

Matlab 应用实践

指 导 书

计算机科学与技术学院

二〇二〇年

目 录

| | |
|------------------------|----|
| 实验一 基础练习..... | 3 |
| 实验二 离散时间信号分析..... | 13 |
| 实验三 数字信号处理综合设计..... | 15 |
| 实验四 Simulink 简单应用..... | 17 |
| 项 目 综合项目备选内容..... | 19 |

实验一 基础练习

一、实验目的：

1. 熟悉使用 MATLAB 软件
2. 掌握基本的 MATLAB 语言操作
3. 掌握基本的数学运算、绘图方法

二、实验内容：

练习 1 基本操作和简单语句输入

本联系旨在大家对 MATLAB 这一软件有一个宏观的印象，是大家了解 MATLAB 命令窗口的功能，能够在命令窗口中输入简单语句，且得到运行结果

1.1 系统的在线帮助

1.2 数值型向量（矩阵）的输入

1.3 数值型向量（矩阵）的输入 常用特殊矩阵等

1.4 符号向量（矩阵）的输入

1. 用函数 sym 定义符号矩阵：

函数 sym 实际是在定义一个符号表达式，这时的符号矩阵中的元素可以是任何的符号或者是表达式，而且长度没有限制。只需将方括号置于单引号中。

```
>> sym_matrix = str2sym(' [a b c; Jack Help_Me NO_WAY]')
```

```
sym_matrix =
```

```
 [ a,      b,      c]
```

```
 [Jack, Help_Me, NO_WAY]
```

2. 用函数 syms 定义符号矩阵

先定义矩阵中的每一个元素为一个符号变量，而后像普通矩阵一样输入符号矩阵。

```
>> syms a b c ;
```

```
>> M1 = sym('Classical');
```

```
>> M2 = sym('Jazz');
```

```
>> M3 = sym('Blues');
```

```
>> A = [a b c; M1, M2, M3; sym([2 3 5])]
```

```
A =
```

```
 [ a,      b,      c]
```

```
 [Classical, Jazz, Blues]
```

```
 [ 2,      3,      5]
```

① 数组（矩阵）的点运算

② 矩阵的运算

运算符：+（加）、-（减）、*（乘）、/（右除）、\（左除）、^（乘方）、'（转置）等；常用函数：det（行列式）、inv（逆矩阵）、rank（秩）、eig（特征值、特征向量）、rref（化矩阵为行最简形）

练习要求：

1. 练习数据和符号的输入方式，自定义数据和变量将前面的命令在命令窗口中执行通过
2. 输入 A=[7 1 5; 2 5 6; 3 1 5], B=[1 1 1; 2 2 2; 3 3 3], 在命令窗口中执行下列表达式，掌握其含义：

| | | | | |
|---------------|--------|--------|------------|----------------|
| A(2, 3) | A(:,2) | A(3,:) | A(:,1:2:3) | A(:,3).*B(:,2) |
| A(:,3)*B(2,:) | A*B | A.*B | A^2 | A.^2 |
| | | | | B/A |
| | | | | B./A |

3. 输入 $C=1:2:20$, 则 $C(i)$ 表示什么? 其中 $i=1,2,3,\dots,10$;
4. 查找已创建变量的信息, 删除无用的变量;

练习 2 编程

2.1 无条件循环

当需要无条件重复执行某些命令时,可以使用 for 循环:for 循环变量 t=表达式 1 : 表达式 2 : 表达式 3 语句体 end

说明:表达式 1 为循环初值,表达式 2 为步长,表达式 3 为循环终值;当表达式 2 省略时则默认步长为 1;for 语句允许嵌套.

例: 矩阵输入程序生成 3×4 阶的 Hilbber 矩阵.

```
m=input('矩阵行数:m=');for i=1 : 3 n= input('矩阵列数:n=');
for j=1 : 4 for i=1:m
H(i,j)=1/(i+j-1); for j=1:n
end disp(['输入第',num2str(i),'行,第', num2str(j),'列元素'])
end A(i, j) = input(' ')
end end
```

2.2 条件循环

1) if-else-then 语句

If-else-then 语句的常使用三种形式为:(1) if 逻辑表达式 (3) if 逻辑表达式 1 语句体 语句体 1 end elseif 逻辑表达式 2 语句体 2(2) if 逻辑表达式 1 elseif 逻辑表达式 3 语句体 ...else else 语句体 2 语句体 nend end

2) while 循环语句 while 循环的一般使用形式为:while 表达式 语句体 end

2.3 分支结构

若需要对不同的情形执行不同的操作,可用 switch 分支语句:

switch 表达式(标量或字符串)

```
case 值 1
语句体 1
case 值 2
语句体 2
Otherwise
语句体 n
end
```

说明:当表达式不是"case"所列值时,执行 otherwise 语句体。

① 建立 M 文件

将多个可执行的系统命令,用文本编辑器编辑后并存放在后缀为 .m 的文件中,若在 ATLAB 命令窗口中输入该 m-文件的文件名(不跟后缀.m!),即可依次执行该文件中的多个命令.这个后缀为.m 的文件,也称为 Matlab 的脚本文件(Script File)。

注意:文件存放路径必须在 Matlab 能搜索的范围内。

② 建立函数文件

对于一些特殊用户函数,系统提供了一个用于创建用户函数的命令 function,以备用户随时调用

1. 格式:

```
function [输出变量列表]=fun_name(输入变量列表)
```

用户自定义的函数体

2. 函数文件名为:fun_name,注意:保存时文件名与函数名最好相同;
3. 存储路径:最好在系统的搜索路径上。
4. 调用方法:输出参量=fun_name (输入变量)

例:

计算 $s = n!$,在文本编辑器中输入:

```
function s=pp(n);
```

```
s=1;
```

```
for i=1:n
```

```
s=s*i;
```

```
end
```

```
s;
```

在 MATLAB 命令窗口中输入:s=pp(5)

结果为 $s = 120$

练习要求:

1. 编写程序,计算 $1+3+5+7+\dots+(2n+1)$ 的值(用 input 语句输入 n 值).
2. 编写分段函数

$$f(x) = \begin{cases} x & 0 \leq x < 1 \\ 2-x & 1 \leq x \leq 2 \\ 0 & \text{其它} \end{cases}$$

的函数文件,存放于文件 ff.m 中,

计算出 $f(-3)$ 、 $f(\sqrt{2})$ 、 $f(\infty)$ 的值.

练习 3 矩阵计算

3.1 矩阵的创建

1.加、减运算

运算符:“+”和“-”分别为加、减运算符。

运算规则:对应元素相加、减,即按线性代数中矩阵的“+”、“-”运算进行。

例 3-1 在 Matlab 编辑器中建立 m 文件: LX0701.m

```
A=[1, 1, 1; 1, 2, 3; 1, 3, 6]
```

```
B=[8, 1, 6; 3, 5, 7; 4, 9, 2]
```

```
A+B=A+B
```

```
A-B=A-B
```

在 Matlab 命令窗口建入 LX0701, 则

结果显示: A+B=

```
9    2    7
```

```
4    7   10
```

```
5   12    8
```

```
A-B=
```

```
-7    0   -5
```

```
-2   -3   -4
```

```
-3   -6    4
```

2.乘法

运算符：*

运算规则：按线性代数中矩阵乘法运算进行，即放在前面的矩阵的各行元素，分别与放在后面的矩阵的各列元素对应相乘并相加。

(1) 两个矩阵相乘

例 3-2 在 Matalab 编辑器中建立 M 文件：LX0702.m

```
X=[2 3 4 5
```

```
1 2 2 1];
```

```
Y=
```

```
[0 1 1
```

```
1 1 0
```

```
0 0 1
```

```
1 0 0];
```

```
Z=X*Y
```

存盘

在命令行中建入 LX0702，回车后显示：

```
Z=
```

```
8 5 6
```

```
3 3 3
```

(2) 矩阵的数乘：数乘矩阵

上例中：a=2*X

则显示：a =

```
4 6 8 10
```

```
2 4 4 2
```

(3) 向量的点乘（内积）：维数相同的两个向量的点乘。

命令：dot 向量点乘函数

例：X=[-1 0 2]; Y=[-2 -1 1];

```
Z=dot(X,Y)
```

则显示：Z =

```
4
```

还可用另一种算法：

```
sum(X.*Y)
```

```
ans=
```

```
4
```

(4) 向量叉乘

在数学上，两向量的叉乘是一个过两相交向量的交点且垂直于两向量所在平面的向量。

在 Matlab 中，用函数 cross 实现。

命令 cross 向量叉乘函数

例 3-3 计算垂直于向量(1, 2, 3)和(4, 5, 6)的向量。

在 Matalab 编辑器中建立 M 文件：LX0703.m

```
a=[1 2 3];
```

```
b=[4 5 6];
```

```
c=cross(a,b)
```

结果显示：

```
c=
```

-3 6 -3

可得垂直于向量(1, 2, 3)和(4, 5, 6)的向量为±(-3, 6, -3)

(5) 混合积

混合积由以上两函数实现:

例 3-4 计算向量 $a=(1, 2, 3)$ 、 $b=(4, 5, 6)$ 和 $c=(-3, 6, -3)$ 的混合积

在 Matlab 编辑器中建立 M 文件: LX0704.m

```
a=[1 2 3]; b=[4 5 6]; c=[-3 6 -3];
```

```
x=dot(a, cross(b, c))
```

结果显示: x =

54

注意: 先叉乘后点乘, 顺序不可颠倒。

3.矩阵的除法

Matlab 提供了两种除法运算: 左除 (\) 和右除 (/)。一般情况下, $x=a\backslash b$ 是方程 $a*x=b$ 的解, 而 $x=b/a$ 是方程 $x*a=b$ 的解

例: $a=[1 \ 2 \ 3; 4 \ 2 \ 6; 7 \ 4 \ 9]$

```
b=[4; 1; 2];
```

```
x=a\b
```

则显示: x=

-1.5000

2.0000

0.5000

如果 a 为非奇异矩阵, 则 $a\backslash b$ 和 b/a 可通过 a 的逆矩阵与 b 阵得到:

```
a\b = inv(a)*b
```

```
b/a = b*inv(a)
```

4.矩阵乘方

运算符: ^

运算规则:

(1) 当 A 为方阵, p 为大于 0 的整数时, A^p 表示 A 的 P 次方, 即 A 自乘 P 次; p 为小于 0 的整数时, A^p 表示 A^{-1} 的 P 次方。

(2) 当 A 为方阵, p 为非整数时, 则其中 V 为 A 的特征向量, 为特征值矩阵

5.矩阵的转置

运算符: '

运算规则: 与线性代数中矩阵的转置相同。

6.矩阵的逆矩阵

例 3-5 求的逆矩阵

方法一: 在 Matlab 编辑器中建立 M 文件: LX07051.m

```
A=[1 2 3; 2 2 1; 3 4 3];
```

```
inv(A)或 A^(-1)
```

则结果显示为

ans =

```

1.0000    3.0000   -2.0000
-1.5000   -3.0000    2.5000
1.0000    1.0000   -1.0000

```

方法二：由增广矩阵进行初等行变换

在 Matlab 编辑器中建立 M 文件：LX07052.m

```

B=[1, 2, 3, 1, 0, 0; 2, 2, 1, 0, 1, 0; 3, 4, 3, 0, 0, 1];
C=rref(B)    %化行最简形
X=C(:, 4:6)

```

在 Matlab 命令窗口建入 LX07052，则显示结果如下：

```

C =
1.0000    0    0    1.0000    3.0000   -2.0000
0    1.0000    0   -1.5000   -3.0000    2.5000
0    0    1.0000    1.0000    1.0000   -1.0000

X =
1.0000    3.0000   -2.0000
-1.5000   -3.0000    2.5000
1.0000    1.0000   -1.0000

```

这就是 A 的逆矩阵。

7. 方阵的行列式

命令：det 计算行列式的值

例 3-6 计算上例中 A 的行列式的值

在 Matlab 编辑器中建立 M 文件：LX0706.m

```

A=[1 2 3; 2 2 1; 3 4 3];
D=det(A)

```

则结果显示为

D =

2

3.2 符号矩阵的运算

1. 符号矩阵的四则运算

把符号矩阵的四则运算简化为与数值矩阵完全相同的运算方式，其运算符为：加（+），减（-）、乘（×）、除（/、\）等或：符号矩阵的和（symadd），差（symsub），乘（symmul）。

例 3-7

```

A=sym('[1/x,1/(x+1);1/(x+2),1/(x+3)]');
B=sym('[x,1;x+2,0]')
C=B-A
D=A\B

```

则显示：

```

C=
x-1/x 1-1/(x+1)
x+2-1/(x+2) -1/(x+3)

D=
-6*x-2*x^3-7*x^2 1/2*x^3+x+3/2*x^2
6+2*x^3+10*x^2+14*x -2*x^2-3/2*x-1/2*x^3

```


2. 其他基本运算

符号矩阵的其他一些基本运算包括转置（'）、行列式（det）、逆（inv）、秩（rank）、幂（^）和指数（exp和expm）等都与数值矩阵相同

3. 符号矩阵的简化

符号工具箱中提供了符号矩阵因式分解、展开、合并、简化及通分等符号操作函数。（1）

(1) 因式分解

命令：factor 符号表达式因式分解函数

格式：factor(s)

说明：s为符号矩阵或符号表达式。常用于多项式的因式分解

例3-8 将 x^9-1 分解因式

在Matlab命令窗口建入

```
syms x
```

```
factor(x^9-1)
```

则显示：ans =

$$(x-1)*(x^2+x+1)*(x^3+x^3+1)^6$$

(2) 符号矩阵的展开

命令：expand 符号表达式展开函数

格式：expand(s)

说明：s为符号矩阵或表达式。常用在多项式的因式分解中，也常用于三角函数，指数函数和对数函数的展开中

例3-10 将 $(x+1)^3$ 、 $\sin(x+y)$ 展开

在Matlab编辑器中建立M文件：LX0710.m

```
syms x y
```

```
p=expand((x+1)^3)
```

```
q=expand(sin(x+y))
```

则结果显示为

```
p =
```

```
x^3+3*x^2+3*x+1
```

```
q =
```

$$\sin(x)\cos(y)+\cos(x)\sin(y)$$

(3) 同类式合并

命令：**Collect** 合并系数函数

格式：Collect(s,v) 将s中的变量v的同幂项系数合并。

Collect(s) s — 矩阵或表达式，此命令对由命令findsym函数返回的默认变量进行同类项合并。

(4) 符号简化

命令：simple或simplify 寻找符号矩阵或符号表达式的最简型

格式：Simple(s) s — 矩阵或表达式

说明：Simple(s)将表达式s的长度化到最短。若还想让表达式更加精美，可使用函数Pretty。

格式：Pretty(s) 使表达式s更加精美

练习要求：

将前面的命令在命令窗口中执行通过，掌握矩阵运算的基本方法和操作。

练习 4、秩与线性相关性

4.1 矩阵和向量组的秩以及向量组的线性相关性。

矩阵A的秩是矩阵A中最高阶非零子式的阶数；向量组的秩通常由该向量组构成的矩阵来计算。

命令：**rank**

格式：**rank(A)** A为矩阵式向量组构成的矩阵

例4-1 求向量组 $\alpha_1 = (1 \ -2 \ 2 \ 3)$ $\alpha_2 = (-2 \ 4 \ -1 \ 3)$ $\alpha_3 = (-1 \ 2 \ 0 \ 3)$ $\alpha_4 = (0 \ 6 \ 2 \ 3)$ $\alpha_5 = (2 \ -6 \ 3 \ 4)$ 的秩，并判断其线性相关性。

在Matlab编辑器中建立M文件：LX0714.m

```
A=[1 -2 2 3;-2 4 -1 3;-1 2 0 3;0 6 2 3;2 -6 3 4];
```

```
B=rank(A)
```

运行后结果如下：

```
B = 3
```

由于秩为 $3 <$ 向量个数，因此向量组线性相关。

4.2 向量组的最大无关组

命令：**rref**

格式：**rref(A)** A为矩阵

例4-2 求向量组 $a_1=(1,-2,2,3)$ ， $a_2=(-2,4,-1,3)$ ， $a_3=(-1,2,0,3)$ ， $a_4=(0,6,2,3)$ ， $a_5=(2,-6,3,4)$ 的一个最大无关组。

在Matlab编辑器中建立M文件：LX0715.m

```
a1=[1 -2 2 3]';
```

```
a2=[-2 4 -1 3]';
```

```
a3=[-1 2 0 3]';
```

```
a4=[0 6 2 3]';
```

```
a5=[2 -6 3 4]';
```

```
A=[a1 a2 a3 a4 a5]
```

```
format rat %以有理格式输出
```

```
B=rref(A) %求A的行最简形
```

运行后的结果为A =

```
1 -2 -1 0 2
```

```
-2 4 2 6 -6
```

```
2 -1 0 2 3
```

```
3 3 3 3 4
```

```
B =
```

```
1 0 1/3 0 16/9
```

```
0 1 2/3 0 -1/9
```

```
0 0 0 1 -1/3
```

```
0 0 0 0 0
```

从B中可以得到：向量 $a_1 \ a_2 \ a_4$ 为其中一个最大无关组

练习 5 线性方程的组的求解

我们将线性方程的求解分为两类：一类是方程组求唯一解或求特解，另一类是方程组求无穷解即通解。可以通过系数矩阵的秩来判断：若系数矩阵的秩 $r=n$ (n 为方程组中未知变量的个

数)，则有唯一解若系数矩阵的秩 $r < n$ ，则可能有无穷解。线性方程组的无穷解 = 对应齐次方程组的通解+非齐次方程组的一个特解；其特解的求法属于解的第一类问题，通解部分属第二类问题。

5.1求线性方程组的唯一解或特解（第一类问题）

这类问题的求法分为两类：一类主要用于解低阶稠密矩阵 —— 直接法；另一类是解大型稀疏矩阵 —— 迭代法。

利用矩阵除法求线性方程组的特解（或一个解）

方程： $AX=b$

解法： $X=A \setminus b$

例 5-1 求方程组

$$\begin{cases} 5x_1 + 6x_2 & = 1 \\ x_1 + 5x_2 + 6x_3 & = 0 \\ x_2 + 5x_3 + 6x_4 & = 0 \\ x_3 + 5x_4 + 6x_5 & = 0 \\ x_4 + 5x_5 & = 1 \end{cases} \text{ 的解}$$

解：在Matlab编辑器中建立M文件：LX0716.m

```
A=[5 6 0 0 0
```

```
1 5 6 0 0
```

```
0 1 5 6 0
```

```
0 0 1 5 6
```

```
0 0 0 1 5];
```

```
B=[1 0 0 0 1]';
```

```
R_A=rank(A) %求秩
```

```
X=A \ B %求解
```

运行后结果如下

```
R_A =
```

```
5
```

```
X = 2.2662
```

```
-1.7218
```

```
1.0571
```

```
-0.5940
```

```
0.3188
```

这就是方程组的解。

例5-2 求方程组

$$\begin{cases} x_1 + x_2 - 3x_3 - x_4 = 1 \\ 3x_1 - x_2 - 3x_3 + 4x_4 = 4 \\ x_1 + 5x_2 - 9x_3 - 8x_4 = 0 \end{cases}$$

解：在Matlab编辑器中建立M文件

```
A=[1 1 -3 -1;3 -1 -3 4;1 5 -9 -8];
```

```
B=[1 4 0]';
```

```
X=A \ B
```

```
X =
```

```
0
```

0

-0.5333

0.6000

练习 6 基本绘图方法（参考课本相关内容）

实验二 离散时间信号分析

一、实验目的

1. 掌握各种常用的序列，理解其数学表达式和波形表示。
2. 掌握在计算机中生成及绘制数字信号波形的方法。
3. 掌握序列的相加、相乘、移位、反褶等基本运算及计算机实现与作用。
4. 掌握线性卷积软件实现的方法。
5. 掌握计算机的使用方法和常用系统软件及应用软件的使用。
6. 通过编程，上机调试程序，进一步增强使用计算机解决问题的能力。

二、实验原理

1. 序列的基本概念

离散时间信号在数学上可用时间序列 $\{x(n)\}$ 来表示，其中 $x(n)$ 代表序列的第 n 个数字， n 代表时间的序列， n 的取值范围为 $-\infty < n < \infty$ 的整数， n 取其它值 $x(n)$ 没有意义。离散时间信号可以是由模拟信号通过采样得到，例如对模拟信号 $x_a(t)$ 进行等间隔采样，采样间隔为 T ，得到 $\{x_a(nT)\}$ 一个有序的数字序列就是离散时间信号，简称序列。

2. 常用序列

常用序列有：单位脉冲序列（单位抽样） $\delta(n)$ 、单位阶跃序列 $u(n)$ 、矩形序列 $R_N(n)$ 、实指数序列、复指数序列、正弦型序列等。

3. 序列的基本运算

序列的运算包括移位、反褶、和、积、标乘、累加、差分运算等。

4. 序列的卷积运算

$$y(n) = \sum_{m=-\infty}^{\infty} x(m)h(n-m) = x(n) * h(n)$$

上式的运算关系称为卷积运算，式中 $*$ 代表两个序列卷积运算。两个序列的卷积是一个序列与另一个序列反褶后逐次移位乘积之和，故称为离散卷积，也称两序列的线性卷积。其计算的过程包括以下 4 个步骤。

(1) 反褶：先将 $x(n)$ 和 $h(n)$ 的变量 n 换成 m ，变成 $x(m)$ 和 $h(m)$ ，再将 $h(m)$ 以纵轴为对称轴反褶成 $h(-m)$ 。

(2) 移位：将 $h(-m)$ 移位 n ，得 $h(n-m)$ 。当 n 为正数时，右移 n 位；当 n 为负数时，左移 n 位。

(3) 相乘：将 $h(n-m)$ 和 $x(m)$ 的对应点值相乘。

(4) 求和：将以上所有对应点的乘积累加起来，即得 $y(n)$ 。

三、主要实验仪器及材料

微型计算机、Matlab 教学版

四、实验内容

1. 知识准备

认真复习以上基础理论，理解本实验所用到的实验原理。

2. 离散时间信号（序列）的产生

利用 MATLAB 语言编程产生和绘制下列有限长序列：

(1) 单位脉冲序列 $\delta(n)$

(2) 单位阶跃序列 $u(n)$

(3) 矩形序列 $R_g(n)$

(4) 正弦型序列 $x(n) = A \sin(\frac{\pi}{5}n + \frac{\pi}{3})$

(5) 任意序列

$$x(n) = \delta(n) + 2\delta(n-1) + 3\delta(n-2) + 4\delta(n-3) + 5\delta(n-4)$$

$$h(n) = \delta(n) + 2\delta(n-1) + \delta(n-2) + 2\delta(n-3)$$

3. 序列的运算

利用 MATLAB 语言编程完成上述两序列的移位、反褶、和、积、标乘、累加等运算，并绘制运算后序列的波形。

4. 卷积运算

利用 MATLAB 语言编制一个计算两个序列线性卷积的通用程序，计算上述两序列 $x(n) * h(n)$ ，并绘制卷积后序列的波形。

5. 上机调试并打印或记录实验结果。

6. 完成实验报告。

五、思考题

1. 如何产生方波信号序列和锯齿波信号序列？

2. 实验中所产生的正弦序列的频率是多少？是否是周期序列？

六、实验要求

1. 简述实验原理及目的。

2. 列出计算卷积的公式，画出程序框图，并列出实验程序清单（可略）（包括必要的程序说明）。

3. 记录调试运行情况及所遇问题的解决方法。

4. 给出实验结果，并对结果作出分析。

5. 简要回答思考题。

实验三 数字信号处理综合设计

一、实验目的

1. 学会 MATLAB 的使用，掌握 MATLAB 的程序设计方法；
2. 掌握在 Windows 环境下语音信号采集的方法；
3. 掌握数字信号处理的基本概念、基本理论和基本方法；
4. 掌握 MATLAB 设计 FIR 和 IIR 数字滤波器的方法；
5. 学会用 MATLAB 对信号进行分析和处理。

二、实验原理

参考《数字信号处理》教材。

三、主要实验仪器及材料

微型计算机、Matlab 教学版

四、实验内容

1. 语音信号的采集

要求利用 windows 下的录音机或其他软件，录制一段自己的话音，时间控制在 1 秒左右。然后在 MATLAB 软件平台下，利用函数 `audioread` 对语音信号进行采样，记住采样频率和采样点数。通过 `audioread` 函数的使用，要求理解采样频率、采样位数等概念。

`audioread` 函数调用格式和使用方法，请使用 Matlab 中的 `help` 或者 `doc` 指令查询

2. 语音信号的频谱分析

要求首先画出语音信号的时域波形；然后对语音信号进行频谱分析，在 MATLAB 中，可以利用函数 `fft` 对信号进行快速付立叶变换，得到信号的频谱特性；从而加深对频谱特性的理解。

3. 设计数字滤波器和画出频率响应

根据语音信号的特点给出有关滤波器的性能指标：

- 1) 低通滤波器性能指标， $f_p=1000\text{Hz}$ ， $f_s=1200\text{Hz}$ ， $A_s=100\text{dB}$ ， $A_p=1\text{dB}$ ；
- 2) 高通滤波器性能指标， $f_s=4800\text{Hz}$ ， $f_p=5000\text{Hz}$ ， $A_s=100\text{dB}$ ， $A_p=1\text{dB}$ ；
- 3) 带通滤波器性能指标， $f_{p1}=1200\text{Hz}$ ， $f_{p2}=3000\text{Hz}$ ， $f_{s1}=1000\text{Hz}$ ， $f_{s2}=3200\text{Hz}$ ， $A_s=100\text{dB}$ ， $A_p=1\text{dB}$ 。

要求学生首先用窗函数法设计上面要求的三种滤波器，在 MATLAB 中，可以利用函数 `fir1` 设计 FIR 滤波器；然后在用双线性变换法设计上面要求的三种滤波器，在 MATLAB 中，可以利用函数 `butter`、`cheby1` 和 `ellip` 设计 IIR 滤波器；最后，利用 MATLAB 中的函数 `freqz` 画出各

滤波器的频率响应。

4. 用滤波器对信号进行滤波

比较两种滤波器的性能，然后用性能好的各滤波器分别对采集的信号进行滤波，在 MATLAB 中，FIR 滤波器利用函数 `fft`, `filter` 对信号进行滤波，IIR 滤波器利用函数 `filter` 对信号进行滤波。

5. 比较滤波前后语音信号的波形及频谱

要求在一个窗口同时画出滤波前后的波形及频谱。

6. 回放语音信号

在 MATLAB 中，函数 `sound` 可以对声音进行回放。其调用格式：

```
sound(x,fs,bits);
```

可以感觉滤波前后的声音有变化。

五、实验思考

1. 双线性变换法中 Ω 和 ω 之间的关系是非线性的，在实验中你注意到这种非线性关系了吗？从那几种数字滤波器的幅频特性曲线中可以观察到这种非线性关系？

2. 能否利用公式完成脉冲响应不变法的数字滤波器设计？为什么？

六、实验要求

1. 简述实验原理及目的。

2. 按照实验步骤及要求，比较各种情况下的滤波性能。

3. 总结实验所得主要结论。

4. 简要回答思考题。

实验四 Simulink 简单应用

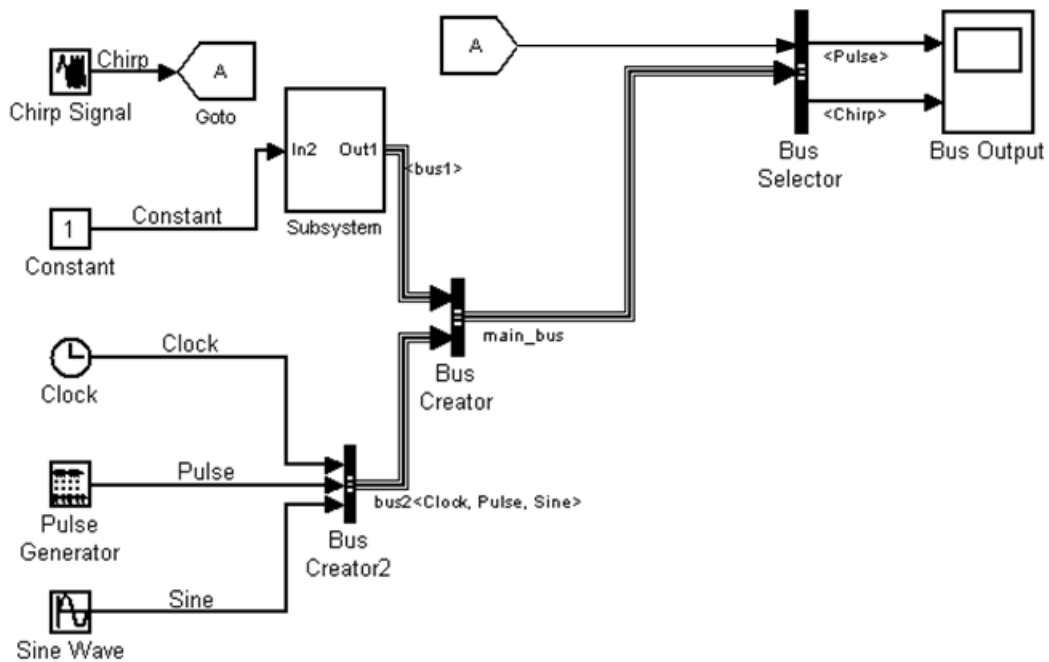
一、实验目的

1. 学会 SIMULINK 的使用，掌握 SIMULINK 设计方法；
2. SIMULINK 设计 FIR 和 IIR 数字滤波器的方法；
3. 学会用 SIMULINK 对信号进行分析和处理。

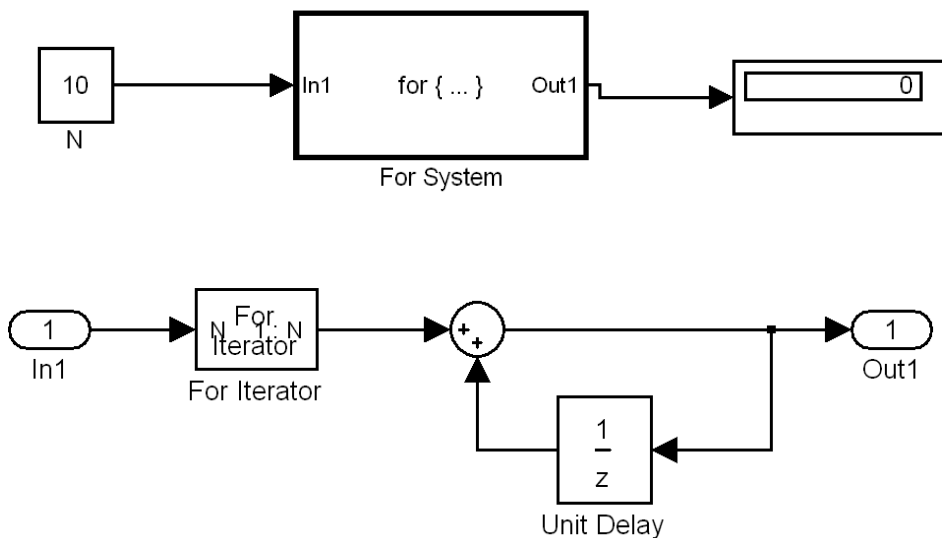
二、实验内容

1. 初步 SIMULINK 的使用方法，识别常用的模块，如信号源模块 Sources 接收模块 Sinks，以及数学运算模块 Math Operations，接口模块 Ports & Operations。

任务 1：构建如下图所示的 Bus Signal 系统。

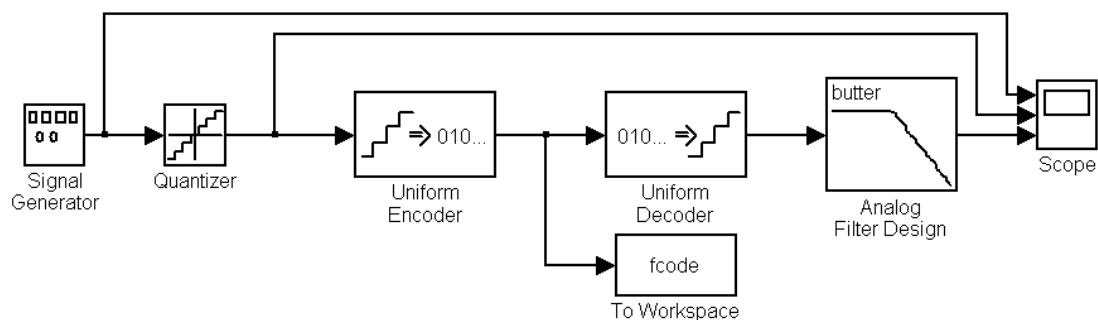


任务 2：参照下面系统，实现 1-100 求和运算

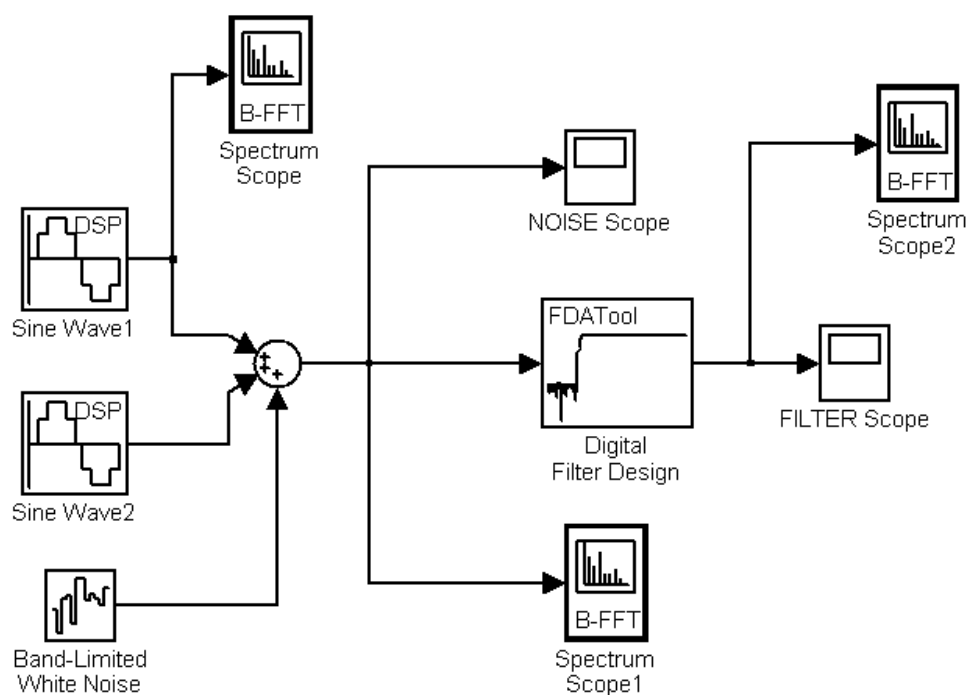


任务 3：利用相关模块实现 4 个 D 触发器实现 16 进制计数器

2.理解 PCM 编解码原理，构建如下系统实现 PCM 编解码仿真



3.理解数字滤波设计方法，相关参数自拟，实现数字滤波器设计



三、实验要求

1. 简述实验原理及目的。
2. 完成实验任务及要求，简述参数设定、实现方法和过程
3. 总结实验所得主要结论。

综合项目备选内容

1、语音信号的一些分析处理

任务：

利用 Matlab 设计一个简单易用的图形用户界面（GUI），能够实现对语音信号进行时域和频域上的一些相关处理和分析，并能够通过图形界面功能，将处理后的语音进行播放或者输出显示相关图形。

内容、步骤和要求：

（1）语音信号的采集

使用电脑的声卡设备采集一段语音信号，并存储在电脑中。

（2）语音信号的处理

语音信号的处理分析的目的就在于有效提取并表示语音信号所携带的信息。语音信号分析可分为时域分析和频域分析等处理方法。时域分析主要是对信号的时域波形进行分析，

1) 语音信号的时域分析

- 提取：通过图形用户界面上的菜单功能键采集电脑设备上的一段音频信号，完成音频信号的频率、幅度等信息的提取，并获得该语音信号的波形图。
- 调整：通过图形用户界面上的菜单功能键对输入的音频信号进行各种变换，如幅度变换、频率变换等操作，已实现对于因信号的调整。

2) 语音信号的频域分析

- 变换：在图形用户界面下对采集的语音信号进行傅里叶等变换，并给出变换前后的频谱图和变换后的倒谱图。
- 滤波：对语音信号进行滤波，分别采用低通、高通、带通和高阻滤波器，并比较各种滤波后的效果。

（3）语音信号的效果显示处理

通过用户图形界面的输出功能，将处理后的信号进行播放，试听处理后的效果。

扩展选做部分：可考虑添加一些实际项目中用到的其他关于语音信号的相关处理。

2、心电信号分析处理

任务：

利用 Matlab，设计一个简单的心电信号分析系统，能够对输入的数字心电信号，进行一定的初步分析处理和频谱分析。要求分别采用两种方式进行仿真，即直接采用 Matlab 语言编程的静态仿真方式、采用 Simulink 进行动态建模和仿真的方式。

内容、步骤和要求：

(1) 对原始数字心电信号进行采集，并绘制出其时域波形。

访问美国麻省理工学院提供的 MIT-BIH 数据库，并载数据，为了方便，采用其 txt 格式的数据文件作为我们的原心电信号数据。利用 Matlab 提供的文件 `textread` 或 `textscan` 函数，读取 txt 数据文件中的信号，并且还原实际波形。

可以用这个链接方面的浏览 MIT 的各种生理数据库：

<https://www.physionet.org/content/mitdb/1.0.0/>

(2) 对数字信号数据做一次线性插值，使其成为均匀数字信号，以便后面的信号分析；并思考计算：如何根据心电图波形，计算出心率。

(3) 根据心电信号的频域特征（自己查阅相关资料），设计相应的低通和高通滤波器。如果分析频谱，滤波效果不明显，则需变动滤波器参数指标，重新设计滤波器。通过频谱分析，多次试验确定最合适的滤波器。

(4) 编程绘制实现信号处理前后的频谱，做频谱分析，得出相关结论。

(5) 根据前面的设计，进行基于 Simulink 的动态仿真设计。实现心电信号的分析 and 处理。给出系统的基于 Simulink 的动态建模和仿真的系统框图，同时记录系统的各个输出点的波形和频谱图。

扩展选做部分：50Hz（或者 60Hz）工频陷波器设计

由于电子设备采集到的信号经常会混有电源线干扰。电源线干扰是以 50Hz（或者 60Hz）为中心的窄带噪声，带宽小于 1Hz。设计相应的带阻滤波器滤除电源线干扰，并对处理后的信号做频谱分析。