

Matlab 平台的使用

1. 总论

Matlab 是信号处理中功能强大且广泛使用的工具。本实验的主要目标是使你们熟悉使用 Matlab。我们假定你们有过其他某种语言的编程经验，应该已经了解变量，循环和函数等基本概念。这里介绍一下 Matlab 的基本编程结构。

本实验的三个目标：

1. 学习基本的 Matlab 命令和语法，包括“帮助”系统
2. 学习编写你自己的 Matlab 的脚本文件，并像命令一样运行它们
3. 学习一点 Matlab 的高级编程技术，例如向量化

2. 初识 Matlab

1. 启动 Matlab，运行 Matlab 的帮助。在命令窗口输入“doc”，或在菜单中选“help”。

2. 在命令窗口中，试试：

```
helpwin
helpwin plot
helpwin colon %<--- a VERY IMPORTANT notation
helpwin ops
helpwin zeros
helpwin ones
lookfor filter %<--- keyword search
```

3. 运行 Matlab 的 demo：在命令窗口输入“demo”

4. 把 Matlab 当计算器用，试试：

```
pi*pi - 10
sin(pi/4)
ans ^ 2 %<--- "ans" holds the last result
```

5. 在 Matlab 中给变量赋值，试试：

```
xx = sin( pi/5 );
cos( pi/5 ) %<--- assigned to what?
yy = sqrt( 1 - xx*xx )
ans
```

注意观察语句末尾有分号和无分号有什么差别。

6. Matlab 中的复数运算的基本操作，试试：

```
zz = 3 + 4i, ww = -3 + 4j
```

```

real(zz), imag(zz)
abs([zz, ww]) %<-- Vector constructor
conj(zz+ww)
angle(zz)
exp( j*pi )
exp(j*[ pi/4, 0, -pi/4 ])

```

3. 再识 Matlab

3.1 Matlab 数组索引

(a). 冒号的含义

试试如下命令的结果：

```

jkl = 0 : 6
jkl = 2 : 4 : 17
jkl = 99 : -1 : 88
ttt = 2 : (1/9) : 4
tpi = pi * [ 0:0.1:2 ]; %注意这里有分号

```

(b). 从向量提取数字和 / 或插入数字，思考如下xx的定义：

```

xx = [ zeros(1,3), linspace(0,1,5), ones(1,4) ]
xx(4:6)
size(xx)
length(xx)
xx(2:2:length(xx))
xx(2:2:end))

```

解释以上代码最后4行的结果

(c). 观察如下赋值的结果：

```
yy = xx; yy(4:6) = pi*(1:3)
```

现在写一条语句，xx用(b)中定义的方法，把xx的偶数索引的元素(即xx(2), xx(4)等)的值替换为常数 π 。使用向量替换，不要用循环。

3.2 Matlab 脚本文件

(a)看看关于向量的实验。把向量看作一个数字集合，试试：

```
xk = cos( pi*(0:11)/4 ) %<---comment: compute cosines
```

解释一下余弦函数的值是怎样存储在向量xk中的。xk(1)是什么意思？有定义xk(0)吗？

注：语句后没有分号，所以语句会把xk的计算结果依次显示在命令窗口，在百分号%后面的文字是注释，可以忽略。

(b) Matlab中可以写循环，但是这不是最有效率的解决办法。尽量避免循环，并使用冒号会提高效率。下面的代码是用一个循环来计算余弦函数的值的：

```
yy = [ ]; %<--- initialize the yy vector to be empty
for k=-5:5
yy(k+6) = cos( k*pi/3 )
end
yy
```

解释为什么需要写yy(k+6)。如果用yy(k)会发生什么？

不使用循环，而用3.1的方法，利用冒号，重新实现这个功能。

(c) Matlab中实数和复数的绘制都很容易。基本绘图命令绘制向量yy和向量xx的关系，试试：

```
xx = [-3 -1 0 1 3];
yy = xx.*xx - 3*xx;
plot( xx, yy )
zz = xx + yy*sqrt(-1)
plot( zz ) %<---- complex values: plot imag vs. real
```

在命令窗口输入“helpwin arith”，了解当xx是一个向量时，操作xx.*xx是如何工作的。当xx是一个矩阵时，操作xx.*xx又是如何工作的。

当你没把握某个命令时，可随时使用帮助系统。

(d) 使用Matlab的编译器创建脚本文件，命名为mylab1.m，文件内容如下：

```
clear all; close all;
tt = -1 : 0.01 : 1;
xx = cos(5*pi*tt);
zz = 1.4*exp(j*pi/2)*exp(j*5*pi*tt);
plot(tt, xx, 'b-', tt, real(zz), 'r--') %<--- plot a sinusoid
grid on
title(' TEST PLOT of a SINUSOID')
xlabel(' TIME (sec)')
```

解释为什么real(zz)的图形是一个正弦曲线。它的相位和幅度是多少？由所绘图形的时移量计算相位。

(e) 在命令窗口输入“mylab1”即可运行上述m文件，即

```
mylab1 %<---will run the commands in the file
type mylab1 %<---will type out the contents of
% mylab1.m to the screen
```

3.3 Matlab 与声音相关的函数命令

这一节练习是关于声音信号的，所以需要带耳机来机房听效果。

(a) 运行Matlab的声音demo：在命令窗口输入“xpsound”。你可以听到多种声音，并可以查看对应的时间序列，功率谱密度和声谱图（声谱图详见7.3）。

(b) 现在在Matlab中生成一个音调（例如正弦），用soundsc()命令¹听一听在3.2 (d) 的第2行和第3行创建了一个向量xx为2.5Hz的正弦序列。而你们要生成的正弦信号的频率为2000Hz，时长为0.9秒，采样率fs为11025个样本 / 秒。采

¹ soundsc(xx,fs)函数需用两个参数，xx是播放文件的数据的向量，fs是播放样本的采样率。另外，soundsc(xx,fs)会自动把xx的值尺度变化成-1到1之间，然后调用sound(xx,fs)播放信号。

样率说明了采样点之间的时间间隔，所以时间向量应定义为：

`tt = 0:(1/fs):dur;`

其中fs为采样率，dur为时长（以秒为单位）。可以在帮助系统中查找sound()和soundsc()了解它们的用法和含义。那么，你们的tt向量的长度（即样本数）是多少？

4. 用 Matlab 处理正弦信号

在实验报告中要包括本节的图形和实验结果描述。写一个Matlab的m文件来完成步骤（a）到步骤（d）。你们的实验报告也要包括这个m文件的代码。

（a）生成一个时间向量tt，需要覆盖下面的（b）中频率为4000Hz的正弦信号的两个周期。tt的定义方式使用3.2（d）的方式。如果我们使用T表示正弦的周期，定义向量tt的开始时间为-T，结束时间为+T，那两个周期就会包括tt=0。最后，确保你们正弦波的每个周期有至少25个样本。也就是说，当你们使用冒号操作符定义时间变量时，要使增量足够小，才能保证每个周期可以产生至少25个样本。

（b）生成以下两个4000Hz的正弦序列：

$$x1(t) = A1 \cos(2\pi(4000)(t - tm1)) \quad x2(t) = A2 \cos(2\pi(4000)(t - tm2))$$

其中A1是你的年龄，A2=1.2A1，tm1 = (37.2/M)T，tm2 = -(41.3/D)T，这里M和D分别是你生日的月和日，T是周期

绘制以上两个信号在 $-T \leq t \leq T$ 的图形。为了输出下面（d）的要求，使用subplot(3,1,1)和subplot(3,1,2)来使图在一个图形窗口内，详见helpwin subplot

一般用法是：

`subplot(2,1,2);`

`plot(xx);`

（c）创建第3个正弦信号为 $x3(t) = x1(t) + x2(t)$ 。在Matlab中，这表示把两个正弦向量中的值对应相加。绘制 $x3(t)$ 在 $-T \leq t \leq T$ 的图形，使用subplot(3,1,3)绘图。

（d）对每个图都添加一个图名，图名都要包含你的姓名，用title函数

详见 helpwin title, helpwin subplot

4.1 复数幅度

用一条语句生成以上正弦信号x1(t) 的值，可以使用复数幅度表示方法：

$$x1(t) = \text{Re}\{Xe^{j\omega t}\}$$

其中X和w为某个常数。

5. 复指数简介

本节目标是让大家熟悉复数，并了解怎么使用它们表示正弦信号(例如 $x(t) = A\cos(\omega t + \phi)$) 为复指数形式(例如 $z(t) = Ae^{j\omega t}$)，方法如下所示：

$$x(t) = A\cos(\omega t + \phi) = \text{Re}\{ Ae^{j\omega t} e^{j\phi} \}$$

5.2 Matlab 中的复数

Matlab可用于计算复值式子，并可以向量显示结果，以下是一些Matlab的复数操作符：

conj—复共轭

abs—幅度

angle—以弧度为单位的角度或相位

real - 实部

imag—虚部

i, j - 已预定义为 $\sqrt{-1}$

$x = 3 + 4i$

exp(j*theta) - 复指数函数 $e^{j\theta}$

每个函数都以向量或矩阵作为输入参量。

注意：Matlab中没有 mag() 和 phase() 函数。

5.3 向量化

Matlab最重要的部分就是矩阵—向量语法。大多数情况下，循环都可以用向量操作代替，因为形如exp() 和cos() 等函数都被定义为向量输入，例如

$$\begin{aligned} \cos(vv) &= [\cos(vv(1)), \cos(vv(2)), \cos(vv(3)), \dots \\ &\quad \cos(vv(N))] \end{aligned}$$

其中vv是N个元素的行向量。向量化可用于简化代码。如果有如下代码来绘制某个信号：

```
M = 200;
for k=1:M
    x(k) = k;
    y(k) = cos( 0.001*pi*x(k)*x(k) );
end
plot( x, y, 'ro-' )
```

你可以把循环替换掉，而只用如下3行代码即可：

```
M = 200;
```

```
y = cos( 0.001*pi*(1:M).*(1:M) );
plot( 1:M, y, 'ro-' )
```

用这种向量化的方法编写2—3行代码完成以下Matlab代码，不使用循环。（注：当xx是向量时，xx*xx和xx.*xx是不同的）

```
%--- make a plot of a weird signal
N = 200;
for k=1:N
    xk(k) = k/50;
    rk(k) = sqrt( xk(k)*xk(k) + 2.25 );
    sig(k) = exp(j*2*pi*rk(k));
end
plot( xk, real(sig), 'mo-' )
```

5.4 函数

函数是一种特别的M-文件，可以接受输入（矩阵或向量），也可以得到输出。关键词function必须是文件的第一个词，并且定义了函数，M-文件的第一行定义函数如何传递输入和输出参量。文件扩展名必须使用小写的“m”，例如my_func.m。

如下函数有一些小错误，请找找错在哪里：（至少有3处错误）

```
matlab mfile [xx,tt] = badcos(ff,dur)
%BADCOS Function to generate a cosine wave
% usage:
% xx = badcos(ff,dur)
% ff = desired frequency in Hz
% dur = duration of the waveform in seconds
%
tt = 0:1/(100*ff):dur; %-- gives 100 samples per period
badcos = cos(2*pi*freeq*tt);
```

错误有：第一个词必须是“function”。而且“freeq”在使用前没有定义。最后，第一行指明要“xx”作输出，因此“xx”需要在函数体中出现在至少一个赋值行的左边。

正确的函数写法应该是：

```
function [xx,tt] = goodcos(ff,dur)
tt = 0:1/(100*ff):dur; %-- gives 100 samples per period
xx = cos(2*pi*ff*tt);
```

可在命令窗口输入函数的文件名（注意不是函数名）来调用它（也可以在另一个m文件调用它），例如，如果以上goodcos函数的文件名为testcos.m，且ff=1，dur=10，则调用时应输入testcos(1,10)。你可以尝试一下，如果你输入goodcos(1,10)，matlab会有什么反应。

6. 复指数

6.1 生成正弦信号的 M-文件

写一个可以生成单一正弦信号 $x(t) = A\cos(\omega t + \phi)$ 的函数，使用4个输入参量：幅度 A ，频率 ω ，相位 ϕ 和时长 dur 。函数应当返回两个输出参量：正弦信号的值 x 和对应的时间 t 。确保函数生成的正弦信号在每个周期有20个值，函数名为`one_cos()`。提示：可借鉴上面的`goodcos()`函数。

绘制你们的`one_cos()`函数，参数选为： $A=95$ ， $\omega=200\pi$ 弧度/秒， $\phi=\pi/5$ 弧度，时长为0.025秒。推导所绘图形的周期和相位是否正确。如果周期以毫秒为单位是多少？

7 线性调频脉冲 chirp

线性调频脉冲chirp信号（或线性扫频信号）是一个正弦信号，其频率从一个初值线性变化到终值。Chirp信号的表达式可写为：

$$x(t) = \cos(\psi(t)), \quad \text{其中 } \psi(t) = 2\pi \mu t^2 + 2\pi f_0 t + \phi$$

对 $\psi(t)$ 求导后得到随时间线性变化的瞬时频率：

$$f_i(t) = 2\mu t + f_0$$

$f_i(t)$ 的斜率等于 2μ ，截距为 f_0 。如果信号于 $t=0$ 开始， f_0 也是初始频率。这种时变角度造成的频率变化称为频率调制。这类信号是频率调制FM信号的一个例子。更一般的，我们通常考虑它们为更大一类的角度调制信号。最后，因为频率的线性变化可以生成一个类似汽笛声或鸟叫声（chirp），所以线性FM信号也可以称为chirp信号。

7.1 chirp 信号的 Matlab 合成方法

以下的Matlab代码可以合成一个chirp信号

```
fsamp = 11025;
dt = 1/fsamp;
dur = 1.8;
tt = 0 : dt : dur;
psi = 2*pi*(100 + 200*tt + 500*tt.*tt);
xx = real( 7.7*exp(j*psi) );
soundsc( xx, fsamp );
```

(a) 确定合成信号的总时长（秒），确定`tt`向量的长度（样本数）

(b) 在Matlab中，只能合成离散时间信号，所以对于chirp信号，我们处理为：

$$x(t_n) = A \cos(2\pi \mu t_n^2 + 2\pi f_0 t_n + \phi)$$

其中, $t_n = nT_s$ 表示离散时间值。在以上的 Matlab 代码中, t_n 的值是多少? A , f_0 和 ϕ 的值是多少?

(c) 确定以上 Matlab 代码合成的频率的范围 (Hz 为单位)。手绘瞬时频率随时间的变化情况。听到的最小频率和最大频率是多少?

(d) 听听信号的频率是上升还是下降 (使用 `soundsc()`)。注意 `soundsc()` 需要知道信号创建时的采样率。详见帮助。

7.2 chirp 的函数

使用以下代码来写一个可以合成 chirp 信号的函数, 根据注释的提示编写。

```
function [xx,tt] = mychirp( f1, f2, dur, fsamp )
%MYCHIRP generate a linear-FM chirp signal
%
% usage: xx = mychirp( f1, f2, dur, fsamp )
%
% f1 = starting frequency
% f2 = ending frequency
% dur = total time duration
% fsamp = sampling frequency (OPTIONAL: default is 11025)
%
% xx = (vector of) samples of the chirp signal
% tt = vector of time instants for t=0 to t=dur
%
if( nargin < 4 ) %-- Allow optional input argument
    fsamp = 11025;
end
```

请生成一个 chirp, 其频率起始于 2500Hz, 终止于 500Hz。时长应当为 1.5 秒。听听你的 chirp。

7.3 声谱图

考虑信号的频谱非常有用。信号频谱是信号所含频率的表示方式。固定单一频率的正弦信号的频谱包含两个成分, 一个在 $2\pi f_0$, 一个在 $-2\pi f_0$ 。对于更复杂的信号, 频谱会非常有趣, FM 信号的频谱可以认为是时变的。一种表示时变频谱信号的方式就是声谱图。

声谱图可通过估计信号短时的频率成分得到。把频谱在每个部分的幅度画成关于频率和时间的二维的密度或颜色, 即可得到一幅声谱图。

关于声谱图需要知道:

1. Matlab 的函数 `specgram` 可计算声谱图。详见帮助, 了解语法和所需参量。
2. 声谱图是数值计算出来的, 仅提供一个信号的时变频率成分的估计。

3. 一般调用形式为 `specgram(xx, 1024, fs)`。第二个参量是窗长度²，可以选用不同的长度观察不同尺度的声谱图。如果窗更长，例如 1024 或 2048³，声谱图可以查看单独的谱线。

为了看看声谱图是什么样，可以在命令窗口运行如下代码：

```
fs=8000;  
xx = cos(3000*pi*(0:1/fs:0.5));  
specgram(xx, 1024, fs);
```

注意声谱图在正弦的频率处有一条水平的线。

7.4 chirp 的声谱图

使用 `mychirp` 函数合成 chirp 信号。实验参数为：

1. 总时长为 3 秒，D/A 转换率为 $f_s=11025\text{Hz}$

2. 瞬时频率起始于 5000Hz，终止于 300Hz

听一听信号。Chirp 信号的声音频率是否线性变化，频率变高还是变低？

7.5 有趣的 chirp

再合成一个 chirp 信号，使用如下参数：

1. 总时长为 3 秒，采样率为 $f_s=11025\text{Hz}$

2. 频率起始于 3000Hz，终止于 -2000Hz（负频率）

听一听信号。频率是怎么变化的？

显示这个 chirp 信号的声谱图。

使用频谱理论（正频率成分和负频率成分）解释你听到的声音和看到的声谱图。

² 如果第 2 项为“空矩阵”，则默认为使用 256

³ 通常窗长度选为 2 的幂次，因为用了 FFT 计算。