

《数字信号处理》 实验报告

Matlab 平台的使用

学生姓名	赵展
学 号	U202117282
专业班级	种子2101班
实验平台	Matlab R2023b on Windows
联系方式	15225929727

2023年11月1日

目录

1	实验目的	3
2	实验内容	3
3	实验过程和结果分析	3
3.1	初识Matlab	3
3.1.1	查询相关命令	3
3.1.2	简单计算	3
3.1.3	复数相关运算	5
3.2	再识Matlab	5
3.2.1	Matlab数组索引	5
3.2.2	Matlab脚本文件	7
3.2.3	Matlab与声音相关的函数命令	9
3.3	用Matlab处理正弦信号	10
3.3.1	处理正弦信号	10
3.3.2	复数幅度	13
3.4	复指数简介	13
3.4.1	向量化	13
3.4.2	函数	14
3.5	复指数	14
3.5.1	生成正弦信号的M-文件	14
3.6	线性调频脉冲chirp	15
3.6.1	chirp 信号的Matlab合成方法	15
3.6.2	chirp的函数	17
3.6.3	声谱图	17
3.6.4	chirp的声谱图	18
3.6.5	有趣的chirp	18
4	实验小结	19

1 实验目的

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 实验内容

1. 了解Matlab基本功能
2. 了解Matlab数组索引、脚本文件、与声音相关函数命令等
3. 用Matlab 处理正弦信号
4. 用Matlab 处理复指数相关函数
5. 用Matlab 生成正弦信号的M-文件
6. 学习线性调频脉冲chirp的使用

3 实验过程和结果分析

3.1 初识Matlab

3.1.1 查询相关命令

在Matlab的命令行中输入一下命令：

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

最后均弹出相应的帮助窗口，lookfor filter命令则是打印出所有可用的滤波器，如图1

3.1.2 简单计算

在Matlab 命令行中依次输入以下命令：

```
>> lookfor filter
eventfilter - Create event filter for selecting timetable rows
filter - 1 维数字滤波器
filter2 - 二维数字滤波器
matlab.io.rowfilter - Selectively import rows of interest
double.groupfilter - 按组过滤
timeseries.filter - 修改 timeseries 对象的频率成分
timeseries.idealfilter - timeseries 理想滤波器
matlab.buildtool.io.FileCollection.select - Filter file collection
matlab.graphics.interaction.internal.brush.ButtonDownFilter - 刷亮隐藏回调
matlab.graphics.interaction.internal.pan.ButtonDownFilter - Pan suppression callback
matlab.graphics.interaction.internal.rotate3d.ButtonDownFilter - Rotation suppression callback
matlab.graphics.interaction.internal.zoom.ButtonDownFilter - 缩放隐藏回调
matlab.io.datastore.ParquetDatastore.RowFilter - 用于选择要导入的行的过滤器
```

图 1: 可用滤波器

```
pi*pi - 10
sin(pi/4)
ans ^ 2 %<--- "ans" holds the last result
xx = sin( pi/5 );
cos( pi/5 ) %<--- assigned to what?
yy = sqrt( 1 - xx*xx )
ans
```

部分结果展示如图2所示，可见Matlab是类似于python那样动态性语言，一个表达式命令会将值暂时赋给ans变量，自己也可以直接声明变量。

```
>> xx = sin( pi/5 );
      cos( pi/5 )

ans =

      0.8090

>> yy = sqrt( 1 - xx*xx )

yy =

      0.8090

>> ans

ans =

      0.8090
```

图 2: 实现简单计算

3.1.3 复数相关运算

在Matlab 命令行中依次输入以下命令：

```
real(zz), imag(zz)
zz = 3 + 4i, ww = -3 + 4j
abs([zz, ww]) %<-- Vector constructor
conj(zz + ww)
angle(zz)
exp(j * pi)
exp(j * [pi / 4, 0, -pi / 4 ])
```

会得到相应的结果

3.2 再识Matlab

3.2.1 Matlab数组索引

冒号的含义 在Matlab 命令行中依次输入以下命令：

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

所得结果如图3所示，可以了解到冒号的作用是“步进”，第一个数表示开始值，最后一个数表示结束值。如果存在中间值那么中间值为步距，不存在则默认步距为1。

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

jkl =
     2     6    10    14

jkl =
     0     1     2     3     4     5     6

jkl =
    99    98    97    96    95    94    93    92    91    90    89    88

ttt =
  列 1 至 13
  2.0000    2.1111    2.2222    2.3333    2.4444    2.5556    2.6667    2.7778    2.8889    3.0000    3.1111    3.2222    3.3333
  列 14 至 19
  3.4444    3.5556    3.6667    3.7778    3.8889    4.0000

tpi =
  列 1 至 13
  0.0000    0.3142    0.6283    0.9425    1.2566    1.5708    1.8849    2.1991    2.5133    2.8274    3.1416    3.4558    3.7699
  列 14 至 19
  4.0841    4.3982    4.7124    5.0265    5.3407    5.6549    5.9690    6.2832
```

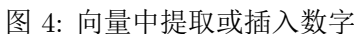
图 3: 冒号相关命令

从向量中提取/插入数字 在Matlab 命令行中依次输入以下命令：

得到如图4所示的结果，可以知道：第一行代码：创建了一个名为xx的数组，它是由三段拼接而成，第一段是由三个零组成的数组，第二段是从0到1均匀分布的包含5个元素的数组，最后一段由四个1组成的数组。它们由“[]”拼接成一个新的数组。

第三行代码：计算xx的大小，返回值是包含两个元素的数组，依次表示行数和列数，所以应该返回xx的行数：1，列数：12。

最后两行代码：类似于切片的操作，三个参数分别表示起点索引，步距和终点索引，返回一个新的数组。而end可以用来表示数组的末尾。



可以知道数组可以直接赋给一个新的变量，可以使用类似与切片的写法批量修改数组中一段元素的值。

6

使用以下一行代码：

```
xx(2 : 2 : end) = pi^pi
```

如图5所示，将xx的偶数索引的元素的值改成了常数 π^π 。

```
>> xx(2 : 2 : end) = pi^pi
xx =
    列 1 至 10
    0    36.4622    0    36.4622    0.2500    36.4622    0.7500    36.4622    1.0000    36.4622
    列 11 至 12
    1.0000    36.4622
fx >> |
```

图 5: 替换向量索引为偶数的值

3.2.2 Matlab脚本文件

关于向量的实验 在Matlab的命令行中输入以下命令：

```
xk = cos(pi * (0 : 11) / 4)
```

得到如图6所示的结果，该条命令是先经过(0:11)生成一个0到11的整数数组，然后再经过 $\pi * (0 : 11) / 4$ 将这些整数进行统一的计算生成一个全是弧度的12元素的数组，然后整个数组再进行余弦计算，得到最终的一个数组存在xk中。xk(1)表示xk的第一个元素，没有xk(0)，因为上文已经提到了数组的索引从1开始。

```
>> xk = cos(pi * (0 : 11) / 4)
xk =
    列 1 至 10
    1.0000    0.7071    0.0000   -0.7071   -1.0000   -0.7071   -0.0000    0.7071    1.0000    0.7071
    列 11 至 12
    0.0000   -0.7071
fx >> |
```

图 6: 关于向量的实验

循环 在Matlab的命令行中输入以下命令：

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

得到如图7所示的结果，向量yy被循环赋值

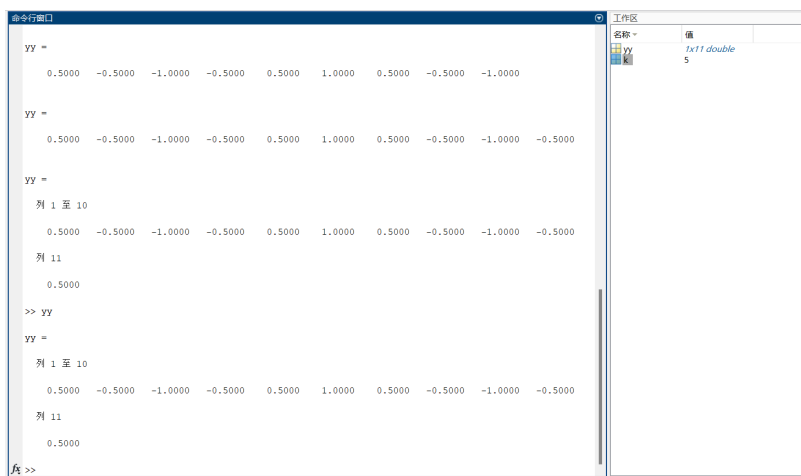


图 7: 循环

解释为什么需要写 $yy(k+6)$ 。如果用 $yy(k)$ 会发生什么？

因为 k 从-5开始，而 yy 的索引必须从1开始，如果使用 $yy(k)$ 将会报错。如果使用冒号而非循环，则应该是以下命令：

```
yy = cos(pi * (-5:5) / 3)
```

基本绘图命令 在Matlab的命令中输入以下命令：

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

得到如图8所示的结果， $plot(xx,yy)$ 命令和 $plot(zz)$ 命令都是绘制一个以 xx 为横坐标， yy 为纵坐标的图形，这里的 zz 是一个复数。当 xx 是一个向量时： $xx.*xx$ 是将 xx 的所有元素都变为自身的平方。

当 xx 是一个矩阵时： $xx.*xx$ 时将 xx 的每一项变为本身的平方。

创建Matlab脚本文件 创建一个名为mylab1.m的脚本文件，文件里是以下命令：

```
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
```

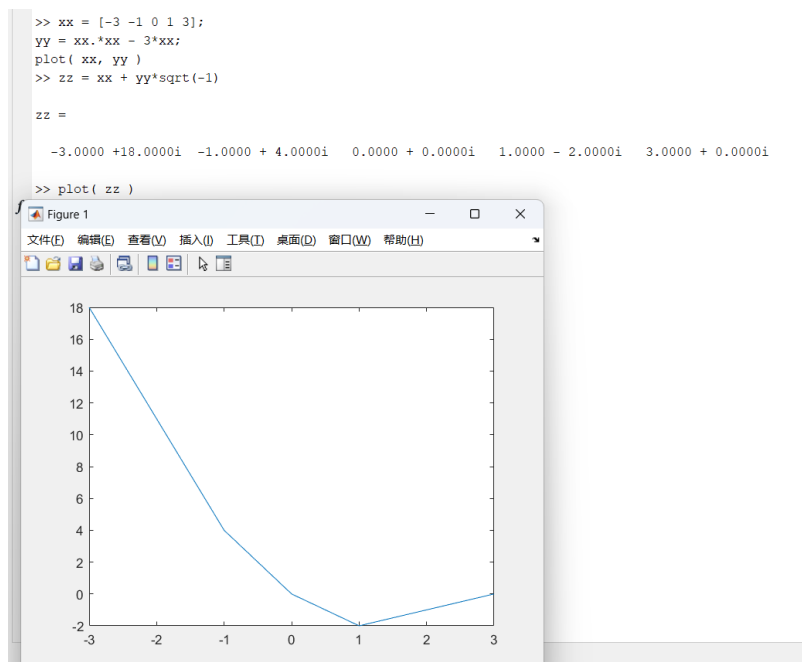



图 8: 绘图命令结果

```
title('TEST PLOT of a SINUSOID')
xlabel('TIME (sec)')
```

得到如图9所示的结果，红色虚线为zz的实部，蓝色实线为xx。

解释为什么 $\text{real}(zz)$ 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。

对 $zz = 1.4 * \exp(j * \pi / 2) * \exp(j * 5 * \pi * tt)$ 命令进行计算：

$$zz = 1.4 * e^{\frac{j\pi}{2}} * e^{j5\pi tt} = j * 1.4 * e^{j5\pi tt} = -1.4 \sin(5\pi tt) + 1.4j \cos(5\pi tt)$$

所以 $\text{real}(zz) = -1.4 \sin(5\pi tt)$,即为一个正弦函数。

3.2.3 Matlab与声音相关的函数命令

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）。那么就可以使用以下命令：

```
fs = 11025;
```

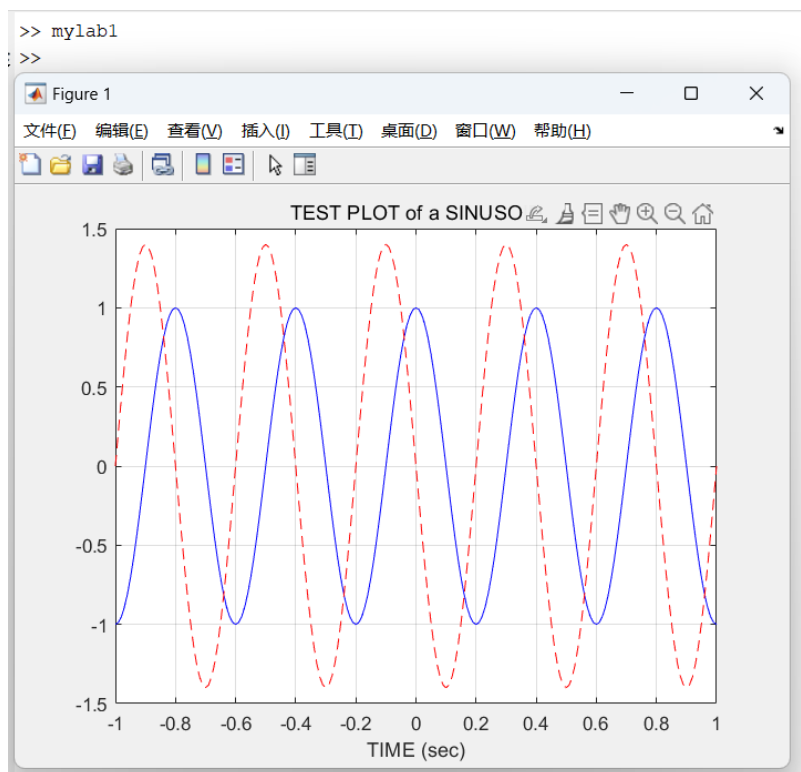


图 9: mylab1.m运行结果

```
dur = 2;
tt = (0 : 1 / fs : dur);
xx = sin(2 * pi * 2000 * tt);
plot(tt, xx);
soundsc(xx, fs)
```

得到如图10所示绘制出来的波形，同时可以听到将近2秒的尖锐的声音

3.3 用Matlab处理正弦信号

3.3.1 处理正弦信号

(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： $x_1(t) = A_1 \cos(2\pi(4000)(t - t_{m1}))$, $x_2(t) = A_2 \cos(2\pi(4000)(t - t_{m2}))$ 。其中 A_1 是你的年龄， $A_2 = 1.2A_1$ ， $t_{m1} = (37.2/M)T$ ， $t_{m2} = -(41.3/D)T$ ，这里M和D分别是你生日的月和日，T是周期。

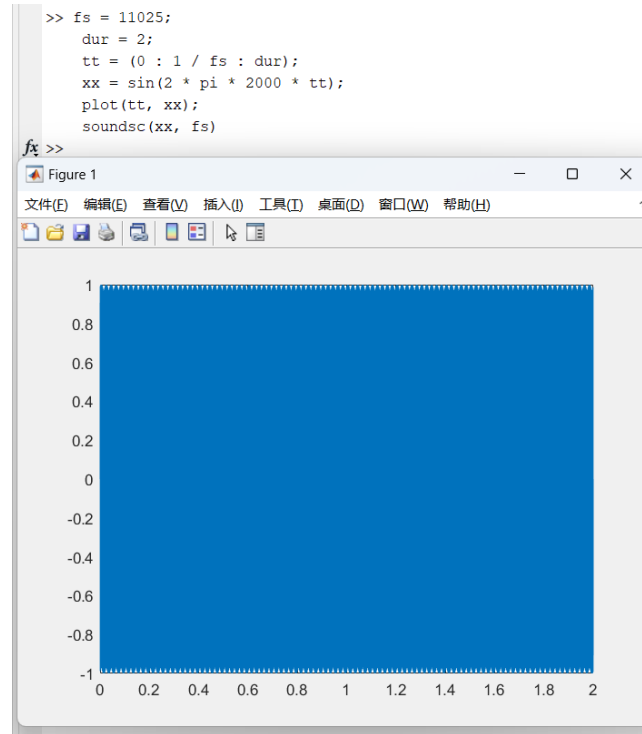


图 10: 正弦信号波形

(c)创建第3个正弦信号为 $x_3(t) = x_1(t) + x_2(t)$. 在Matlab中, 这表示把两个正弦向量中的值对应相加。绘制 $x_3(t)$ 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。

根据个人的情况可得:

$$A_1 = 19, A_2 = 1.2A_1 = 22.8$$

$$M = 1, D = 1$$

$$t_{m1} = (37.2/M)T = 37.2T, t_{m2} = -(41.3/D)T = -41.3T$$

所以在脚本文件`processing_inusoid_signals.m`中输入以下代码:

```
f = 4000; %frequency 4000Hz
T = 1 / f; %cycle 1/4000
fs = 30;
%Sampling frequency 30 per sec
tt = (-1 * T) : (1 / (fs * f)) : T;
A1 = 19;
A2 = 1.2 * A1;
M = 1;
D = 1;
```

```

tm1 = (37.2 / M) * T;
tm2 = -(41.3 / D) * T;
x1t = A1 * cos(2 * pi * 4000 * (tt- tm1));
x2t = A2 * cos(2 * pi * 4000 * (tt- tm2));
x3t = x1t + x2t;
subplot(3,1,1);
plot(tt,x1t);
title('赵展 x1(t)');
subplot(3,1,2);
plot(tt,x2t);
title('赵展 x2(t)');
subplot(3,1,3);
plot(tt,x3t);
title('赵展 x3(t)')

```

得到如图11所示绘制出来的波形。

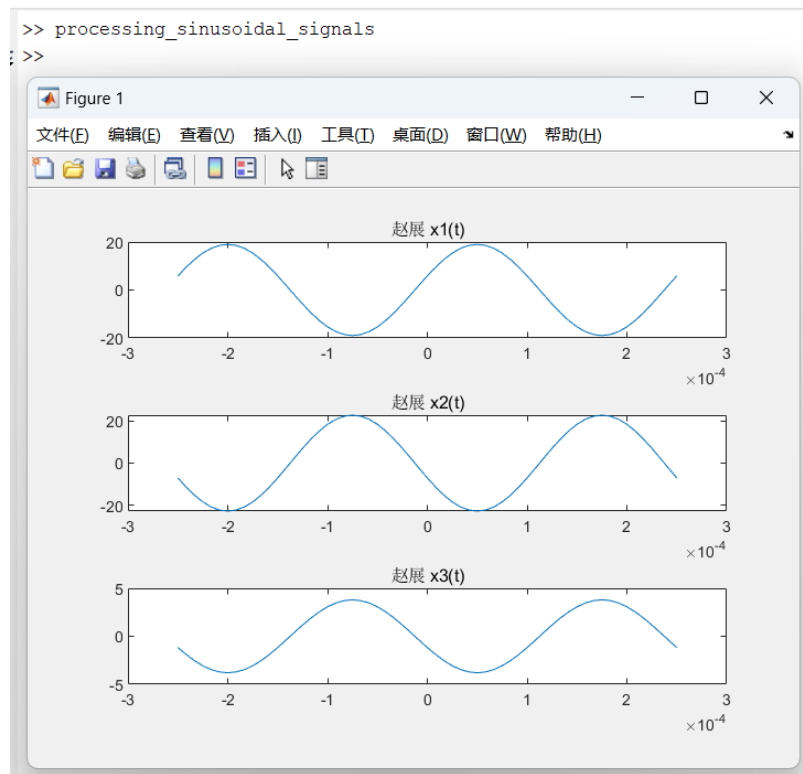


图 11: 处理正弦信号波形图

3.3.2 复数幅度

用一条语句生成以上正弦信号 $x_1(t)$ 的值, 可以使用复数幅度表示方法: $x_1(t) = \text{Re}\{Xe^{jt}\}$, 其中X和w为某个常数。

已知 $x_1(t) = A_1 \cos(2\pi(4000)(t-37.2T))$, 所以可以将 $x_1(t)$ 写成: $\text{Re}\{19e^{j2\pi 4000t-37.2}\}$, 在Matlab的命令行中输入以下命令:

```
f = 4000; %frequency 4000Hz
T = 1 / f; %cycle 1/4000
fs = 30;
%Sampling frequency 30 per sec
tt = (-1 * T) : (1 / (fs * f)) : T;
x1 = real(19 * exp(j * 2 * pi * (4000 * tt - 37.2)));
plot(tt, x1)
```

得到如图12所示绘制出来的波形, 和上边使用正弦函数表达式画出来的x1t图一样

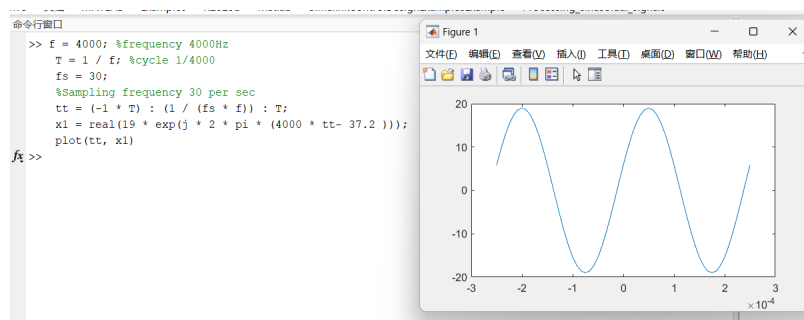


图 12: 复数幅度表示正弦函数

3.4 复指数简介

3.4.1 向量化

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-' )
```

使用向量化方法就是将循环等繁冗的代码转化成冒号形式，转化后的代码如下：

```
xk = (1 : 200) / 50;
sig = exp(j * 2 * pi * sqrt(xk .* xk + 2.25));
plot (xk, real(sig), 'mo-')
```

在Matlab中运行之后得到如图13所示的结果。

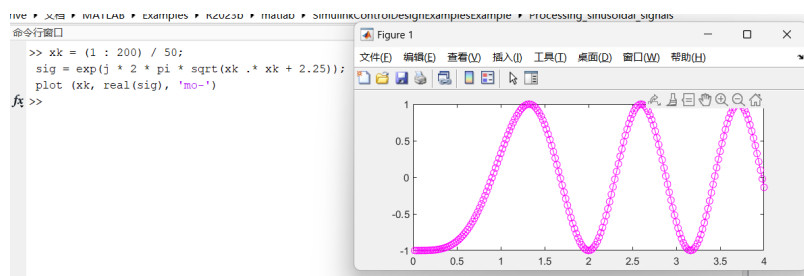


图 13: 向量化之后代码运行结果

3.4.2 函数

在Matlab中新建一个脚本文件`testcos.m`用来写一个函数，正确写法如下：

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

在命令行中输入：`testcos(1,10)`后显示出如图14所示的结果(部分)，表明函数调用成功。如果输入：`goodcos(1,10)`则会报错：“函数或变量 ‘goodcos’ 无法识别”。需要注意的是调用和函数是用的脚本文件名称而非脚本文件中写的函数的名称，但是最好这两个名称保持一致。

3.5 复指数

3.5.1 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，相位和时长 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。

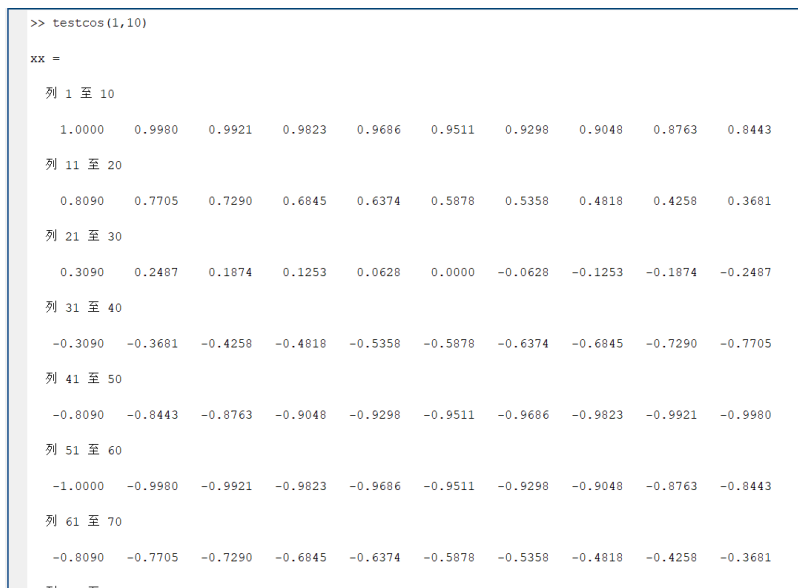


图 14: 测试函数运行结果

绘制你们的 $one_cos()$ 函数，参数选为： $A=95$ ， $\omega = 200\pi$ 弧度 / 秒， $\phi = \frac{\pi}{5}$ 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？

根据要求新建一个脚本文件 $one_cos.m$,其中内容如下：

```
function [x, t] = one_cos(A, Ang_freq, phase, dur)
t = 0 : 1 / (20 * (Ang_freq / 2 / pi)) : dur;
x = A * cos(Ang_freq * t + phase)
```

保存之后在命令行中调用：

```
[xx, tt] = one_cos(95, 200 * pi, pi / 5, 0.025);
plot(tt, xx)
```

得到的如图15所示的结果，生成正弦函数的表达式应为 $x(t) = 95 \cos(200\pi t + \frac{\pi}{5})$ ，角频率为 200π 可以算出频率 $f = \frac{\omega}{2\pi} = 100$ ，周期 $T = \frac{1}{f} = \frac{1}{100} s = 10ms$ 。从周期上看，0.025s绘制出2.5个周期，从相位上看，初相位 $\phi = \frac{\pi}{5}$ 对应到图像中 $95 \cos(\frac{\pi}{5}) \approx 76.9$ ，可以看出图像确为该函数。

3.6 线性调频脉冲chirp

3.6.1 chirp 信号的Matlab合成方法

在Matlab的命令行中输入以下命令生成一个chirp信号：

```
fsamp = 11025;
dt = 1 / fsamp;
dur = 1.8;
```

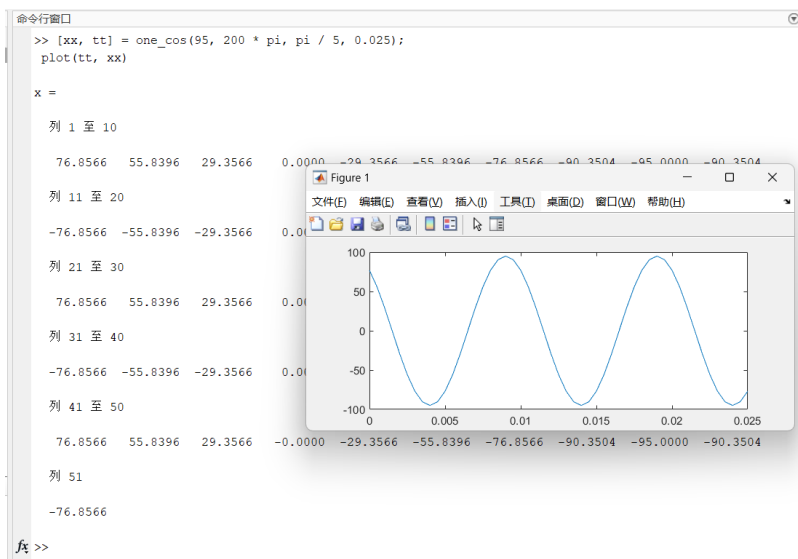


图 15: *one_cos*生成正弦函数的图像

```
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 ， μ ， 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。

上述命令运行之后我可以听到一个音调由低到高的声音，合成信号的总时长为1.8s，tt向量的长度(样本数)为19846。上述代码中生成的chirp信号可以处理为：

$$x(t_n) = 7.7 \cos(2\pi \times 500 t_n^2 + 2\pi \times 200 t_n + 200\pi)$$

其中 $t_n = nT_s$ ， t_n 即为tt， $T_s = dt = \frac{1}{f_{\text{samp}}} = \frac{1}{11025}$ ， $A = 7.7$ ， $\mu = 500$ ， $f_0 = 200$ ， $\phi = 200\pi$ 。tt = 0 时，频率为200Hz，tt=1.8时，频率为 $200 + 2 \times 500 \times 1.8 = 2000\text{Hz}$ ，可以绘制图像如图16所示。

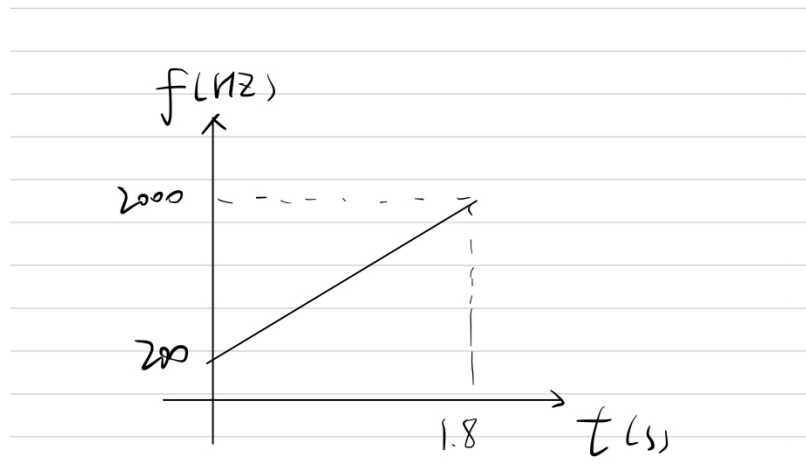


图 16: f 关于 t 的图像

3.6.2 chirp的函数

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。

在Matlab中新建一个脚本文件mychirp.m,输入以下内容：

```
function [xx, tt] = mychirp(f1, f2, dur, fsamp)
if(nargin < 4)
    fsamp = 11025;
end
tt = 0 : 1 / (fsamp) : dur;
psi = 2 * pi * (f1 * tt + (f2- f1) / dur / 2 * tt .* tt);
xx = real(exp(j * psi))
```

在Matlab中输入以下命令进行调用：

```
[xx, tt] = mychirp(2500, 500, 1.5, 11025);
soundsc(xx, 11025)
```

最终可以听到音调逐渐降低的声音，持续1.5秒。

3.6.3 声谱图

为了了解声谱图，在Matlab中输入以下命令：

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

可以得到如图17所示的声谱图。

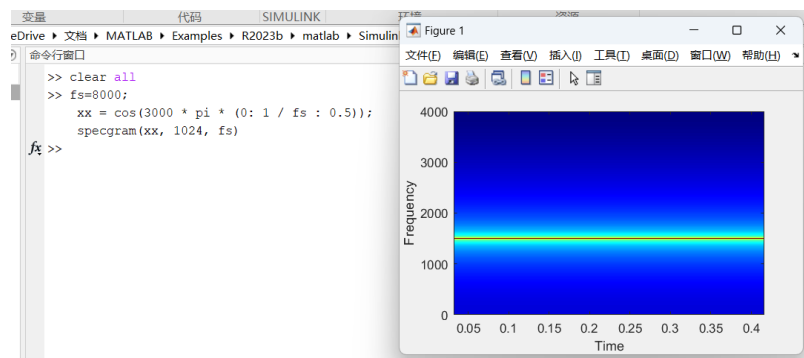


图 17: 声谱图

3.6.4 chirp的声谱图

在Matlab中输入以下命令调用mychirp生成chirp信号并查看其声谱图：

```
[xx, tt] = mychirp(5000, 300, 3, 11025);
soundsc(xx, 11025);
spectrogram(xx, 2048, 11025)
```

可以听到音调逐渐降低的声音，生成的声谱图如图18所示。

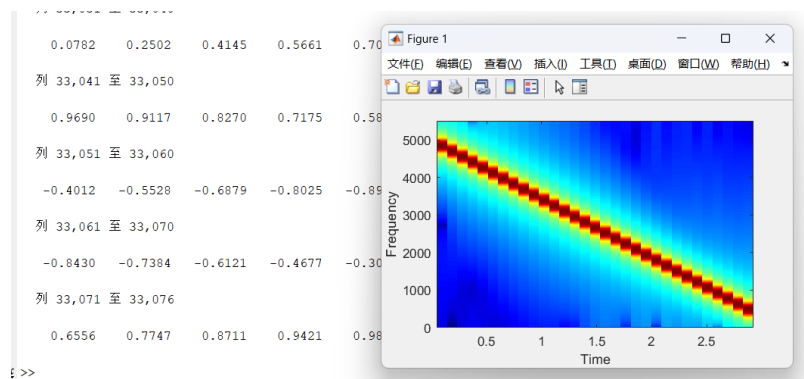


图 18: chirp的声谱图

3.6.5 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。

在Matlab中输入以下命令合成一个新的chirp信号：

```
[xx, tt] = mychirp(3000,-2000, 3, 11025);
soundsc(xx, 11025);
specgram(xx, 2048, 11025);
```

可以听到一个音调逐渐降低又升高的声音，说明频率先降低后升高，而得到声谱图如图19所示也是如此。声谱图中频率逐渐降低又升高，这是因为频率变为负的时候，只是相当于角频率的方向发生了改变，而真正的听到的频率还是增加的。

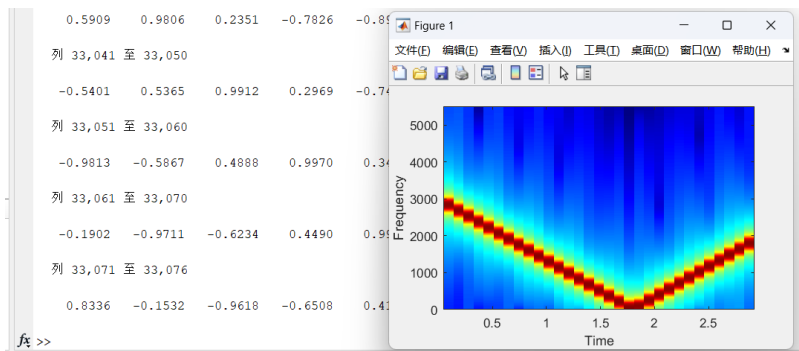


图 19: 有趣的chirp的声谱图

4 实验小结

通过本次实验，我成功熟悉了Matlab的一些基本使用，因为是从软件学院转到了这里，第一次接触Matlab,所以此次实验花了我不少的时间和精力。同时经老师推荐，这也是我第一次使用 \LaTeX 进行排版设计，掌握了一些基本的排版、公式代码。所以这次实验对我来说收获满满。希望可以给后续的实验和课程打下一个好的基础。