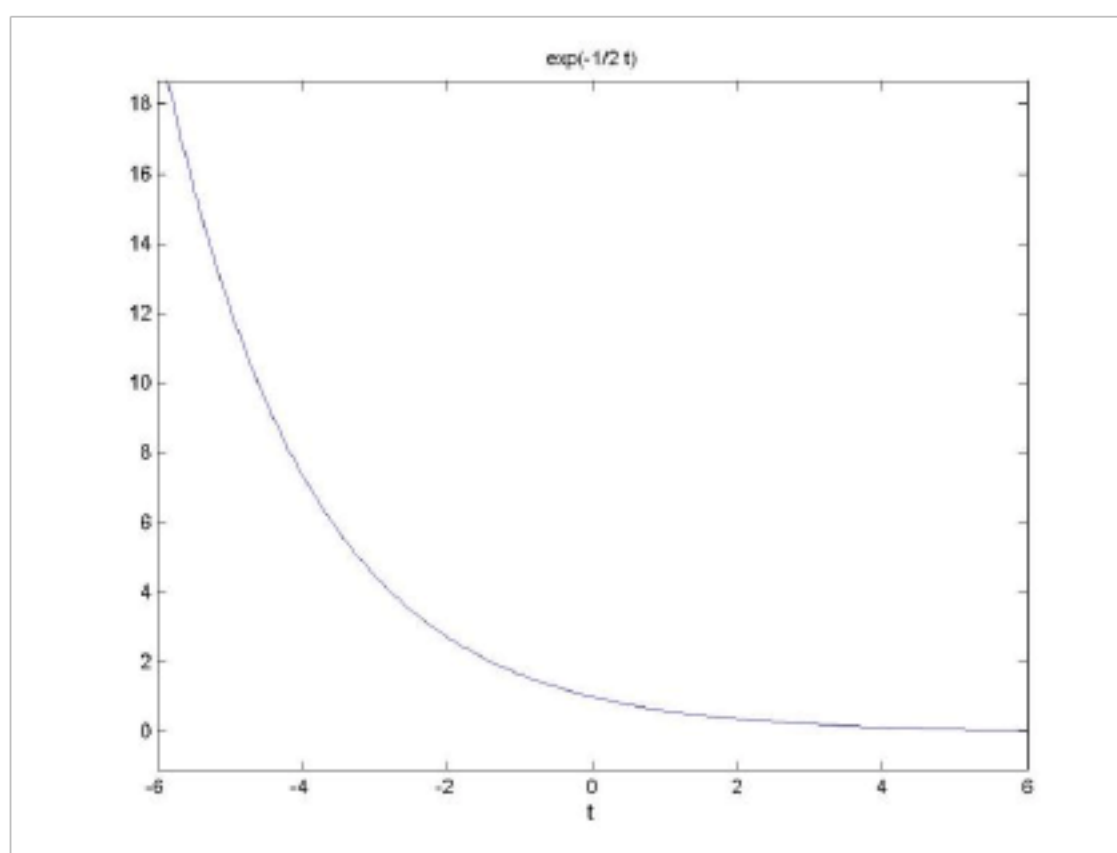


图 3.3 指数信号波形

利用上面两种表达方式我们可以描述出很多种连续信号，常用的信号有：阶跃信号，门信号，斜线信号，指数信号等等。

二、离散时间信号

一般说来，离散时间信号用 $f(k)$ 表示，其中变量 k 为整数，代表离散的采样时间点。

$f(k)$ 可表示为：

$$f(k) = \{ \dots, f(-2), f(-1), f(0), f(1), f(2), \dots \}$$

↑ $k=0$

在 MATLAB 中，用一个向量 f 即可表示一个有限长度的序列。但是，这样的向量并没有包含其对应的时间序号信息。所以，要完整地表示一个离散信号需要用两个向量。

如序列： $f(k) = \{1, 2, -1, 3, 2, 4, -1\}$

↑ $k=0$

在 MATLAB 中应表示为：

$$k = [-3, -2, -1, 0, 1, \text{或是}] k = -3: 3; f = [1, 2, -1, 3, 2, 4, -1]$$

在用 MATLAB 表示离散序列并将其可视化时，我们要注意以下几点：第一，与连续时间信号不同，离散时间信号无法用符号运算来表示；第二，由于在 MATLAB 中，矩阵的元素个数是有限的，因此，MATLAB 无法表示无限序列；第三，在绘制离散信号波形时，要使用专门绘制离散数据的 stem 命令，而不是 plot 命令。如对于上面定义的二向量 f 和 k ，可用如下 stem 命令绘图：stem (k, f) 得到对应序列波形图，如图 3.4 所示。

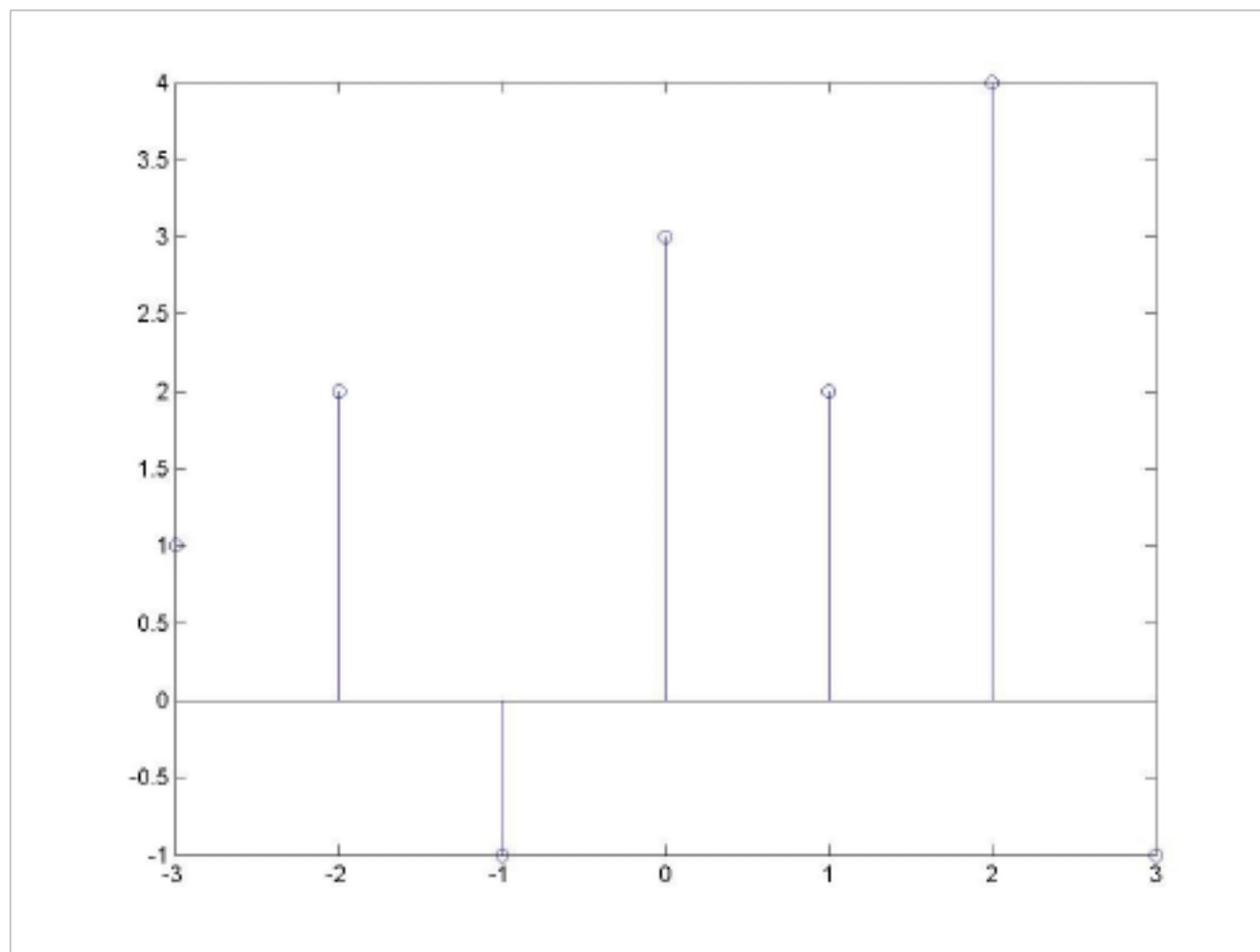


图 3.4 随机序列的波形

同样的，单位阶跃序列，正弦序列，离散时间指数序列等的离散信号我们都可以用类似的方法描述并绘制出图形。

本节使用了 `plot`、`stem` 和 `ezplot` 这几个绘图命令 (显示连续信号中的不连续点用 `stairs` 命令绘图)，在绘制连续信号时要得到光滑曲线用 `plot` 命令；绘制离散信号则要用 `stem` 命令；绘制用 MATLAB 符号表达式表示的信号用 `ezplot` 命令。

三、信号的时域变换

信号的时域变换包括信号的平移、反折、倒相及信号的尺度变换。我们就分别介绍连续时间信号和离散时间信号的各种时域变换^[11]。

1、连续信号的时域变换

如前所述，MATLAB 可以有两种方法来表示连续信号。用这两种方法均可实现连续信号的时域变换，但用符号运算的方法则较为简便。

(1) 移位

对于连续信号 $f(t)$ 若有常数 $t_0 > 0$ ，延时信号 $f(t-t_0)$ 是将原信号沿正 t 轴方向平移时间 t_0 ，而 $f(t+t_0)$ 是将原信号沿负 t 轴方向移动时间 t_0 。我们可用下面的命令来实现连续信号的平移及其结果可视化，其中 f 是用符号表达式表示的连续时间信号， t 是符号变量，`subs` 命令则将连续信号中的时间变量 t 用 $t-t_0$ 替换：

```
y=subs(f,t,t-t0);ezplot(y)
```

(2) 反折

连续信号的反折，是指将信号以纵坐标为轴反折，即将信号 $f(t)$ 中的自变量 t 换为 $-t$ 。实现如下：

```
y=subs(f,t,-t);ezplot(y)
```

(3) 尺度变换

连续信号的尺度变换，是指将信号的横坐标进行展宽或压缩变换，即将信号 $f(t)$ 中的自变量 t 换为 at ，当 $a>1$ 时，信号 $f(at)$ 以原点为基准，沿横轴压缩到原来的 $1/a$ ；当 $0<a<1$ 时，就展宽至原来的 $1/a$ 倍。实现如下：

```
y=subs(f,t,*t);ezplot(y)
```

(4) 倒相

连续信号的倒相是指将信号 $f(t)$ 以横轴为对称轴对折得到 $-f(t)$ 。实现如下：

```
y=-f; ezplot(y)
```

对于以上的命令，可在画图命令之后加入坐标轴的调整的命令(即加入 `axis()` 命令)，以使画出的图形更清晰、直观。

2、离散时间序列的时域变换

与连续信号不同的是，在 MATLAB 中，离散序列的时域变换不能用符号运算来实现，而必须用向量表示的方法，即在 MATLAB 中离散序列的变换需表示成两个向量的变换。

(1) 离散序列反折

离散序列的反折，即是将表示离散序列的两向量以零时刻的取值为基准点，以纵轴为对称轴反折，向量的反折可用 MATLAB 中的 `fliplr` 函数来实现，具体实现如下：

```
function [f,k]=lsfz(f1,k1)
```

```
f=fliplr(f1);k=-fliplr(k1) 调用此函数实现向量 f1和 k1 的反折
```

```
stem(k, ffillé)
```

```
axis([min(k)-1,max(k)+1,min(f)-0.5,max(f)+0.5])
```

(2) 离散序列的平移

离散序列的平移可看作是将离散序列的时间序号向量平移,而表示对应时间序号点的序列样值不变,当序列向左移动 k_0 个单位时,所以时间序号向量都减小 k_0 个单位,反之则增加 k_0 个单位。实现如下:

```
function [f, k]=lsyw(ff, kk, k0)
k=kk+k0; f=ff;
stem(k, f, 'filled')
axis([min(k)-1, max(k)+1, min(f)-0.5, max(f)+0.5])
```

(3) 离散序列的倒相

离散序列的倒相可看作是将表示序列样值的向量取反,而对应的时间序号向量不变,得到的离散时间序列。实现如下:

```
function [f, k]=lsdx(ff, kk)
f=-ff;
k=kk;
stem(k, f, 'filled')
axis([min(k)-1, max(k)+1, min(f)-0.5, max(f)+0.5])
```

这些时域变换,我们可以把我们在第一节中描述过的信号带入其中,来看看信号时域变换的结果如何。除此以外,我们通过时域变换也可以锻炼我们描述信号的能力,一些复杂信号,往往都是一些简单信号经过一系列的时域变换得到。

四、实验设备

计算机 MATLAB 软件

五、实验内容:

1、在 MATLAB 中连续信号的向量或符号运算功能来表示

严格意义上来讲, MATLAB 并不能处理连续信号,在 MATLAB 中,是用连续信号在等时间间隔点的样值来近似地表示连续信号的,当取样时间间隔足够小时,这些离散的样值就能较好地近似出连续信号。

(1) 向量表示法

对于连续时间信号 $f(t)$ ，我们可以用两个行向量 f 和 t 来表示，其中向量 t 是行如 $t=t_1:p:t_2$ 的 MATLAB 命令定义的时间范围向量， t_1 为信号起始时间， t_2 为中止时间， p 为时间间隔。向量 f 为连续信号 $f(t)$ 在向量 t 所定义的时间点上的样值，例如对于连续信号 $f(t)=\sin t$ ，我们可以用如下两个向量来表示：

```
t=-10:1.5:10 f=sin(t)
```

用上述向量对连续信号表示后，就可以用 `plot` 命令来绘出该信号的时域波形。`Plot` 命令可将点与点间用直线连接，当点与点间的距离很小时，绘出的图形就成了光滑的曲线。命令如下：

```
plot(t, f)
title('f(t)=sint')
xlabel('t')
axis([-10, 10, -1.1, 1.1])
```

(2) 符号运算表示法

如果信号可以用一个符号表达式来表示它，则我们可用 `ezplot` 命令绘制出信号的波形。例如对于连续信号 $f(t)=e^{-t/2}$ ，我们可以用符号表达式表示为：

```
syms t
f=sym('exp(-t/2)')
f= exp(-t/2)
```

然后用 `ezplot` 命令绘制其波形：`ezplot(f, [-6, 6])`

2、在 MATLAB 中离散时间信号描述：

一般说来，离散时间信号用 $f(k)$ 表示，其中变量 k 为整数，代表离散的采样时间点。

$f(k)$ 可表示为：

$$f(k) = \{ \dots, f(-2), f(-1), f(0), f(1), f(2), \dots \}$$

↑ $k=0$

在 MATLAB 中，用一个向量 f 即可表示一个有限长度的序列。但是，这样的向量并没有包含其对应的时间序号信息。所以，要完整地表示一个离散信号需要用两个向量

，如序列： $f(k) = \{1, 2, -1, 3, 2, 4, -1\}$
 $\uparrow k=0$

在 MATLAB 中应表示为：

$k = [-3, -2, -1, 0, 1, 2, 3]$ 或是 $k = -3: 3$; $f = [1, 2, -1, 3, 2, 4, -1]$

在用 MATLAB 表示离散序列并将其可视化时，我们要注意以下几点：

第一，与连续时间信号不同，离散时间信号无法用符号运算来表示；

第二，由于在 MATLAB 中，矩阵的元素个数是有限的，因此，MATLAB 无法表示无限序列；第三，在绘制离散信号波形时，要使用专门绘制离散数据的 stem 命令，而不是 plot 命令

3、单位阶跃序列，正弦序列，离散时间指数序列等的离散信号我们都可以用类似的方法描述并绘制出图形：

指数信号 Aeat

```
y=A*exp(a*t)
取 A =1, a=-0.4
A=1;a=-0.4;
t=0:0.01:10;
ft= A*exp(a*t);
plot(t, ft)
```

正弦信号

```
y=A*cos(w0*t+phi)
y=A*sin(w0*t+phi)
```

抽样函数

```
y=sinc(t)
```

矩形脉冲信号

```
y=rectpuls(t, width)
```

产生幅度为 1，宽度为 width 以 $t=0$ 为对称的矩形波，Width 默认值为 1

三角波脉冲信号

```
y=tripuls(t, width, skew)
```

指数序列 Aa^k

```
A*a.^k
K=0:10;A=1;a=-1.6;
fk= A*a.^k;
stem(k, fk)
```

单位脉冲序列

零矩阵函数 `zeros(1, N)` 产生一个由 N 个零组成的列向量，

```
k=-50:50;
delta=[zero(1, 50), 1, (1, 50)];
stem(k, delta)
```

单位阶跃序列

...0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 ...

零矩阵函数 `zeros(1, N)` 产生一个由 N 个零组成的列向量，`ones(1, N)` 产生一个由 N 个 1 组成的列向量

```
k=-50:50;
uk=[zero(1, 50), ones(1, 51)];
stem(k, uk)
```

信号的时域变换包括信号的平移、反折、倒相及信号的尺度变换。我们就分别介绍连续时间信号和离散时间信号的各种时域变换

4、连续信号的时域变换

移位

对于连续信号 $f(t)$ 若有常数 $t_0 > 0$ ，延时信号 $f(t-t_0)$ 将原信号沿正 t 轴方向平移时间 t_0 ，而 $f(t+t_0)$ 将原信号沿负 t 轴方向移动时间 t_0 。我们可用下面的命令来实现连续信号的平移及其结果可视化，其中 f 是用符号表达式表示的连续时间信号， t 是符号变量，`subs` 命令则将连续信号中的时间变量 t 用 $t-t_0$ 替换：

```
y=subs(f, t, t-t0); plot(y)
```

反折

连续信号的反折，是指将信号以纵坐标为轴反折，即将信号 $f(t)$ 中的自变量 t 换为 $-t$ 。实现如下：

```
y=subs(f,t,-t);ezplot(y)
```

3. 尺度变换

连续信号的尺度变换，是指将信号的横坐标进行展宽或压缩变换，即将信号 $f(t)$ 中的自变量 t 换为 at ，当 $a>1$ 时，信号 $f(at)$ 以原点为基准，沿横轴压缩到原来的 $1/a$ ；当 $0<a<1$ 时，就展宽至原来的 $1/a$ 倍。实现如下：

```
y=subs(f,t,*t);ezplot(y)
```

4. 倒相

连续信号的倒相是指将信号 $f(t)$ 以横轴为对称轴对折得到 $-f(t)$ 。实现如下：

```
y=-f;ezplot(y)
```

对于以上的命令，可在画图命令之后加入坐标轴的调整的命令（即加入 `axis` 命令），以使画出的图形更清晰、直观。

与连续信号不同的是，在 MATLAB 中，离散序列的时域变换不能用符号运算来实现，而必须用向量表示的方法，即在 MATLAB 中离散序列的变换需表示成两个向量的变换。

离散信号的反折，即是将表示离散序列的两向量以零时刻的取值为基准点，以纵轴为对称轴反折，向量的反折可用 MATLAB 中的 `fliplr` 函数来实现，具体实现如下：

```
function [f,k]=lsfz(f1,k1)
f=fliplr(f1);k=-fliplr(k1);%调用此函数实现向量 f1和 k1 的反折
stem(k,f,'filled')
axis([min(k)-1,max(k)+1,min(f)-0.5,max(f)+0.5])
```

离散序列的平移可看作是将离散序列的时间序号向量平移，而表示对应时间序号点的序列样值不变，当序列向左移动 k_0 个单位时，所以时间序号向量都减小 k_0 个单位，反之则增加 k_0 个单位。实现如下：

```
function [f,k]=lsyw(ff,kk,k0)
```

```

k=kk+k0;f=ff;
stem(k, f, ' filled' )
axis([min(k)-1, max(k)+1, min(f)-0.5, max(f)+0.5])

```

离散序列的倒相可看作是将表示序列样值的向量取反，而对应的时间序号向量不变，得到的离散时间序列。实现如下：

```

function [f, k]=lsdx(ff, kk)
f=-ff;
k=kk;
stem(k, f, ' filled' )
axis([min(k)-1, max(k)+1, min(f)-0.5, max(f)+0.5])

```

4、离散时间序列的时域变换

与连续信号不同的是，在 MATLAB 中，离散序列的时域变换不能用符号运算来实现，而必须用向量表示的方法，即在 MATLAB 中离散序列的变换需表示成两个向量的变换。

离散序列的反折，即是将表示离散序列的两向量以零时刻的取值为基准点，以纵轴为对称轴反折，向量的反折可用 MATLAB 中的 `fliplr` 函数来实现，具体实现如下：

```

function [f, k]=lsfz(f1, k1)
f=fliplr(f1);k=-fliplr(k1)调用此函数实现向量 f1和 k1 的反折
stem(k, f, ' filled' )
axis([min(k)-1, max(k)+1, min(f)-0.5, max(f)+0.5])

```

离散序列的平移可看作是将离散序列的时间序号向量平移，而表示对应时间序号点的序列样值不变，当序列向左移动 k_0 个单位时，所以时间序号向量都减小 k_0 个单位，反之则增加 k_0 个单位。实现如下：

```

function [f, k]=lsyw(ff, kk, k0)
k=kk+k0;f=ff;
stem(k, f, ' filled' )
axis([min(k)-1, max(k)+1, min(f)-0.5, max(f)+0.5])

```

离散序列的倒相可看作是将表示序列样值的向量取反,而对应的时间序号向量不变,得到的离散时间序列。实现如下:

```
function [f, k]=lsdx(ff, kk)
f=-ff;
k=kk;
stem(k, f, 'filled' )
axis([min(k)-1, max(k)+1, min(f)-0.5, max(f)+0.5])
```

实验三 信号卷积的 MATLAB 实现

一、实验名称：

信号卷积的 MATLAB 实现

二、实验目的：

- 1.增加学生对卷积的认识
- 2.了解 MATLAB 这个软件的一些基础知识
- 3.利用 MATLAB 计算信号卷积
- 4.验证卷积的一些性质

三、实验原理：

用 MATLAB 实现卷积我们先必须从信号下手，先把信号用 MATLAB 语句描述出来，然后再将这些信号带入到我们写好的求卷积的函数当中来计算卷积。在本章中我们将信号分为连续信号和离散序列两种来实现卷积并验证卷积的一些性质。

MATLAB 强大的图形处理功能及符号运算功能，为我们实现信号的可视化提供了强有力的工具。在 MATLAB 中通常有两种方法来表示信号，一种是用向量来表示信号，另一种则是用符号运算的方法来表示信号。用适当的 MATLAB 语句表示出信号后，我们就可以利用 MATLAB 的绘图命令绘制出直观的信号波形。

连续时间信号，是指自变量的取值范围是连续的，且对于一切自变量的取值，除了有若干不连续点以外，信号都有确定的值与之对应的信号。从严格意义上来讲，MATLAB 并不能处理连续信号，在 MATLAB 中，是用连续信号在等时间间隔点的样值来近似地表示连续信号的，当取样时间间隔足够小时，这些离散的样值就能较好地近似出连续信号。在 MATLAB 中连续信号可用向量或符号运算功能来表示。

1. 向量表示法

对于连续时间信号 $f(t)$ ，我们可以用两个行向量 f 和 t 来表示，其中向量 t 是

行如 $t=t_1:p:t_2$ 的 MATLAB 命令定义的时间范围向量, t_1 为信号起始时间,

t_2 为中止时间， p 为时间间隔。向量 f 为连续信号 $f(t)$ 在向量 t 所定义的时间点上的样值。例如对于连续信号 $f(t)=\sin t$ ，我们可以用如下两个向量来表示：

```
t=-10:1.5:10 f=sin(t)
```

用上述向量对连续信号表示后，就可以用 `plot` 命令来绘出该信号的时域波形。`Plot` 命令可将点与点间用直线连接，当点与点间的距离很小时，绘出的图形就成了光滑的曲线。命令如下：

```
plot(t, f)
title('f(t)=sin t')
xlabel('t')
axis([-10, 10, -1.1, 1.1])
```

绘制的信号波形如图 3.1 所示，当把时间间隔 p 取得更小 (如 0.01) 时，就可得到 \sin 较好的近似波形，如图 3.2 所示。

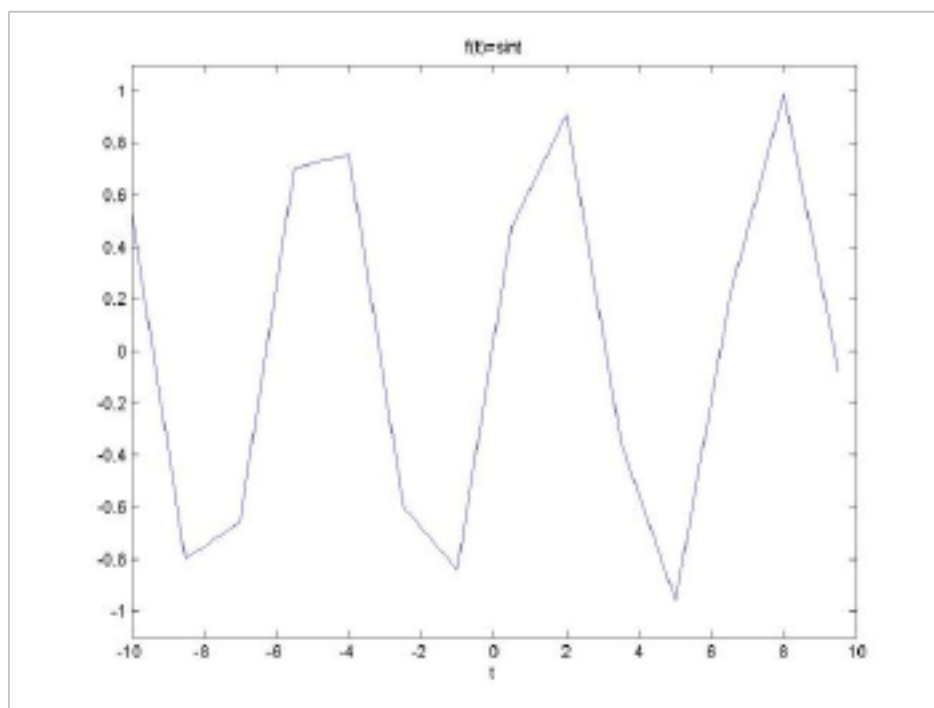


图 3.1

$p=1.5$ 的 \sin 近似波形

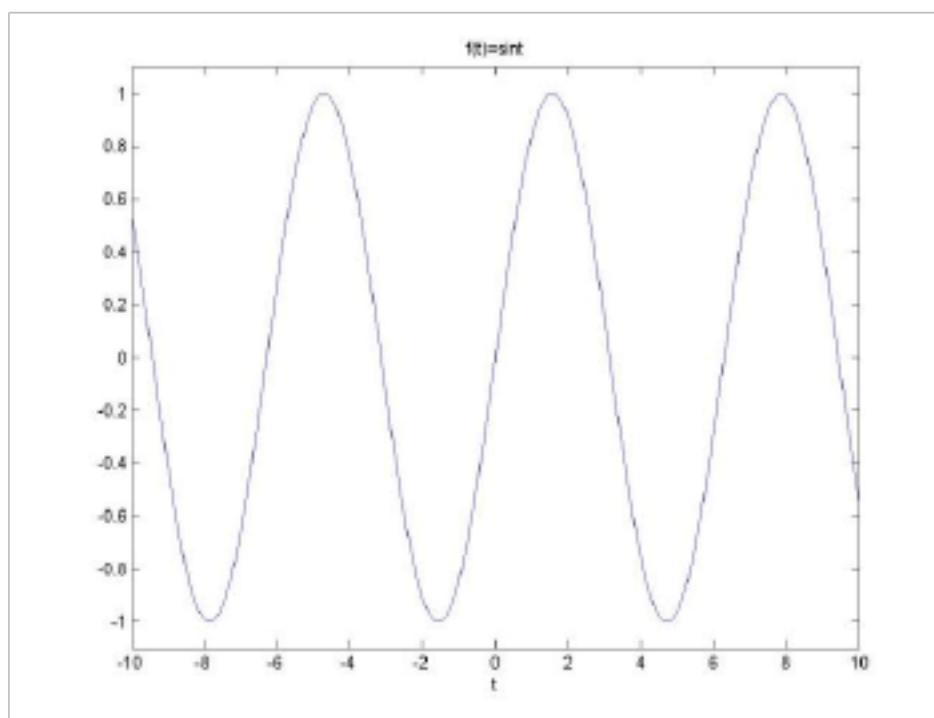


图 3.2 $p=0.0$ 的 \sin 近似波形

2. 符号运算表示法

如果信号可以用一个符号表达式来表示它，则我们可用 `ezplot` 命令绘制出信号的波形。例如对于连续信号 $f(t) = e^{-t/2}$ ，我们可以用符号表达式表示为：

```
syms t
```

```
f=sym('exp(-t/2)')
```

```
f=exp(-t/2)
```

然后用 `ezplot` 命令绘制其波形：`ezplot(f, [-6, 6])`

该命令绘制的信号波形如图 3.3 所示

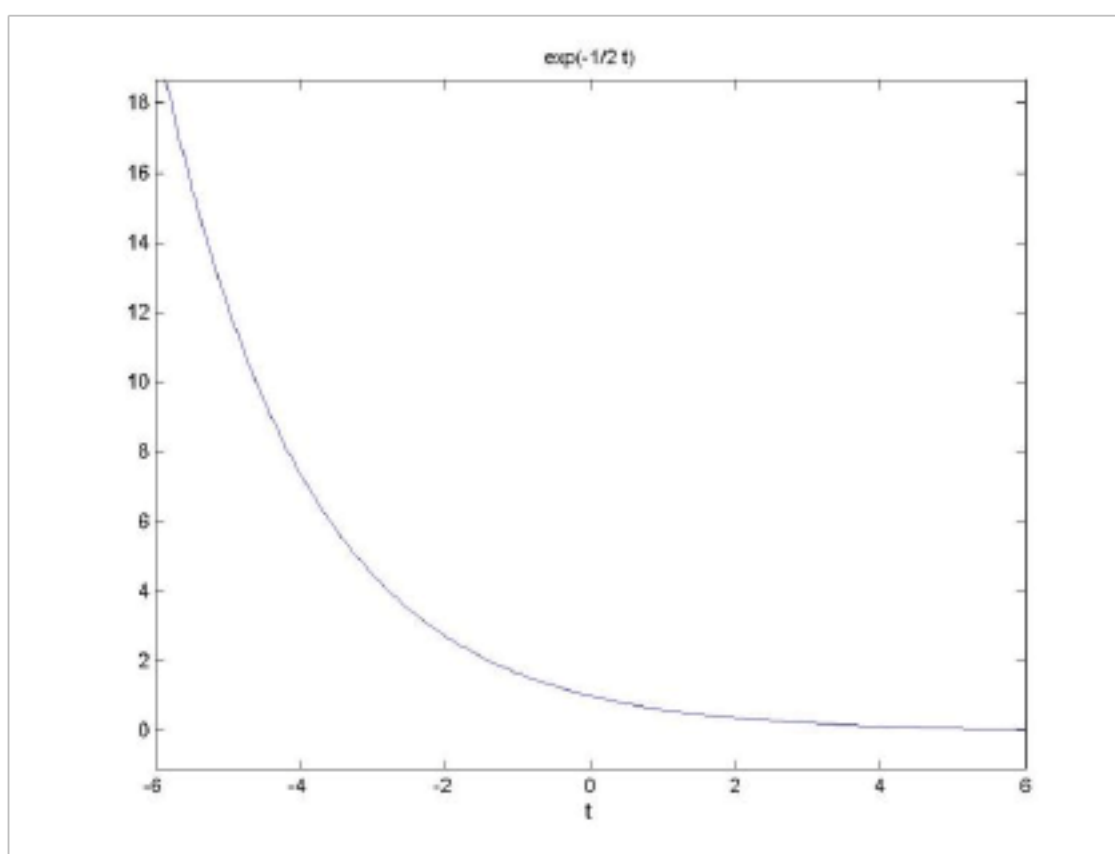


图 3.3 指数信号波形

利用上面两种表达方式我们可以描述出很多种连续信号，常用的信号有：阶跃信号，门信号，斜线信号，指数信号等等。

接下来我们看看离散时间信号，一般说来，离散时间信号用 $f(k)$ 表示，其中变量 k 为整数，代表离散的采样时间点。

$f(k)$ 可表示为：

$$f(k) = \{ \dots, f(-2), f(-1), f(0), f(1), f(2), \dots \}$$

$$\uparrow k=0$$

在 MATLAB 中，用一个向量 f 即可表示一个有限长度的序列。但是，这样的向量并没有包含其对应的时间序号信息。所以，要完整地表示一个离散信号需要用两个向量。

如序列： $f(k) = \{1, 2, -1, 3, 2, 4, -1\}$

↑ $k=0$

在 MATLAB 中应表示为：

$k = [-3, -2, -1, 0, 1, 2, 3]$ 或 $k = -3: 3$; $f = [1, 2, -1, 3, 2, 4, -1]$

在用 MATLAB 表示离散序列并将其可视化时，我们要注意以下几点：第一，与连续时间信号不同，离散时间信号无法用符号运算来表示；第二，由于在 MATLAB 中，矩阵的元素个数是有限的，因此，MATLAB 无法表示无限序列；第三，在绘制离散信号波形时，要使用专门绘制离散数据的 stem 命令，而不是 plot 命令。如对于上面定义的二向量 f 和 k，可用如下 stem 命令绘图：stem(k, f) 得到对应序列波形图，如图 3.4 所示。

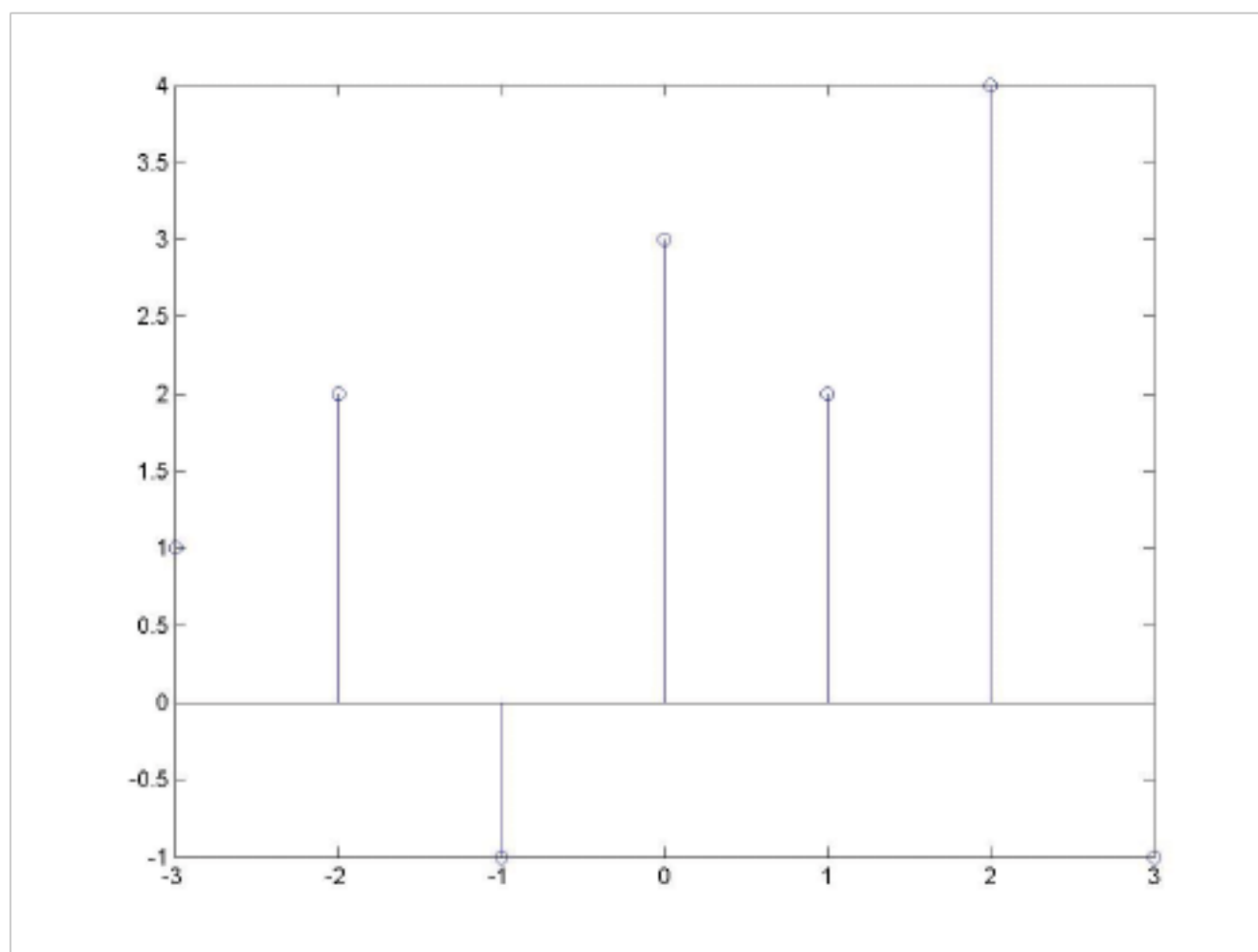


图 3.4 随机序列的波形

同样的，单位阶跃序列，正弦序列，离散时间指数序列等的离散信号我们都可以用类似的方法描述并绘制出图形。

信号的时域变换包括信号的平移、反折、倒相及信号的尺度变换。我们就分别介绍连续时间信号和离散时间信号的各种时域变换。

1、连续信号的时域变换

如前所述，MATLAB 可以有两种方法来表示连续信号。用这两种方法均可实现连续信号的时域变换，但用符号运算的方法则较为简便。

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：<https://d.book118.com/197056164065010002>