

数字信号处理

Matlab用于DSP入门指南

全世界数以百万计的工程师和科学家都在使用 MATLAB 分析和设计改变着我们的世界的系统和产品。基于矩阵的 MATLAB 语言是世界上表示计算数学最自然的方式。可以使用内置图形轻松可视化数据和深入了解数据。

本手册收集了在DSP算法中常用的Matlab函数及简单示例，期望通过较短的篇幅来展现MATLAB 矩阵运算的核心功能和超强的图形化能力，用于帮助未学习过 MATLAB 的同学可以快速上手开展 DSP 课程实验。

中国科学技术大学

电子工程与信息科学系

2023年9月

目录

[目录 2](#_Toc143557326)

[1. Matlab常用功能 4](#_Toc143557327)

[1.1 Matlab的命令行窗口 4](#_Toc143557328)

[1.2 查看Matlab帮助【非常重要！】 4](#_Toc143557329)

[1.3 M文件的使用 5](#_Toc143557330)

[1.4 Matlab的矩阵运算 5](#_Toc143557331)

[1.5 Matlab中的符号计算（Symbolic Computations） 6](#_Toc143557332)

[1.5.1 符号变量、表达式及绘图 6](#_Toc143557333)

[1.5.2 基于符号表达的级数求和 6](#_Toc143557334)

[1.6 Matlab中如何自定义函数 7](#_Toc143557335)

[1.7 Matlab绘图示例 7](#_Toc143557336)

[1.7.1 示例一（用plot同时绘制两条曲线） 7](#_Toc143557337)

[1.7.2 示例二（用stem绘制多个序列） 7](#_Toc143557338)

[1.7.3 示例三（从复平面看幅频特性） 8](#_Toc143557339)

[1.8 plot函数 8](#_Toc143557340)

[1.9 subplot函数 10](#_Toc143557341)

[1.10 Matlab文件操作示例 11](#_Toc143557342)

[1.11 使用Matlab的信号处理工具箱 12](#_Toc143557343)

[2. Matlab中的DSP常用函数/工具 12](#_Toc143557344)

[2.1 与信号产生、分析有关的函数 12](#_Toc143557345)

[2.1.1 用行向量表示一个序列 13](#_Toc143557346)

[2.1.2 示例：产生一个理想采样信号序列 13](#_Toc143557347)

[2.1.3 基于DFT绘制信号的幅度谱和相位谱 13](#_Toc143557348)

[2.1.4 fourier Fourier transform 14](#_Toc143557349)

[2.1.5 Matlab已有的波形生成函数 14](#_Toc143557350)

[2.1.6 Matlab已有的窗函数 14](#_Toc143557351)

[2.2 与传递函数相关的函数 15](#_Toc143557352)

[2.2.1 poly 具有指定根的多项式或特征多项式 15](#_Toc143557353)

[2.2.2 roots 多项式根 15](#_Toc143557354)

[2.2.3 conv 卷积和多项式乘法 15](#_Toc143557355)

[2.2.4 filtfilt Zero-phase digital filtering 15](#_Toc143557356)

[2.2.5 filter 1 维数字滤波器 16](#_Toc143557357)

[2.2.6 zp2tf, tf2zp零极点模型←→传递函数 16](#_Toc143557358)

[2.2.7 freqs函数：模拟滤波器的频率响应 16](#_Toc143557359)

[2.2.8 freqz 数字滤波器的频率响应 16](#_Toc143557360)

[2.3 滤波器设计相关函数和工具 17](#_Toc143557361)

[2.3.1 IIR 滤波器设计相关函数 17](#_Toc143557362)

[2.3.2 FIR 滤波器设计相关函数 18](#_Toc143557363)

[2.3.3 FVTool: Filter Visualization Tool 18](#_Toc143557364)

[2.3.4 FilterDesign 工具 19](#_Toc143557365)

[3. Matlab示例代码 20](#_Toc143557366)

[3.1 Matlab分析信号基本示例 20](#_Toc143557367)

[3.1.1 非周期信号的CFT 20](#_Toc143557368)

[3.1.2 非周期序列的DTFT 20](#_Toc143557369)

[3.2 IIR滤波器设计 21](#_Toc143557370)

[3.2.1 直接设计数字滤波器 21](#_Toc143557371)

[3.2.2 脉冲相应不变法 21](#_Toc143557372)

[3.2.3 双线性变换法 22](#_Toc143557373)

[3.3 模拟频带变换 27](#_Toc143557374)

[3.3.1 模拟低通变换为模拟高通 27](#_Toc143557375)

[3.3.2 低通变换为带通 27](#_Toc143557376)

[3.4 窗函数法设计FIR滤波器 28](#_Toc143557377)

[3.4.1 低通滤波器实现 28](#_Toc143557378)

[3.4.2 带通滤波器实现 29](#_Toc143557379)

[3.4.3 MultiBand实现 30](#_Toc143557380)

[3.5 频率抽样法设计FIR滤波器 30](#_Toc143557381)

[3.5.1 低通滤波器（基本实现方式） 30](#_Toc143557382)

[3.5.2 低通滤波器（增加过渡带抽样点） 31](#_Toc143557383)

# Matlab常用功能

## Matlab的命令行窗口

打开Matlab后，可以看到一个命令行窗口（Command Window）在命令行中输入命令（由提示符 (>>) 表示）。类似Windows操作系统下的命令提示符窗口和Linux操作系统下的Term窗口，这个窗口可用于输入各类命令。

例如，可尝试输入如下命令（输入一条命令后回车），并观察返回结果。

>> pwd

>> dir

>> datetime

>> version

>> doc

如果在命令行窗口输入命令行的末尾加英文标点分号“;”，则执行该命令，但是并不在命令行窗口提示结果，类似操作系统下静默模式执行命令。

## 查看Matlab帮助【非常重要！】

【Matlab中集成了非常多的函数，离开联机帮助很难记住函数的语法格式】

在MATLAB的命令窗口中键入help，即可获得与上一次命令相关的，文本格式的简易帮助信息。

>>help

如果对MATLAB的plot函数感兴趣，想了解使用方法，则键入：

>> help plot

如果需要查阅MATLAB的图形化联机帮助信息，则键入：

>> doc

如果需要查阅MATLAB的图形化联机帮助信息，并直接跳转到plot函数有关的信息，则键入：

>> doc plot

建议第一次使用MATLAB时浏览联机帮助中MATLAB的基本用方法，即键入：

>> doc Matlab

## M文件的使用

使用Matlab的时候，可以在“Command Window”内直接书写Matlab代码，也可以将代码保存到M文件中，然后运行该文件。

使用Matlab主界面“主页”→“新建”→“脚本”（或者使用热键“CTRL+N”）可以打开文本编辑器编辑M文件。M文件及Matlab的代码文件，点击Matlab主界面“编辑器”→“运行”可执行文件中的命令序列。在路径设置正确的情况下，在“Command Window”中直接输入M文件的名称可以运行M文件中的代码。注意：不同版本下运行M文件的方式略有差异，请查看联机帮助。

注意：M文件的命名请以英文字母开头，用字母和数字组成；不要起中文文件名称，也不要在文件名称中使用“(”、“)”等特殊字符；M文件的名称不能和Matlab系统函数重名。

## Matlab的矩阵运算

矩阵运算是MATLAB的核心功能。建议第一次使用MATLAB时通过查看示例较为全面地了解一下相应的功能函数。如果想了解MATLAB中有关矩阵的操作运算函数，可以键入如下两条命令之一。

>> help matfun

>> doc matfun

在开始使用Matlab之前，建议运行以下代码来了解Matlab的矩阵（数组）运算的基本概念。

A=[1 2 3;4 5 6]; %给2×3（2行3列）的矩阵矩阵赋值

A(1) %查看矩阵A的第一个元素

A(9) %查看矩阵A的第9个元素

B=A'; %求矩阵A的转置

B %查看矩阵B的内容

C=A+B'; %矩阵的加法运算

C %查看矩阵C的内容

C=A(1,:) %C的内容为矩阵A的第一行

C=A(2,:) %C的内容为矩阵A的第2行

C=A(:,1) %C的内容为矩阵A的第一列

C=A(:,3) %C的内容为矩阵A的第3列

下表列出了一些Matlab中常用的矩阵运算函数。

表 ‑1 常用线性代数函数

|  |  |
| --- | --- |
| B=A' | 矩阵转置 |
| C=A+B | 矩阵相加 |
| C=A\*B | 矩阵相乘 |
| C=A^k | 矩阵幂 |
| C=A.\*B | 矩阵点乘，即两维数相同的矩阵各对应元素相乘 |
| inv(A) | 矩阵的逆矩阵 |
| det(A) | 矩阵的行列式的值 |
| rank(A) | 计算矩阵的秩 |
| eig(A) | 矩阵的特征值 |
| [X,D]=eig(A) | 矩阵的特征向量X和以特征值为元素的对角阵D |
| p=poly(A) | 矩阵的特征多项式 |
| r=roots(p) | 特征多项式方程的根 |
| conv(p1,p2) | 两多项式相乘 |

## Matlab中的符号计算（Symbolic Computations）

数值计算：必须先对变量赋值，然后才能参与运算，运算结果是数值。

符号计算：无须事先对变量赋值，而将所得到的结果以标准的符号形式来表示。

### 符号变量、表达式及绘图

* 符号变量定义

syms x % 创建符号变量x

f\_exp = exp(-2\*x^2); % 指数函数

* 符号表达式（函数）绘图

ezplot(f\_exp,[-9,9]) % 绘制指数函数

* 符号变量转数值变量

rho = sym('(1 + sqrt(n))/2')

eval(rho);

* 多项式替换

poly2sym([1 0 -2 -5])

poly2sym([1 0 -2 -5],'2\*t+1')

### 基于符号表达的级数求和

MATLAB中求级数的命令为symsum。

F = symsum(f,k,a,b) returns the symbolic sum of the series f with respect to the summation index k from the lower bound a to the upper bound b.

[示例] 计算 

syms k x

F1 = symsum(k^2,k,0,10)

[示例] 计算 

syms k x

F2 = symsum(1/k^2,k,1,Inf)

## Matlab中如何自定义函数

Matlab中一个函数需要定义一个M文件，该文件名称和函数的名称一致。例如：我们需要定义个函数完成两个矩阵的加法和乘法运算。函数名称为“mat\_plus”，则对应写一个名称为“mat\_plus.m”的M文件。“mat\_plus.m”文件内容如下所示，文件中“%”之后的部分被认为是注释信息。

function [C,D]=mat\_plus(A,B)

%矩阵加法和矩阵乘法的计算

C=A+B;

D=A\*B;

在“Command Window”中输入如下命令

>>A=[1 2 3;4 5 6;7 8 9]; %给矩阵赋值

>> [C,D]=mat\_plus(A, A') %矩阵C为矩阵A＋A'的结果；D为矩阵A\*A'的结果

对于上述函数，还可以测试一下如下命令行的运行结果

>> help mat\_plus

## Matlab绘图示例

### 示例一（用plot同时绘制两条曲线）

x=0:0.01:2\*pi

plot(x,sin(x),'green')

hold on

plot(x,sin(2\*x),'red')

### 示例二（用stem绘制多个序列）

n=0:50;

A=444.128;

a=50\*sqrt(2.0)\*pi;

T=0.001;

w0=50\*sqrt(2.0)\*pi;

x=444.128\*exp(-a\*n\*T).\*sin(w0\*n\*T);

X=fft(x);

% Draw x,abs(X),angle(X)

hold on

stem(x)

stem(abs(X),'fill','r-.')

stem(angle(X),'fill','g-.')

### 示例三（从复平面看幅频特性）

th = (0:127)/128\*2\*pi;

x = cos(th);

y = sin(th);

f = abs(fft(ones(10,1),128));

stem3(x,y,f','d','fill')

view([-65 30])

xlabel('Real')

ylabel('Imaginary')

zlabel('Amplitude')

title('Magnitude Frequency Response')

rotate3d on

## plot函数

plot(X,Y, 'str')可以用不同颜色、不同符号绘制曲线，其中'str'可以是下列参数选项的组合。

表 ‑2 plot绘图函数的线型参数

|  |  |  |
| --- | --- | --- |
| 线型 | 描述 | 表示的线条 |
| "-" | 实线 | Sample of solid line |
| "--" | 虚线 | Sample of dashed line |
| ":" | 点线 | Sample of dotted line |
| "-." | 点划线 | Sample of dash-dotted line, with alternating dashes and dots |

表 ‑3 plot绘图函数的标记参数

|  |  |  |
| --- | --- | --- |
| 标记 | 描述 | 生成的标记 |
| "o" | 圆圈 | Sample of circle marker |
| "+" | 加号 | Sample of plus sign marker |
| "\*" | 星号 | Sample of asterisk marker |
| "." | 点 | Sample of point marker |
| "x" | 叉号 | Sample of cross marker |
| "\_" | 水平线条 | Sample of horizontal line marker |
| "|" | 垂直线条 | Sample of vertical line marker |
| "square" | 方形 | Sample of square marker |
| "diamond" | 菱形 | Sample of diamond line marker |
| "^" | 上三角 | Sample of upward-pointing triangle marker |
| "v" | 下三角 | Sample of downward-pointing triangle marker |
| ">" | 右三角 | Sample of right-pointing triangle marker |
| "<" | 左三角 | Sample of left-pointing triangle marker |
| "pentagram" | 五角形 | Sample of pentagram marker |
| "hexagram" | 六角形 | Sample of hexagram marker |

表 ‑4 plot绘图函数的颜色参数

|  |  |  |  |
| --- | --- | --- | --- |
| 颜色名称 | 短名称 | RGB 三元组 | 外观 |
| "red" | "r" | [1 0 0] | Sample of the color red |
| "green" | "g" | [0 1 0] | Sample of the color green |
| "blue" | "b" | [0 0 1] | Sample of the color blue |
| "cyan" | "c" | [0 1 1] | Sample of the color cyan |
| "magenta" | "m" | [1 0 1] | Sample of the color magenta |
| "yellow" | "y" | [1 1 0] | Sample of the color yellow |
| "black" | "k" | [0 0 0] | Sample of the color black |
| "white" | "w" | [1 1 1] | Sample of the color white |

%指定线性的示例

x = 0:pi/100:2\*pi;

y1 = sin(x);

y2 = sin(x-0.25);

y3 = sin(x-0.5);

figure

plot(x,y1,x,y2,'--',x,y3,':')



%指定线型、颜色和标记

x = 0:pi/10:2\*pi;

y1 = sin(x);

y2 = sin(x-0.25);

y3 = sin(x-0.5);

figure

plot(x,y1,'g',x,y2,'b--o',x,y3,'c\*')



## subplot函数

subplot(m,n,p) 将当前图窗划分为 m×n 网格，并在 p 指定的位置创建坐标区。MATLAB按行号对子图位置进行编号。第一个子图是第一行的第一列，第二个子图是第一行的第二列，依此类推。如果指定的位置已存在坐标区，则此命令会将该坐标区设为当前坐标区。

[示例] 显示Chirp信号和正弦信号，及其幅度谱

close all;

clear all;

% 线性扫频信号：matlab中chirp函数可以产生一个在频率上波动的余弦信号。

t = 0:0.001:1; % 1 secs @ 1kHz sample rate

x\_chirp = chirp(t,1,1,100); % Start 1Hz, cross 50Hz at t=1 sec

x\_sin = sin(2\*pi\*t);

subplot(2,1,1);

hold on;

plot(t,x\_chirp);

plot(t,x\_sin);

legend('chirp','sin');

xlabel('Normalized Frequency (\times\pi rad/sample)');

ylabel('Magnitude');

title('信号');

subplot(2,1,2);

hold on;

plot(t, 20\*log10(abs(fft(x\_chirp))));

plot(t, 20\*log10(abs(fft(x\_sin))));

legend('chirp','sin');

xlabel('Normalized Frequency (\times\pi rad/sample)');

ylabel('Magnitude (dB)');

title('幅度谱');



## Matlab文件操作示例

问题1：如何在Matlab中读取数据文件

将如下格式的数据文件data.txt装载到内存中，并生成一个二维数组

1 0.0002 2

2 0.0004 3

3 0.3 4

输入命令“load -ascii data.txt”,此时数组data已经存在，输入命令“data”可以察看data

>> load -ascii data.txt

>> data

data =

1.0000 0.0002 2.0000

2.0000 0.0004 3.0000

3.0000 0.3000 4.0000

问题2：如何将数组的数据保存到数据文件？

Use the Save command.

Save the data in ASCII form using the save command with the -ascii option. For example,

>> A = rand(4,3);

>> save temp.dat A -ascii

creates an ASCII file called temp.dat containing

1.3889088e-001 2.7218792e-001 4.4509643e-001

2.0276522e-001 1.9881427e-001 9.3181458e-001

1.9872174e-001 1.5273927e-002 4.6599434e-001

6.0379248e-001 7.4678568e-001 4.1864947e-001

## 使用Matlab的信号处理工具箱

信号处理相关的工具箱包括：Signal Processing Toolbox、Audio Toolbox、DSP System Toolbox、Wavelet Toolbox。不同版本的Matlab中打开工具箱的入口略有差异，可在主界面上寻找入口，或通过联机帮助中介绍了解打开工具箱的方式。事实上，每个工具箱中都集成了一系列的函数，实际中某一类应用问题只需要使用工具箱中少数几个函数。

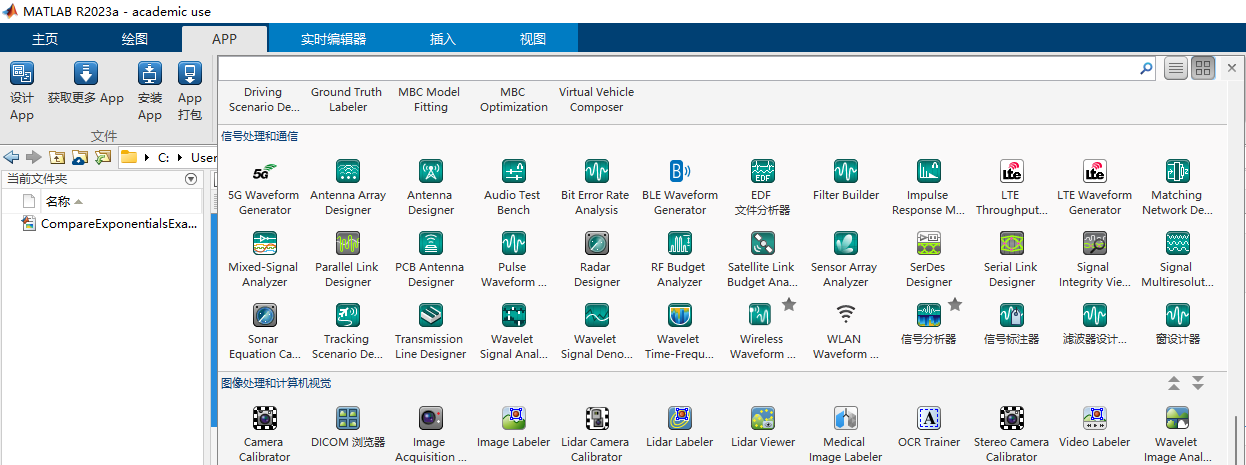


图 1‑1 信号处理工具箱示意

# Matlab中的DSP常用函数/工具

## 与信号产生、分析有关的函数

### 用行向量表示一个序列

Matlab中以矩阵形式保存数据。而DSP中常使用数组，对应为矩阵的行向量或列向量。例如，将 x 创建为由 0 和 2π 之间的线性间隔值组成的向量。在各值之间使用递增量 π/100。将 y 创建为 x 的正弦值。

x = 0:pi/100:2\*pi; % x 是 1×201 的矩阵

y = sin(x); % y 是 1×201 的矩阵

plot(x,y); %绘图

### 示例：产生一个理想采样信号序列

产生信号x(n),0<=n<=50

n=0:50; %定义序列的长度是51

A=444.128; %设置信号有关的参数

a=50\*sqrt(2.0)\*pi;

T=0.001; %采样率

w0=50\*sqrt(2.0)\*pi; %ω符号在MatLab中不能输入，用w代替

%pi是MATLAB定义的π，信号乘（点乘）可采用“.\*”

x=A\*exp(-a\*n\*T).\*sin(w0\*n\*T);

close all %清除已经绘制的x(n)图形

subplot(3,1,1);stem(x); %绘制x(n)的图形

% Matlab中字符串用英文符号“'”

title('理想采样信号序列'); %设置结果图形的标题

### 基于DFT绘制信号的幅度谱和相位谱

% 假设已经产生了一个长度为51的序列（1×51的矩阵）

k=-25:25;

W=(pi/12.5)\*k;

% n' 表示行向量n（1×51的矩阵）的转置

% k 表示行向量k（1×51的矩阵）

% n'\*k 表示 51×1的矩阵（n'）与1×51的矩阵（k）的乘法运算，结果是51×51的矩阵

% (exp(-j\*pi/12.5)) 是复数数值，.^ 表示逐个元素的指数运算

% (exp(-j\*pi/12.5)).^(n'\*k) 是51×51的复数矩阵

% 行向量x（1×51的矩阵）与 51×51的复数矩阵 做乘法，结果是 1×51的复数矩阵

X=x\*(exp(-j\*pi/12.5)).^(n'\*k); %请认真阅读上面几行注释关于本行代码的解析，此行代码充分利用了Matlab的矩阵运算功能

magX=abs(X); %绘制x(n)的幅度谱

subplot(3,1,2);

stem(magX);

title('理想采样信号序列的幅度谱');

angX=angle(X); %绘制x(n)的相位谱

subplot(3,1,3);stem(angX) ; title ('理想采样信号序列的相位谱')

### fourier Fourier transform

fourier([f](https://www.mathworks.com/help/releases/R2023a/symbolic/sym.fourier.html?doclanguage=zh-CN&nocookie=true&prodfilter=ML%20SL%205G%20AE%20AT%20AA%20AU%20DR%20AS%20BI%20BL%20C2%20CM%20VP%20CT%20CF%20DA%20DB%20DF%20DD%20DH%20NN%20HS%20DS%20ET%20EC%20FH%20IT%20FI%20PO%20FL%20GD%20GC%20HD%20ES%20IA%20IP%20OT%20IC%20LP%20LS%20MG%20ME%20CO%20MJ%20MR%20TE%20DX%20AM%20MP%20MB%20MT%20NV%20OP%20DM%20PD%20AR%20PW%20PM%20RA%20RL%20RQ%20RB%20RP%20RF%20RK%20RO%20RC%20RR%20SI%20TF%20SX%20SQ%20SG%20SB%20SE%20SS%20BT%20LD%20PS%20SH%20MS%20VR%20VV%20CI%20RT%20SK%20SD%20CV%20SO%20DV%20WT%20PL%20XP%20SR%20SZ%20HW%20EL%20SF%20ST%20SM%20ZC%20ID%20TA%20UV%20VE%20VN%20VT%20WA%20LH%20WB%20WL&docviewer=helpbrowser&docrelease=R2023a&s_cid=pl_webdoc&loginurl=https%3A%2F%2F127.0.0.1%3A31515%2Ftoolbox%2Fmatlab%2Flogin%2Fweb%2Findex.html%3Fsnc%3DH72RE1%26external%3Dtrue%26channel%3D__mlfpmc__&searchsource=mw&snc=IUF89N&container=jshelpbrowser#buon60m-f)) returns the [Fourier Transform](https://www.mathworks.com/help/releases/R2023a/symbolic/sym.fourier.html#buon7i0-2) of f.

### Matlab已有的波形生成函数

Matlab中可以使用 chirp 生成线性、二次和对数 chirp。使用 square、rectpuls 和 sawtooth 创建方波、矩形波和三角形波。

表 ‑1 Matlab已有的波形生成函数

|  |  |
| --- | --- |
| [chirp](https://www.mathworks.com/help/releases/R2023a/signal/ref/chirp.html) | Swept-frequency cosine |
| [diric](https://www.mathworks.com/help/releases/R2023a/signal/ref/diric.html) | Dirichlet or periodic sinc function |
| [gauspuls](https://www.mathworks.com/help/releases/R2023a/signal/ref/gauspuls.html) | Gaussian-modulated sinusoidal RF pulse |
| [gmonopuls](https://www.mathworks.com/help/releases/R2023a/signal/ref/gmonopuls.html) | Gaussian monopulse |
| [pulstran](https://www.mathworks.com/help/releases/R2023a/signal/ref/pulstran.html) | Pulse train |
| [randn](https://www.mathworks.com/help/releases/R2023a/matlab/ref/randn.html) | 正态分布的随机数 |
| [rectpuls](https://www.mathworks.com/help/releases/R2023a/signal/ref/rectpuls.html) | 采样的非周期性矩形 |
| [sawtooth](https://www.mathworks.com/help/releases/R2023a/signal/ref/sawtooth.html) | 锯齿波或三角波 |
| [sin](https://www.mathworks.com/help/releases/R2023a/matlab/ref/sin.html) | 参数的正弦，以弧度为单位 |
| [sinc](https://www.mathworks.com/help/releases/R2023a/signal/ref/sinc.html) | sinc 函数 |
| [square](https://www.mathworks.com/help/releases/R2023a/signal/ref/square.html) | 方波 |
| [stem](https://www.mathworks.com/help/releases/R2023a/matlab/ref/stem.html) | 绘制离散序列数据 |
| [tripuls](https://www.mathworks.com/help/releases/R2023a/signal/ref/tripuls.html) | Sampled aperiodic triangle |
| [vco](https://www.mathworks.com/help/releases/R2023a/signal/ref/vco.html) | Voltage-controlled oscillator |

* zeros & ones

x=zeros(1,50); % 产生 1×50 的矩阵，每个元素值都是0

x=ones(2,4); % 产生 2×4 的矩阵，每个元素值都是1

* rectpuls

采样的非周期性矩形

y = rectpuls(t,w) 生成一个宽度为 w 的矩形。

[示例]生成 200 毫秒的矩形脉冲，其采样率为 10 kHz，宽度为 20 毫秒。

fs = 10e3;

t = -0.1:1/fs:0.1;

w = 20e-3;

x = rectpuls(t,w);

### Matlab已有的窗函数

<https://www.mathworks.com/help/releases/R2023a/signal/ug/windows.html>

表 ‑1 Matlab已有的窗函数

|  |  |
| --- | --- |
| 窗 | 函数 |
| Bartlett-Hann 窗 | [barthannwin](https://www.mathworks.com/help/releases/R2023a/signal/ref/barthannwin.html) |
| Bartlett 窗 | [bartlett](https://www.mathworks.com/help/releases/R2023a/signal/ref/bartlett.html) |
| Blackman 窗 | [blackman](https://www.mathworks.com/help/releases/R2023a/signal/ref/blackman.html) |
| Blackman-Harris 窗 | [blackmanharris](https://www.mathworks.com/help/releases/R2023a/signal/ref/blackmanharris.html) |
| Bohman 窗 | [bohmanwin](https://www.mathworks.com/help/releases/R2023a/signal/ref/bohmanwin.html) |
| Chebyshev 窗 | [chebwin](https://www.mathworks.com/help/releases/R2023a/signal/ref/chebwin.html) |
| 平顶窗 | [flattopwin](https://www.mathworks.com/help/releases/R2023a/signal/ref/flattopwin.html) |
| 高斯窗 | [gausswin](https://www.mathworks.com/help/releases/R2023a/signal/ref/gausswin.html) |
| Hamming 窗 | [hamming](https://www.mathworks.com/help/releases/R2023a/signal/ref/hamming.html) |
| Hann 窗 | [hann](https://www.mathworks.com/help/releases/R2023a/signal/ref/hann.html) |
| Kaiser 窗 | [kaiser](https://www.mathworks.com/help/releases/R2023a/signal/ref/kaiser.html) |
| Nuttall's Blackman-Harris 窗 | [nuttallwin](https://www.mathworks.com/help/releases/R2023a/signal/ref/nuttallwin.html) |
| Parzen (de la Vallée-Poussin) 窗 | [parzenwin](https://www.mathworks.com/help/releases/R2023a/signal/ref/parzenwin.html) |
| 矩形窗 | [rectwin](https://www.mathworks.com/help/releases/R2023a/signal/ref/rectwin.html) |
| 锥形余弦窗 | [tukeywin](https://www.mathworks.com/help/releases/R2023a/signal/ref/tukeywin.html) |
| 三角形窗 | [triang](https://www.mathworks.com/help/releases/R2023a/signal/ref/triang.html) |

## 与传递函数相关的函数

### poly 具有指定根的多项式或特征多项式

p = poly(r)（其中 r 是向量）返回多项式的系数，其中多项式的根是 r 的元素。

### roots 多项式根

r = roots(p) 以列向量的形式返回 p 表示的多项式的根。输入 p 是一个包含 n+1 多项式系数的向量，以 xn 系数开头。0 系数表示方程中不存在的中间幂。例如：p = [3 2 -2] 表示多项式 3x2+2x−2。

### conv 卷积和多项式乘法

w = conv(u,v) 返回向量 u 和 v 的卷积。如果 u 和 v 是多项式系数的向量，对其卷积与将这两个多项式相乘等效。

### filtfilt Zero-phase digital filtering

y = filtfilt(b,a,x) performs zero-phase digital filtering by processing the input data x in both the forward and reverse directions. After filtering the data in the forward direction, the function reverses the filtered sequence and runs it back through the filter. The result has these characteristics:

* Zero phase distortion.
* A filter transfer function equal to the squared magnitude of the original filter transfer function.
* A filter order that is double the order of the filter specified by b and a.

### filter 1 维数字滤波器

y = filter(b,a,x) 使用由分子和分母系数 b 和 a 定义的有理传递函数 对输入数据 x 进行滤波。

在 MATLAB 中，[filter](https://www.mathworks.com/help/releases/R2023a/matlab/ref/filter.html) 函数会根据以下差分方程对数据 x 的向量进行滤波，该差分方程描述一个抽头延迟线滤波器。

a(1)y(n)=b(1)x(n)+b(2)x(n−1)+…+b(Nb)x(n−Nb+1) −a(2)y(n−1)−…−a(Na)y(n−Na+1)

在此方程中，a 和 b 是滤波器系数的向量，Na 是反馈滤波器阶数，Nb 是前馈滤波器阶数。n 是 x 的当前元素的索引。输出 y(n) 是 x 和 y 的当前元素和前面元素的线性组合。

### zp2tf, tf2zp零极点模型←→传递函数

Convert zero-pole-gain filter parameters to transfer function form

[b,a] = zp2tf(z,p,k) converts a factored transfer function representation

[z,p,k] = tf2zp(b,a) finds the matrix of zeros z, the vector of poles p, and the associated vector of gains k from the transfer function parameters b and a.

### freqs函数：模拟滤波器的频率响应

[示例] 系统传递函数为  的模拟滤波器，在MATLAB中可以用以下程序来实现：

a=[1 0.4 1]; b=[0.2 0.3 1]; %设置分子分母的系数

w=logspace(-1,1); % y = logspace(a,b) 生成一个由在 10^a 和 10^b（10 的 N 次幂）之间的 50 个对数间距点组成的行向量 y

freqs(b,a,w) %根据输入的参数绘制幅度谱和相位谱

### freqz 数字滤波器的频率响应

[h,w] = freqz(b,a,n) 返回数字滤波器的 n 点频率响应向量 h 和对应的角频率向量 w，滤波器的传递函数系数分别存储在 b 和 a 中。

[示例] 系统传递函数为  的数字滤波器，在MATLAB中可以用以下程序来实现：

a=[1 0.4 1];b=[0.2 0.3 1];

% freqz函数根据传递函数的参数绘制幅度谱和相位谱，得到0到π之间128个点处的频率响应

freqz(b,a,128)

[示例] 计算并显示由以下传递函数描述的三阶 IIR 低通滤波器的幅值响应：



将分子和分母表示为多项式卷积。求分布在整个单位圆上的 2023 个点上的频率响应。

b0 = 0.05634;

b1 = [1 1];

b2 = [1 -1.0166 1];

a1 = [1 -0.683];

a2 = [1 -1.4461 0.7957];

b = b0\*conv(b1,b2);

a = conv(a1,a2);

freqz(b,a,'whole',2023);

%[h,w] = freqz(b,a,'whole',2023);



图 2‑1 freqz结果界面

## 滤波器设计相关函数和工具

### IIR 滤波器设计相关函数

与 FIR 滤波器相比，IIR 滤波器的主要优点是，要满足同一组设定，它的滤波器阶数通常远远低于 FIR 滤波器。虽然 IIR 滤波器具有非线性相位，但 MATLAB® 软件中的数据处理通常是“离线”执行的，即整个数据序列在滤波之前是可用的。这允许采用非因果零相位滤波方法（通过 filtfilt 函数），消除 IIR 滤波器的非线性相位失真。

表 ‑2 Matlab工具箱提供的IIR设计相关函数（部分）

|  |  |
| --- | --- |
| **滤波器方法** | **滤波器函数** |
| 模拟原型 | * 整体设计函数：[besself](https://www.mathworks.com/help/releases/R2023a/signal/ref/besself.html)、[butter](https://www.mathworks.com/help/releases/R2023a/signal/ref/butter.html)、[cheby1](https://www.mathworks.com/help/releases/R2023a/signal/ref/cheby1.html)、[cheby2](https://www.mathworks.com/help/releases/R2023a/signal/ref/cheby2.html)、[ellip](https://www.mathworks.com/help/releases/R2023a/signal/ref/ellip.html) * 阶估算函数：[buttord](https://www.mathworks.com/help/releases/R2023a/signal/ref/buttord.html)、[cheb1ord](https://www.mathworks.com/help/releases/R2023a/signal/ref/cheb1ord.html)、[cheb2ord](https://www.mathworks.com/help/releases/R2023a/signal/ref/cheb2ord.html)、[ellipord](https://www.mathworks.com/help/releases/R2023a/signal/ref/ellipord.html) * 低通模拟原型函数：[besselap](https://www.mathworks.com/help/releases/R2023a/signal/ref/besselap.html)、[buttap](https://www.mathworks.com/help/releases/R2023a/signal/ref/buttap.html)、[cheb1ap](https://www.mathworks.com/help/releases/R2023a/signal/ref/cheb1ap.html)、[cheb2ap](https://www.mathworks.com/help/releases/R2023a/signal/ref/cheb2ap.html)、[ellipap](https://www.mathworks.com/help/releases/R2023a/signal/ref/ellipap.html) * 频率变换函数：[lp2bp](https://www.mathworks.com/help/releases/R2023a/signal/ref/lp2bp.html)、[lp2bs](https://www.mathworks.com/help/releases/R2023a/signal/ref/lp2bs.html)、[lp2hp](https://www.mathworks.com/help/releases/R2023a/signal/ref/lp2hp.html)、[lp2lp](https://www.mathworks.com/help/releases/R2023a/signal/ref/lp2lp.html) * 滤波器离散化函数：[bilinear](https://www.mathworks.com/help/releases/R2023a/signal/ref/bilinear.html)、[impinvar](https://www.mathworks.com/help/releases/R2023a/signal/ref/impinvar.html) |
| 直接设计 | [yulewalk](https://www.mathworks.com/help/releases/R2023a/signal/ref/yulewalk.html) |

### FIR 滤波器设计相关函数

FIR 滤波器与无限持续时间脉冲响应 (IIR) 滤波器相比，具有有限持续时间脉冲响应的数字滤波器（全零或 FIR 滤波器）既有优点又有缺点。

表 ‑3 Matlab工具箱提供的FIR设计相关函数（部分）

|  |  |  |
| --- | --- | --- |
| 滤波器设计方法 | 说明 | 滤波器函数 |
| 加窗 | 对指定的矩形滤波器的截断傅里叶逆变换应用加窗 | [fir1](https://www.mathworks.com/help/releases/R2023a/signal/ref/fir1.html), [fir2](https://www.mathworks.com/help/releases/R2023a/signal/ref/fir2.html), [kaiserord](https://www.mathworks.com/help/releases/R2023a/signal/ref/kaiserord.html) |
| 多频带（包含过渡带） | 对频率范围的子带使用等波纹或最小二乘方法 | [firls](https://www.mathworks.com/help/releases/R2023a/signal/ref/firls.html)、[firpm](https://www.mathworks.com/help/releases/R2023a/signal/ref/firpm.html)、[firpmord](https://www.mathworks.com/help/releases/R2023a/signal/ref/firpmord.html) |
| 约束最小二乘 | 根据最大误差约束，在整个频率范围内最小化平方积分误差 | [fircls](https://www.mathworks.com/help/releases/R2023a/signal/ref/fircls.html), [fircls1](https://www.mathworks.com/help/releases/R2023a/signal/ref/fircls1.html) |
| 任意响应 | 任意响应，包括非线性相位和复滤波器 | [cfirpm](https://www.mathworks.com/help/releases/R2023a/signal/ref/cfirpm.html) |
| 升余弦 | 具有平滑正弦过渡的低通响应 | [rcosdesign](https://www.mathworks.com/help/releases/R2023a/signal/ref/rcosdesign.html) |

#### 在MATLAB中产生窗函数

1. 矩形窗（Rectangle Window）

调用格式：w=boxcar(n)，根据长度n产生一个矩形窗w。

1. 三角窗（Triangular Window）

调用格式：w=triang(n) ，根据长度n产生一个三角窗w。

1. 汉宁窗（Hanning Window）

调用格式：w=hanning(n) ，根据长度n产生一个汉宁窗w。

1. 海明窗（Hamming Window）

调用格式：w=hamming(n) ，根据长度n产生一个海明窗w。

1. 布拉克曼窗（Blackman Window）

调用格式：w=blackman(n) ，根据长度n产生一个布拉克曼窗w。

（6）恺撒窗（Kaiser Window）

调用格式：w=kaiser(n,beta) ，根据长度n和影响窗函数旁瓣的β参数产生一个恺撒窗w。

### FVTool: Filter Visualization Tool

>> [b,a] = ellip(6,3,50,300/500);

>> fvtool(b,a)



图 2‑2 fvtool界面

### FilterDesign 工具

滤波器设计工具是 Signal Processing Toolbox 中一个功能强大的图形用户界面 (GUI)，用于设计和分析滤波器。

滤波器设计工具使您能够通过设置滤波器性能设定、从 MATLAB® 工作区导入滤波器或通过添加、移动或删除极点和零点来快速设计数字 FIR 或 IIR 滤波器。滤波器设计工具还提供用于分析滤波器的工具，例如幅值和相位响应图以及零极点图。

>> filterDesigner

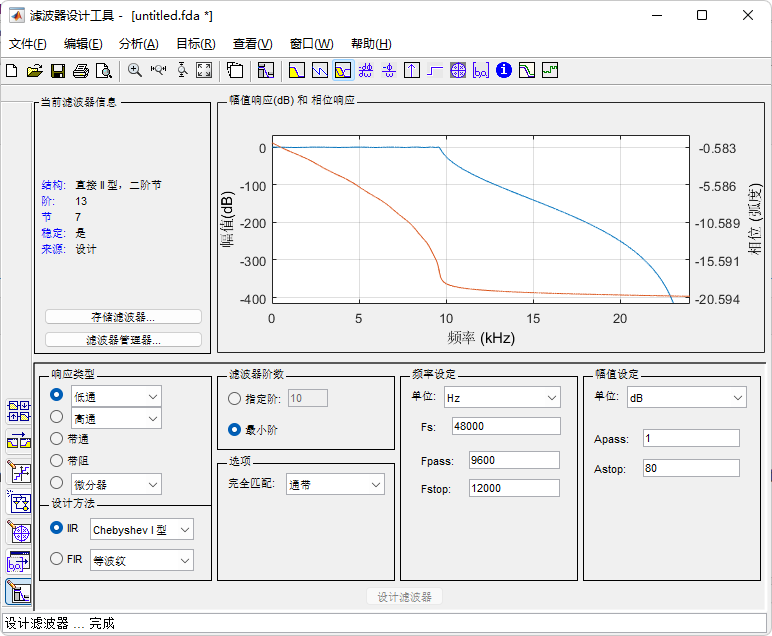


图 2‑3 滤波器设计工具界面

# Matlab示例代码

## Matlab分析信号基本示例

### 非周期信号的CFT

close all

clear all

syms x % 创建符号变量x

f\_exp = exp(-2\*x^2); % 指数函数

FT\_exp = fourier(f\_exp) % 指数函数的 Fourier transform

subplot(2,1,1); hold on;

fplot(f\_exp,[-9,9]) % 绘制指数函数

title('原信号');

legend('$e^{-2{x^2}}$', 'Interpreter','latex');

subplot(2,1,2); hold on;

fplot(FT\_exp,[-10,10]);

title('连续傅里叶变换CFT');

legend('$\mathcal{F}\{e^{-2{x^2}}\}$','Interpreter','latex');



### 非周期序列的DTFT

close all; clear all;

syms w;

DTFT\_RN5 = exp(1i\*w) + exp(1i\*2\*w) + exp(1i\*4\*w)+ exp(1i\*5\*w)

w = 0:0.01:8\*pi;

hold on;

plot(abs(eval(DTFT\_RN5)))

set(gca,'xlim',[0,8\*100\*pi]);%设置x轴坐标范围

set(gca,'XTickLabel',{'0','2\pi','4\pi','6\pi','8\pi','10\pi'});

ylabel('Magnitude: $|H(ej\omega)|$','Interpreter','latex');



## IIR滤波器设计

### 直接设计数字滤波器

低通，采样频率为1Hz，通带临界频率fp =0.2Hz，通带内衰减小于1dB（αp=1）；阻带临界频率fs=0.3Hz，阻带内衰减大于25dB（αs=25）。设计一个数字滤波器满足以上参数。

% Matlab使用归一化的频率参数（临界频率除以采样频率的1/2）。

%这样临界频率参数的取值范围在0和1之间，1代表Fs/2（用角频率表示的时候对应π）

FS=1

[n,Wn]=buttord(0.2/(FS/2),0.3/( FS /2),1,25);

[b,a]=butter(n,Wn);

freqz(b,a,512, FS);



### 脉冲相应不变法

低通，采样频率为1Hz，通带临界频率fp =0.2Hz，通带内衰减小于1dB（αp=1）；阻带临界频率fs=0.3Hz，阻带内衰减大于25dB（αs=25）。设计一个数字滤波器满足以上参数。

FS=1

[n,Wn]=buttord(0.2\*2\*pi,0.3\*2\*pi,1,25,'s'); %临界频率采用角频率表示

[b,a]=butter(n,Wn,'s');

%freqs(b,a) %设计模拟的

[bz,az]=impinvar(b,a,FS); %映射为数字的

freqz(bz,az,512,FS)



### 双线性变换法

#### 双线性变换设计Butterworth低通示例1

低通，采样频率为1Hz，通带临界频率fp =0.2Hz，通带内衰减小于1dB（αp=1）；阻带临界频率fs=0.3Hz，阻带内衰减大于25dB（αs=25）。设计一个数字滤波器满足以上参数。

FS=1

%通带、阻带截止频率

Fl=0.2;Fh=0.3;

%频率预畸

wp=(Fl/FS)\*2\*pi; %临界频率采用角频率表示

ws=(Fh/FS)\*2\*pi; %临界频率采用角频率表示

OmegaP=2\*FS\*tan(wp/2);

OmegaS=2\*FS\*tan(ws/2);

[n,Wn]=buttord(OmegaP,OmegaS,1,25,'s');

[b,a]=butter(n,Wn,'s');

%freqs(b,a) %设计模拟的

[bz,az]=bilinear(b,a,FS); %映射为数字的

freqz(bz,az,512,FS)

%freqz(bz,az,512,FS,'whole')



#### 双线性变换实现ButterWorth低通示例2

%采样频率10Hz，通带截止频率fp=3Hz，阻带截止频率fs=4Hz

%通带衰减小于1dB，阻带衰减大于20dB

%使用双线性变换法由模拟滤波器原型设计数字滤波器

T=0.1; FS=1/T;

fp=3;fs=4;

wp=fp/FS\*2\*pi;

ws=fs/FS\*2\*pi;

Rp = 1; % 通带衰减

As = 20; % 阻带衰减

% 频率预畸

OmegaP = (2/T)\*tan(wp/2); % Prewarp Prototype Passband freq

OmegaS = (2/T)\*tan(ws/2); % Prewarp Prototype Stopband freq

%设计butterworth低通滤波器原型

N = ceil((log10((10^(Rp/10)-1)/(10^(As/10)-1)))/(2\*log10(OmegaP/OmegaS)));

OmegaC = OmegaP/((10^(Rp/10)-1)^(1/(2\*N)));

[z,p,k] = buttap(N); %获取零极点参数

p = p\*OmegaC;

k = k\*OmegaC^N;

B = real(poly(z));

b0 = k;

cs = k\*B;

ds = real(poly(p));

[b,a] = bilinear(cs,ds,FS); % 双线性变换

freqz(b,a,512,FS);



#### 双线性变换法实现Chebyshev低通（I型）

采样频率为10Hz，设计一个数字低通滤波器，要求其通带临界频率，通带内衰减小于1dB()，阻带临界频率，阻带内衰减大于15dB（）。

T=0.1; FS=1/T;

fp=3;fs=4;

Rp = 1;

As = 15;

% 频率预畸

wp=fp/FS\*2\*pi;

ws=fs/FS\*2\*pi;

OmegaP = (2/T)\*tan(wp/2);

OmegaS = (2/T)\*tan(ws/2);

%设计Chebyshev低通滤波器原型

ep = sqrt(10^(Rp/10)-1);

A = 10^(As/20);

OmegaC = OmegaP;

OmegaR = OmegaS/OmegaP;

g = sqrt(A\*A-1)/ep;

N = ceil(log10(g+sqrt(g\*g-1))/log10(OmegaR+sqrt(OmegaR\*OmegaR-1)));

[z,p,k] = cheb1ap(N,Rp); %获取零极点参数

a = real(poly(p));

aNn = a(N+1);

p = p\*OmegaC;

a = real(poly(p));

aNu = a(N+1);

k = k\*aNu/aNn;

b0 = k;

B = real(poly(z));

b = k\*B;

[bz,az] = bilinear(b,a,FS); % 双线性变换

freqz(bz,az,200,FS,'whole');



#### 双线形变换法设计chebyshev高通

高通，采样频率为10Hz，通带临界频率fp =4Hz，通带内衰减小于0.8dB（αp=1）；阻带临界频率fs=3Hz，阻带内衰减大于20dB（αs=25）。设计一个数字滤波器满足以上参数。

%双线形变换法设计chebyshev高通滤波器

FS=10; T=1/FS;

fp=4;fs=3;

wp=fp/FS\*2\*pi;

ws=fs/FS\*2\*pi;

OmegaP=2\*FS\*tan(wp/2);

OmegaS=2\*FS\*tan(ws/2);

[n,Wn]=cheb1ord(OmegaP,OmegaS,0.8,20,'s')

[b,a]=cheby1(n,0.8,Wn,'high','s');

%freqs(b,a) %设计模拟的

[bz,az]=bilinear(b,a,FS); %映射为数字的

freqz(bz,az,512,FS)



#### 双线性变换实现ButterWorth带通

%采样频率100Hz，Wpl=20Hz，Wph=30Hz，Wsl=15Hz，Wsh=35Hz，

% 频率/采样频率\*2\*pi

%Wpl=0.4\*pi，Wph=0.6\*pi，Wsl=0.2\*pi，Wsh=0.8\*pi，

%Rp=1;As=20

T = 1; Fs = 1/T; % Set T=1

%T=2;

Wpl=tan(0.4/2/Fs\*pi);

Wph=tan(0.6/2/Fs\*pi);

Wsl=tan(0.3/2/Fs\*pi);

Wsh=tan(0.7/2/Fs\*pi);

Rp = 1; % Passband ripple in dB

As = 40; % Stopband attenuation in dB

%计算模拟低通原型的参数

OmigaP=Wph-Wpl; %低通滤波器通带截止频率

OmigaS=Wsh-Wsl; %低通滤波器通带截止频率

[N,Wn]=buttord(OmigaP,OmigaS,Rp,As,'s');

[B,A]=butter(N,Wn,'s');

[BT,AT]=lp2bp(B,A,sqrt(Wph\*Wpl),Wph-Wpl);

[b,a]=bilinear(BT,AT,Fs);

freqz(b,a,200,'whole');

%H=freqz(b,a,200,'whole');

%plot(abs(H));



## 模拟频带变换

### 模拟低通变换为模拟高通

close all; clear all;

fp=.2; fs=.3;

Wp=fp\*2\*pi;

Ws=fs\*2\*pi;

alphap=1;

alphas=20;

[N, Wn] = cheb1ord(Wp, Ws, alphap, alphas,'s')

[B, A] = cheby1(N, alphap, Wn, 's');

figure(1);

freqs(B,A);

[BT, AT] = lp2hp(B, A, Wp);

figure(2);

freqs(BT,AT);

### 低通变换为带通

%fpl=20Hz，fph=30Hz，fsl=15Hz，fsh=35Hz，

%Rp=1;As=20

Fs = 100; T=1/Fs;

fpl=20;fph=30;fsl=15;fsh=35;

Wpl=fpl/Fs\*2\*pi;

Wph=fph/Fs\*2\*pi;

Wsl=fsl/Fs\*2\*pi;

Wsh=fsh/Fs\*2\*pi;

Rp = 1;

As = 40;

OmigaP=Wph-Wpl; %低通滤波器通带截止频率

OmigaS=Wsh-Wsl; %低通滤波器通带截止频率

[N,Wn]=buttord(OmigaP,OmigaS,Rp,As,'s');

[B,A]=butter(N,Wn,'s');

[BT,AT]=lp2bp(B,A,sqrt(Wph\*Wpl),Wph-Wpl);

close all;

freqs(B,A);

figure(2);

freqs(BT,AT);

## 窗函数法设计FIR滤波器

### 低通滤波器实现

[例] 设计一个长度为8的线性相位FIR滤波器。

其理想幅频特性满足

Window=boxcar(8); %长度为8的矩形窗Window

b=fir1(7,0.4,Window);

freqz(b,1)



Window=blackman(8); %长度为8的布拉克曼窗Window

b=fir1(7,0.4,Window);

freqz(b,1)



### 带通滤波器实现

[例] 设计线性相位带通滤波器，设计指标：长度N=16 ，上下边带截止频率分别为W1= 0.3π，w2=0.5π。

Window=blackman(16);

b=fir1(15,[0.3 0.5],Window);

freqz(b,1)



### MultiBand实现

b = fir1(48,[0.2 0.3 .45 .55 0.7 0.8]);

freqz(b,1,512)



## 频率抽样法设计FIR滤波器

### 低通滤波器（基本实现方式）

close all;clear all;

N = 0; %滤波器长度N

tau = (N-1)/2;

n = 0:N-1;

Hdr = [1,1,0,0]; wdn = [0,0.25,0.25,1];

Wn = (2\*pi/N)\*n;

Hrs = [1,1,1,zeros(1,15),1,1];

k1 = 0:floor((N-1)/2);

k2 = floor((N-1)/2)+1:N-1;

angH = [-tau\*(2\*pi)/N\*k1, tau\*(2\*pi)/N\*(N-k2)]; %H(k)的相位

H = Hrs.\*exp(j\*angH);

h = real(ifft(H,N));

freqz(h,1)



### 低通滤波器（增加过渡带抽样点）

N = 20; %滤波器长度N

tau = (N-1)/2;

n = 0:N-1;

Hdr = [1,1,0,0]; wdn = [0,0.25,0.25,1];

Wn = (2\*pi/N)\*n;

Hrs = [1,1,0.2,zeros(1,15),0.2,1]; %增加过渡带抽样点

%Hrs = [1,1,1,zeros(1,15),1,1];

k1 = 0:floor((N-1)/2);

k2 = floor((N-1)/2)+1:N-1;

angH = [-tau\*(2\*pi)/N\*k1, tau\*(2\*pi)/N\*(N-k2)]; %H(k)的相位

H = Hrs.\*exp(j\*angH);

h = real(ifft(H,N));

freqz(h,1)

