

华中科技大学

生物医学数字信号处理实验报告

实验 3 数字滤波器

学院

工程科学学院

班级

工程科学学院（生医）1701 班

姓名

汪能志

学号

U201713082

指导老师

全廷伟

2020 年 9 月 4 日

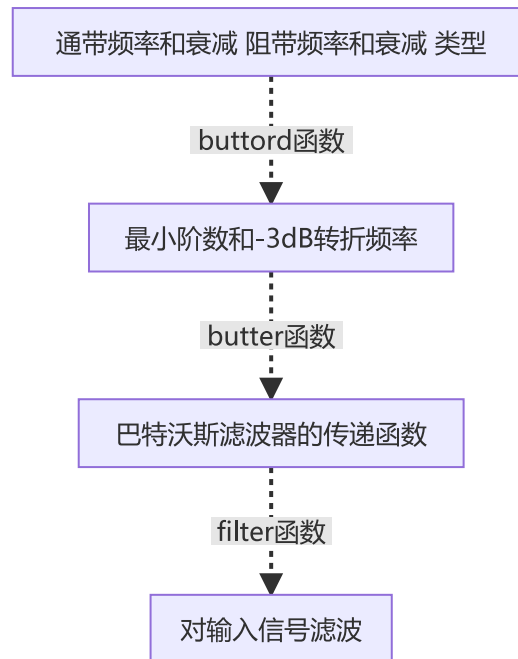
目录

1. 实验原理.....	1
1.1. 数字滤波器的设计流程.....	1
1.2. 模拟滤波器到数字滤波器的映射.....	1
1.3. 双线性变换.....	2
1.4. 归一化巴特沃斯低通滤波器.....	3
1.5. 滤波器阶数和截止频率计算.....	3
1.5.1. 低通滤波器.....	4
1.5.2. 高通滤波器.....	4
1.5.3. 带通滤波器.....	4
1.6. 模拟滤波器去归一化.....	5
1.7. 滤波器函数的实现.....	5
2. MATLAB 实现.....	6
2.1. 计算巴特沃斯滤波器的阶数和转折频率（ <code>buttord</code> ）.....	6
2.2. 计算数字滤波器的传递函数（ <code>butter</code> ）.....	7
2.3. 滤波器函数（ <code>filter</code> ）.....	9
3. 实验结果.....	10
3.1. 低通滤波器.....	10
3.2. 高通滤波器.....	11
3.3. 带通滤波器.....	12
3.4. 带阻滤波器.....	13
3.5. 根据传递函数计算输出信号.....	14
4. 实验总结.....	16

1. 实验原理

1.1. 数字滤波器的设计流程

由于模拟滤波器的设计较为成熟，因此本次实验中设计数字滤波器的方法为：先设计模拟滤波器，再将模拟滤波器的系统函数映射为数字滤波器的系统函数。



1.2. 模拟滤波器到数字滤波器的映射

常用的将模拟滤波器转换为数字滤波器的方法有冲激响应不变法和双线性变换法。

冲激响应不变法从单位冲激响应出发，目标是使数字滤波器的单位冲激响应 $h[n]$ 尽可能逼近模拟滤波器的单位冲激响应 $h(t)$ 即 $h[n] = h_a(nT)$ 。在拉普拉斯变换域和 z 变换域上，满足变换关系：

$$\begin{cases} z = e^{sT} \\ H_a(s) = \sum_{k=1}^N \frac{A_k}{s - s_k} \Rightarrow H(z) = \sum_{k=1}^N \frac{A_k}{1 - e^{s_k T} z^{-1}} \end{cases} \quad (3-1)$$

在变换前后， s 平面和 z 平面不算单值对应的。在 s 平面上相距 $2j\pi$ 的两个点，在变换后的 z 平面上是重合的。因此，冲激响应不变法会出现频率混叠现象。

从时域的角度来看，可以认为冲激响应不变法的思路是：对模拟滤波器的单位冲激响应 $h(t)$ 以间隔 T 进行采样，将采样信号作为数字滤波器的单位冲激响应。此时对应的采样率为 $\Omega_s = \frac{2\pi}{T}$ 。根据奈奎斯特采样定理，模拟滤波器的单位冲激响应必须是截止频率不大于 $\Omega_h = \frac{\pi}{T}$ 的低通信号，否则就会出现频谱混叠的情况。

为此，需要引入双线性变换，双线性变换法得到的数字滤波器的频率响应和模拟滤波器的频率响应相似，其模拟频率 Ω 和数字频率 ω 之间是单值变换。

在 MATLAB 中，数字频率定义为归一化频率，对于以 f_s 采样的模拟信号，有 $\omega = \frac{\Omega}{\pi \cdot f_s}$ 。

为了方便和 MATLAB 中提供的函数进行对比测试，以下讨论内容和 MATLAB 中的定义保持一致。

双线性变换法的映射关系为：

$$s = \frac{2}{T} \cdot \frac{1 - z^{-1}}{1 + z^{-1}} \quad (3-2)$$

其中 T 为数字信号的采样周期，即采样率的倒数。在实际计算中， T 的取值不影响最终的计算结果（因为数字序列对应的频率会随着采样率的变化同步变化）。为简化计算，一般取 $T = 2$ 。

由于其模拟频率和数字频率之间是单值映射，因此不会和冲激响应不变法一样出现频率混叠。但是，由于其变换不是线性变换，因此会出现频率畸变现象。在设计模拟滤波器式，需要引入预畸变进行补偿。

$$\Omega = \frac{2}{T} \tan \frac{\pi \omega}{2} \quad (3-3)$$

在 MATLAB 的库函数 butter.m 中，进行频率预畸变的代码如下：

```
71 % step 1: get analog, pre-warped frequencies
72 if ~analog
73     fs = 2;
74     u = 2*fs*tan(pi*Wn/fs);
75 else
76     u = Wn;
77 end
```

在将 73 行的 fs 修改为其他值后，会得到错误的结果。这与式 3-2 的结论：数字信号采样周期（采样率）不影响最终系统函数矛盾。这里应该是第 74 行代码出现错误，其正确形式应为：

```
74 u = 2*fs*tan(pi*Wn/2);
```

其将常量 2 错误写为变量 fs，因此只有在 fs=2 时，才能得到正确的结果。

1.3. 双线性变换

模拟巴特沃斯滤波器和数字巴特沃斯滤波器的系统函数都可以表示为零极点增益形式。在变换前后，s 平面的极点 s_{pi} 和 z 平面的极点 z_{pi} 一一对应；s 平面的零点 s_{zi} 和 z 平面的零点 z_{zi} 一一对应。因此只需要计算 s 平面的零点、极点和增益变换到 z 平面后的值，就可以得到数字滤波器的系统函数。

由于差分（微分）方程和系统函数之间可以转换。因此该双线性变换公式可以用在任意线性时不变系统上。

变换公式如下式 3-4：

$$\left\{ \begin{array}{l} H(s) = k_s \cdot \frac{\prod_{i=1}^M (s - s_{zi})}{\prod_{i=1}^N (s - s_{pi})} \xleftrightarrow[\text{Transform}]{\text{Bilinear}} H(z) = k_z \cdot \frac{\prod_{i=1}^M (z - z_{zi}) \cdot \prod_{i=1}^{M-N} (z + 1)}{\prod_{i=1}^N (z - z_{pi})} \\ z = \frac{1 + \frac{T}{2} \cdot s}{1 - \frac{T}{2} \cdot s} \quad z = z_{zi}, z_{pi} \quad s = s_{zi}, s_{pi} \\ k_z = k_s \cdot \frac{\prod_{i=1}^M (\frac{2}{T} - s_{zi})}{\prod_{i=1}^N (\frac{2}{T} - s_{pi})} \end{array} \right. \quad (M \leq N) \quad (3-4)$$

1.4. 归一化巴特沃斯低通滤波器

模拟巴特沃斯低通滤波器是全极点系统，其幅度平方函数为：

$$|H_a(j\Omega)|^2 = \frac{1}{1 + (\frac{\Omega}{\Omega_c})^{2N}} \quad (3-5)$$

将其拓展到拉普拉斯变换域，有：

$$|H_a(s)|^2 = H_a(s)H_a(-s) = \frac{1}{1 + (\frac{s}{j\Omega_c})^{2N}} \quad (3-6)$$

式 3-8 的极点为：

$$s_k = \Omega_c e^{j(\frac{1}{2} + \frac{2k-1}{2N})\pi} \quad k = 1, 2, \dots, 2N \quad (3-7)$$

这些极点均匀分布在 s 平面上，半径为 Ω_c 的巴特沃斯圆上。且其中 $H_a(s)$ 和 $H_a(-s)$ 的极点两两一组，关于虚轴对称。为了使系统稳定， $H_a(s)$ 的极点需要全部在虚轴左侧，共有 N 个。

1.5. 滤波器阶数和截止频率计算

计算流程为：

1. 将数字频率预畸变为模拟频率；
2. 计算模拟滤波器的阶数和 3dB 转折频率；
3. 将模拟频率反畸变为数字频率。

对于阻带衰减 R_s ，通带衰减 R_p ，阻带频率 ω_s 和通带频率 ω_p 的数字滤波器，将数字频率畸变为对应的模拟频率 Ω_s 和 Ω_p 后，以下将讨论在低通、高通和带通的情况下的计算方法。

1.5.1. 低通滤波器

$$\begin{cases} g = \frac{10^{0.1 \cdot R_s - 1}}{10^{0.1 \cdot R_p - 1}} \\ \lambda_s = \frac{\Omega_s}{\Omega_p} \\ N \geq \frac{\log(g)}{2 \cdot \log(\lambda_s)} \end{cases} \quad (3-8)$$

以恰好满足通带条件和恰好满足阻带条件进行计算，可以得到两个截止频率的表达式：

$$\Omega_{c0} \geq \frac{\Omega_p}{\sqrt[2N]{10^{0.1 \cdot R_p - 1}}} = \Omega_{cp} \quad (3-9)$$

$$\Omega_{c0} \leq \frac{\Omega_s}{\sqrt[2N]{10^{0.1 \cdot R_s - 1}}} = \Omega_{cs} \quad (3-10)$$

为了获得更大的阻带衰减，一般使用 $\Omega_{c0} = \Omega_{cs}$ 。

对于低通滤波器，有：

$$\Omega_c = \Omega_p \cdot \Omega_{c0} \quad (3-11)$$

1.5.2. 高通滤波器

$$\begin{cases} g = \frac{10^{0.1 \cdot R_s - 1}}{10^{0.1 \cdot R_p - 1}} \\ \lambda_s = \frac{\Omega_p}{\Omega_s} \\ N \geq \frac{\log(g)}{2 \cdot \log(\lambda_s)} \end{cases} \quad (3-12)$$

类似低通滤波器计算截止频率，有：

$$\Omega_c = \frac{\Omega_p}{\Omega_{c0}} \quad (3-13)$$

1.5.3. 带通滤波器

$$\begin{cases} B_p = \Omega_{p2} - \Omega_{p1} & \Omega_{p0}^2 = \Omega_{p2} \Omega_{p1} \\ \frac{\Omega_{s1}^2 - \Omega_{p1}^2}{\Omega_{s1} B_p} & \frac{\Omega_{s2}^2 - \Omega_{p2}^2}{\Omega_{s2} B_p} & \lambda_s = \overline{\Omega_s} = \min(|\overline{\Omega_{s1}}|, |\overline{\Omega_{s2}}|) \\ g = \frac{10^{0.1 \cdot R_s - 1}}{10^{0.1 \cdot R_p - 1}} \\ N \geq \frac{\log(g)}{2 \cdot \log(\lambda_s)} \end{cases} \quad (3-14)$$

$$\Omega_c = \pm \frac{\Omega_{c0} \cdot (\Omega_2 - \Omega_1)}{2} + \sqrt{\frac{\Omega_{c0}^2}{4} \cdot (\Omega_2 - \Omega_1)^2 + (\Omega_2 \cdot \Omega_1)} \quad (3-15)$$

带阻滤波器的阶数和截止频率求解较为复杂,在之后的实验中直接调用 MATLAB 中的 buttord 函数。

1.6. 模拟滤波器去归一化

归一化模拟巴特沃斯低通滤波器是全极点滤波器,其系统函数可以表示为如下的形式:

$$H(\bar{s}) = k \cdot \frac{1}{\prod_{i=1}^N (\bar{s} - p_i)} = k \cdot \prod_{i=1}^N \frac{1}{(\bar{s} - p_i)} \quad (3-16)$$

如果 Ω_c 为去归一化后的模拟滤波器的 3dB 转折频率,则对于不同类型的模拟巴特沃斯滤波器,去归一化前后拉普拉斯变换域自变量 s 和系统函数的变换公式如下。

去归一化到低通滤波器:

$$\bar{s} = \frac{s}{\Omega_c} \Rightarrow \frac{1}{\bar{s} - p_i} = \Omega_c \cdot \frac{1}{s - \Omega_c \cdot p_i} \quad (3-17)$$

去归一化到高通滤波器:

$$\bar{s} = \frac{\Omega_c}{s} \Rightarrow \frac{1}{\bar{s} - p_i} = -p_i \cdot \frac{s}{s - \frac{\Omega_c}{p_i}} \quad (3-18)$$

去归一化到带通滤波器:

$$\bar{s} = \frac{s^2 + \Omega_0^2}{B_\omega \cdot s} \Rightarrow \frac{1}{\bar{s} - p_i} = B_\Omega \cdot \frac{s}{s^2 - B_\Omega \cdot p_i \cdot s + \Omega_0^2} \quad (3-19)$$

$$\Omega_0 = \sqrt{\Omega_{c1} \Omega_{c2}} \quad B_\Omega = |\Omega_{c1} - \Omega_{c2}|$$

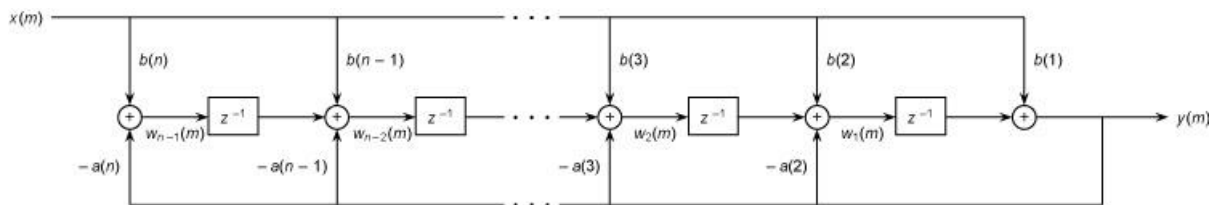
去归一化到带阻滤波器:

$$\bar{s} = \frac{s \cdot B_\Omega}{s^2 + \Omega_0^2} \Rightarrow \frac{1}{\bar{s} - p_i} = \left(-\frac{1}{p_i} \right) \cdot \frac{s^2 + \Omega_0^2}{s^2 - B_\Omega \cdot p_i \cdot s + \Omega_0^2} \quad (3-20)$$

$$\Omega_0 = \sqrt{\Omega_{c1} \Omega_{c2}} \quad B_\Omega = |\Omega_{c1} - \Omega_{c2}|$$

根据以上公式,就可以计算去归一化后模拟滤波器的系统函数的零点、极点和增益,并得到其系统函数的表达式。再使用式 3-5 和 3-6 进行双线性变换,即可得到对应的数字滤波器的系统函数。

1.7. 滤波器函数的实现



通过传递函数和输入信号计算输出信号的实现参考了数字信号处理教程和 MATLAB 的帮助文档中说明的方法，使用直接 II 型 IIR 滤波器的转置结构实现。

2. MATLAB 实现

2.1. 计算巴特沃斯滤波器的阶数和转折频率（buttord）

```
function [order, wc] = my_buttord(wp, ws, Rp, Rs)
% in MATLAB $ w = W / (pi * Fs) $ not $ w = W / Fs $ in textbook
% 滤波器类型
wp = sort(abs(wp));
ws = sort(abs(ws));
if numel(wp) == 1
    if wp < ws
        % 低通
        ftype = 1;
    elseif wp > ws
        % 高通
        ftype = 2;
    end
elseif numel(wp) == 2
    if wp(1) > ws(1) && wp(2) < ws(2)
        % 带通
        ftype = 3;
    elseif wp(1) < ws(1) && wp(2) > ws(2)
        % 带阻，直接调用库函数
        ftype = 4;
    end
end

if ftype ~= 4
    % 根据数字频率计算模拟频率，进行预畸变
    T = 2;
    Wp = (2 / T) .* tan(pi .* wp ./ 2);
    Ws = (2 / T) .* tan(pi .* ws ./ 2);

    if ftype == 1
        % 低通
        Wst = Ws / Wp;
    elseif ftype == 2
        % 高通
        Wst = Wp / Ws;
    elseif ftype == 3
        % 带通
```



```

        Wst = (Ws.^2 - Wp(1) * Wp(2)) ./ (Ws * (Wp(1) - Wp(2)));
        Wst = min(abs(Wst));
    elseif ftype == 4
        % 带阻，直接调用库函数
    end

    % 计算最小阶数
    G = (10^(0.1 * Rs) - 1) / (10^(0.1 * Rp) - 1);
    order = ceil(log(G) / (2 * log(Wst)));
    % 计算截止频率
    W0 = Wst / ((10^(0.1 * Rs) - 1)^(1 / (2 * order)));

    if ftype == 1
        % 低通
        Wc = Wp * W0;
    elseif ftype == 2
        % 高通
        Wc = Wp / W0;
    elseif ftype == 3
        % 带通
        W0 = [-W0, W0];
        W2 = Wp(2);
        W1 = Wp(1);
        Wc = W0 * (W2 - W1) / 2 + sqrt((W0.^2) / 4 * (W2 - W1)^2 + W1 * W2);
        Wc = sort(abs(Wc));
    end
    % 根据模拟频率计算数字频率，反畸变
    wc = (2 / pi) .* atan(T .* Wc ./ 2);
elseif ftype == 4
    % 带阻，直接调用库函数
    [order, wc] = buttord(wp, ws, Rp, Rs);
end
end
end

```

2.2. 计算数字滤波器的传递函数（butter）

```

function [bz, az] = my_butter(N, wc, ftype)
%%
% 频率预畸变
T = 2;
Wc = (2 / T) .* tan(pi .* wc ./ 2);
%%
% 计算截止频率为 1 rad/s 的归一化模拟巴特沃斯滤波器的系统函数
% [zn, pn, kn] 零极点增益形式
% [bn, an] 传递函数形式

```

```

% H(s)不存在零点（在无穷大处）
zn = inf .* ones(N, 1);
% 计算归一化巴特沃斯滤波器的极点
% P 为 H(s)*H(-s)的所有极点
if mod(N, 2) == 0
    P = exp(1j .* pi .* ((1:2 * N) - 0.5) ./ N);
elseif mod(N, 2) == 1
    P = exp(1j .* pi .* (1:2 * N) ./ N);
end
% H(s)的极点都需要在左半平面
pn = P(real(P) < 0);
kn = 1;

bn = kn .* poly(zn);
an = poly(pn);
%%
% 去归一化，将归一化低通滤波器映射到低通/高通/带通滤波器
% [zs, ps, ks] 零极点增益形式
% [bs, as] 传递函数形式
if ftype == 'lp'
    % 模拟低通到模拟低通
    zs = zn;
    ps = Wc .* pn;
    ks = kn * (Wc^N);
elseif ftype == 'hp'
    % 模拟低通到模拟高通
    zs = 1 ./ zn;
    ps = Wc ./ pn;
    ks = kn * prod(-pn);
elseif ftype == 'bp'
    % 模拟低通到模拟带通
    W0 = sqrt(Wc(1) * Wc(2)); % 中心频率
    Bw = abs(Wc(1) - Wc(2)); % 带宽
    zs = 1 ./ zn;
    for i = 1:N
        ps(2 * i - 1:2 * i) = roots([1, -Bw * pn(i), W0^2]);
    end
    ks = kn * (Bw^N);
elseif ftype == 'bs'
    % 模拟低通到模拟带阻
    W0 = sqrt(Wc(1) * Wc(2)); % 中心频率
    Bw = abs(Wc(1) - Wc(2)); % 带宽
    for i = 1:N
        zs(2 * i - 1:2 * i) = roots([1, 0, W0^2]);
    end

```

```

        ps(2 * i - 1:2 * i) = roots([1, -Bw * pn(i), W0^2]);
    end
    ks = kn * prod(-1 ./ pn);
end

bs = ks .* poly(zs);
as = poly(ps);
%%
% 双线性变换, 将模拟系统映射到数字系统
% [zz, pz, kz] 零极点增益形式
% [bz, az]      传递函数形式
zs = zs(isfinite(zs));
Z = (1 + zs * (T / 2)) ./ (1 - zs * (T / 2));
pz = (1 + ps * (T / 2)) ./ (1 - ps * (T / 2));
kz = ks .* real((prod((2 / T) - zs)) ./ prod((2 / T) - ps));
zz = -ones(size(pz));
for i = 1:numel(Z)
    zz(i) = Z(i);
end

az = real(poly(pz));
bz = real(kz .* poly(zz));
end

```

2.3. 滤波器函数 (filter)

```

function y = my_filter(b, a, x)
% 直接 II 型 IIR 滤波器的转置结构
% x    输入信号
% y    输出信号
% u    缓存
order = max(numel(a), numel(b)) - 1;

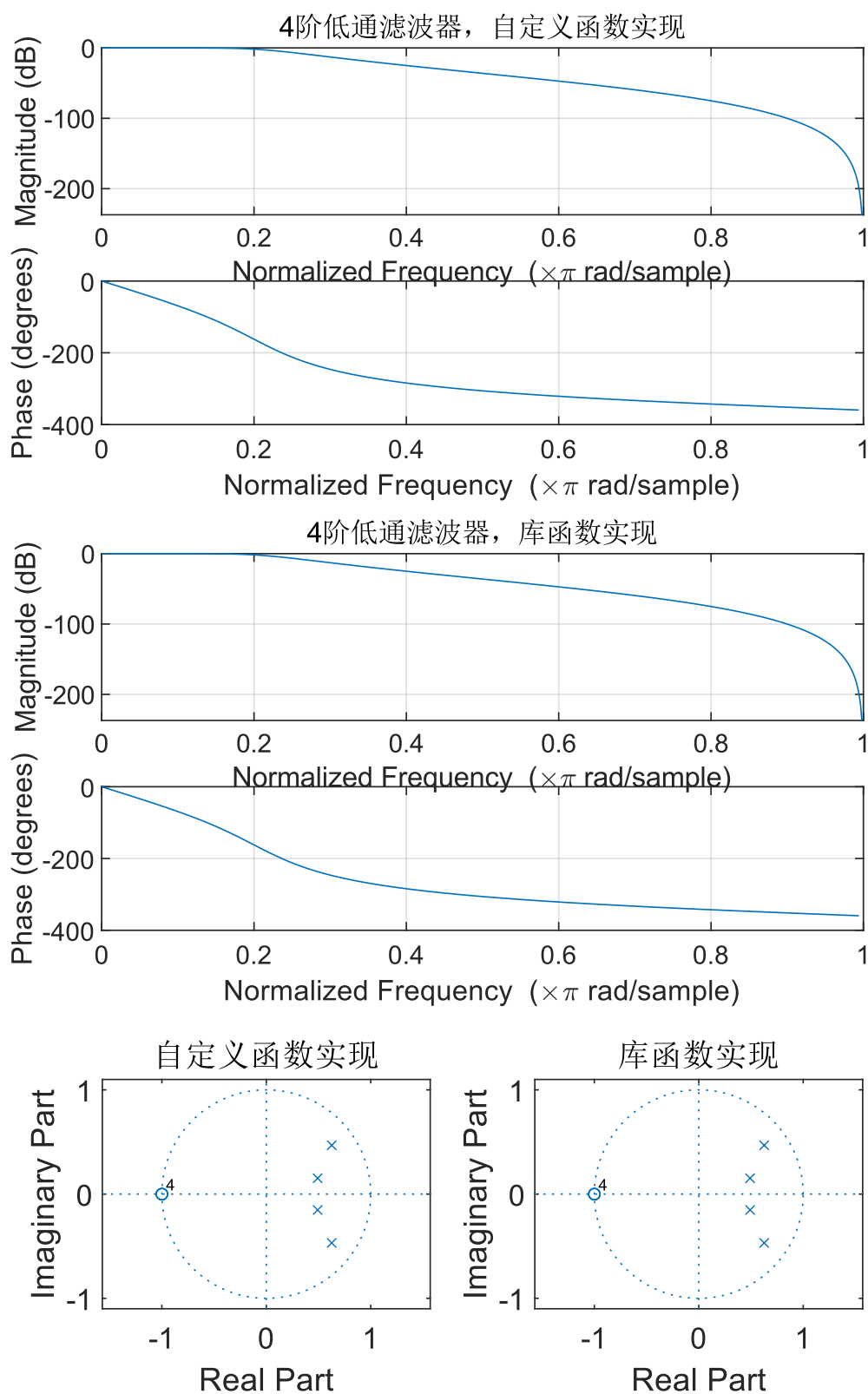
a = a ./ a(1);
b = b ./ a(1);
a(1) = [];
y = zeros(size(x));
u = zeros(1, order);

for i = 1:length(x)
    y(i) = b(1) * x(i) + u(1);
    u = [u(2:order), 0] + b(2:end) * x(i) - a * y(i);
end
end

```

3. 实验结果

3.1. 低通滤波器

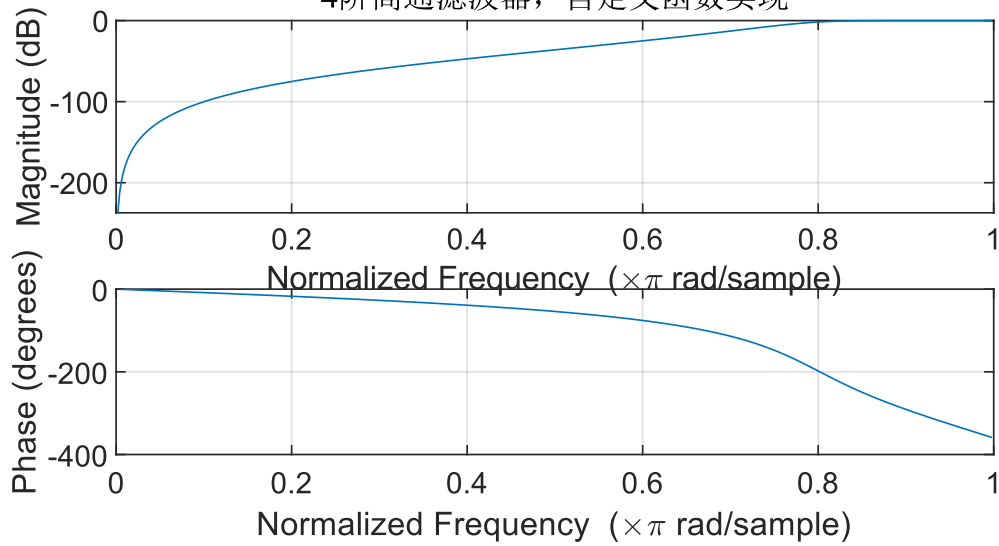


低通滤波器的频率响应曲线和零极点分布

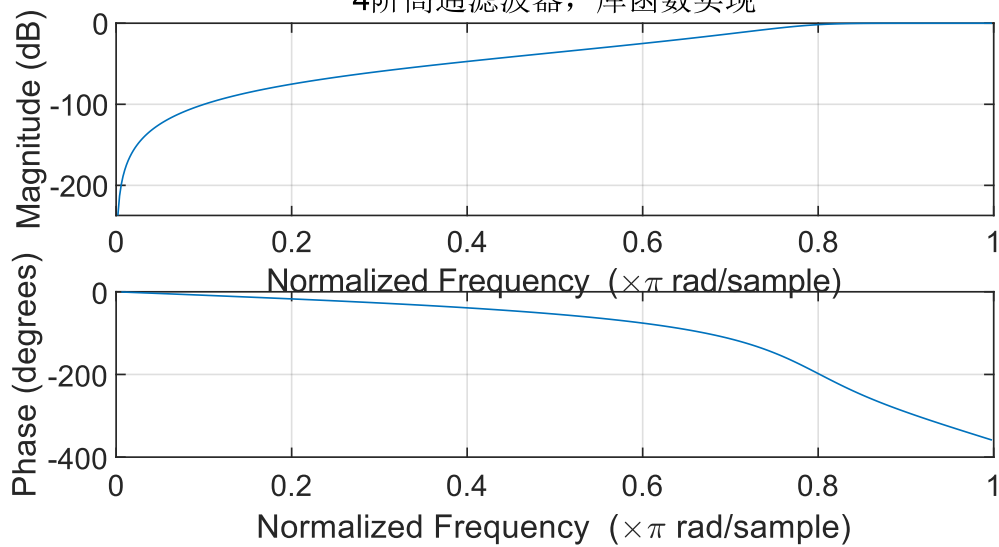
通带频率 0.2π ，衰减小于 2dB；阻带频率 0.4π ，衰减大于 25dB

3.2. 高通滤波器

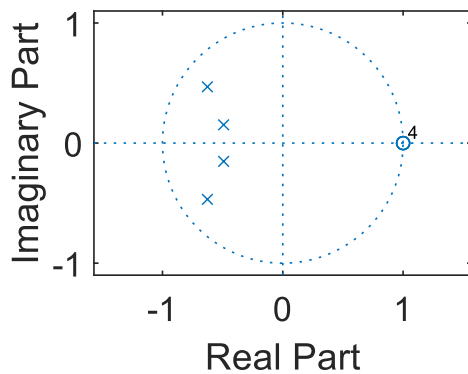
4阶高通滤波器，自定义函数实现



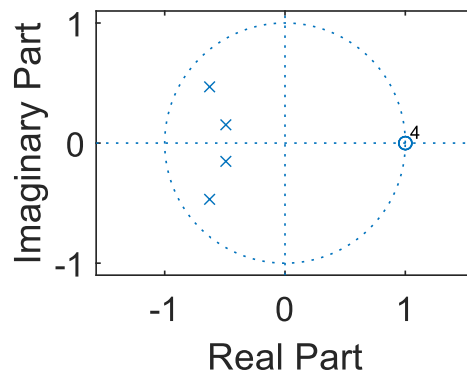
4阶高通滤波器，库函数实现



自定义函数实现



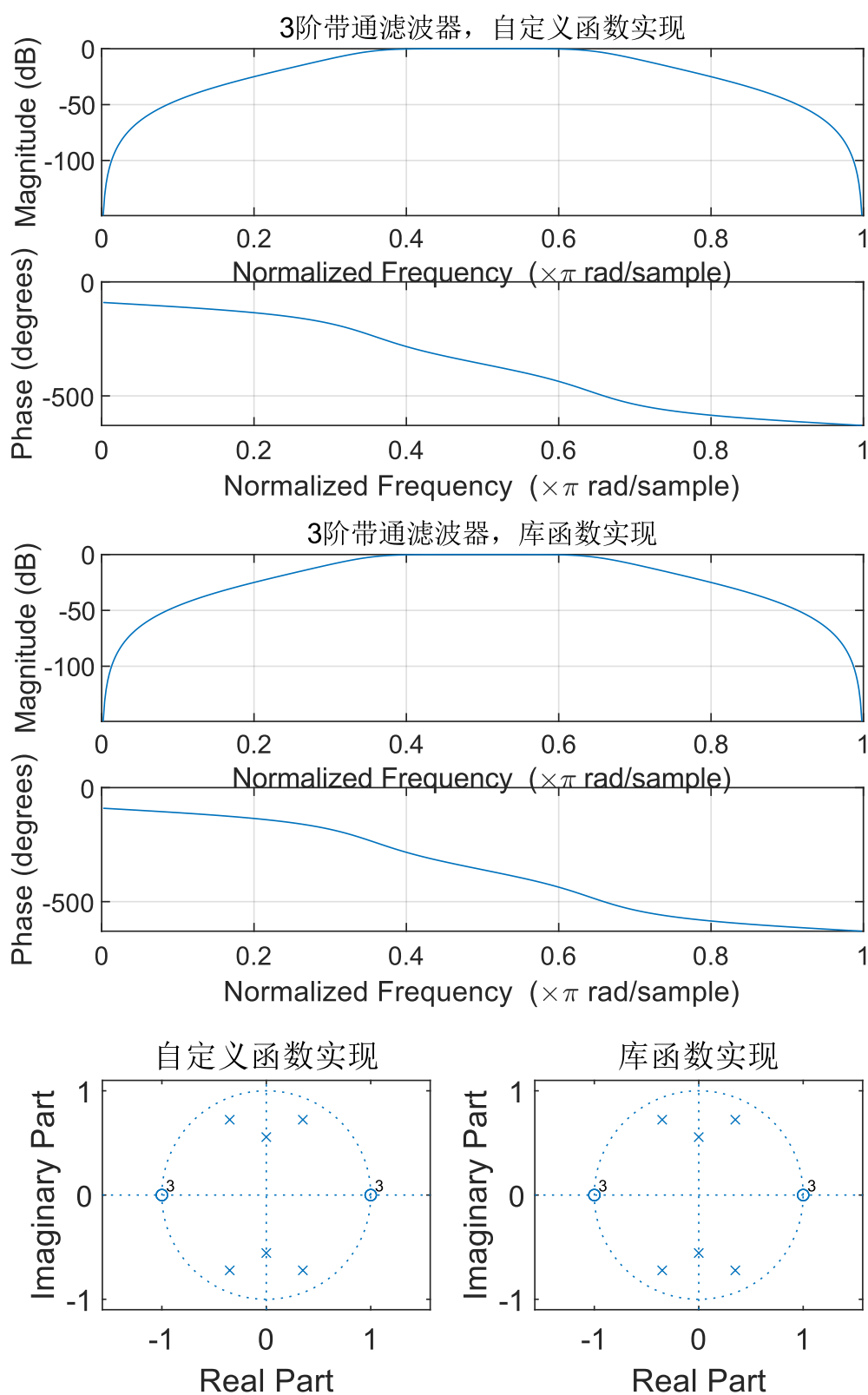
库函数实现



高通滤波器的频率响应曲线和零极点分布

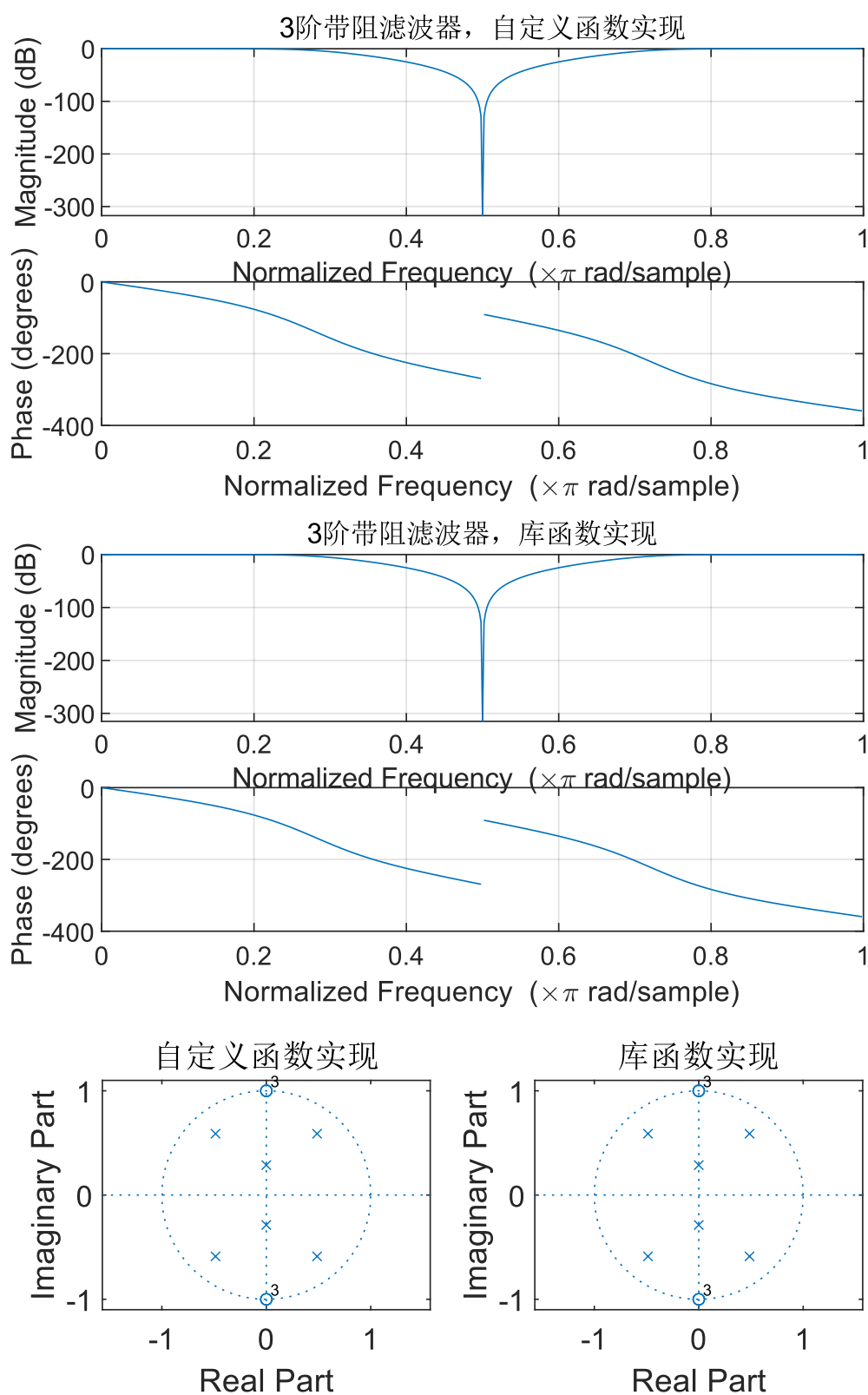
通带频率 0.8π ，衰减小于 2dB；阻带频率 0.6π ，衰减大于 25dB

3.3. 带通滤波器



带通滤波器的频率响应曲线和零极点分布
通带频率 0.4π 和 0.6π ，衰减小于 2dB；阻带频率 0.2π 和 0.8π ，衰减大于 25dB

3.4. 带阻滤波器

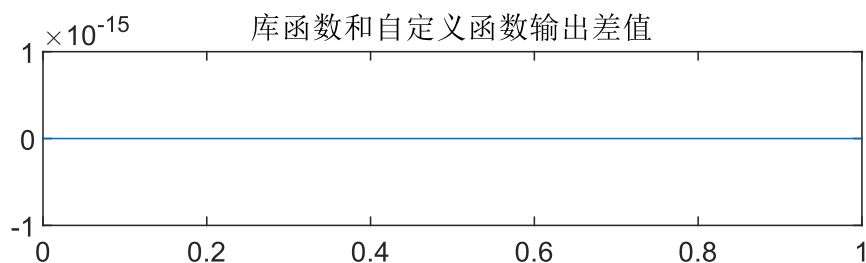


带阻滤波器的频率响应曲线和零极点分布
通带频率 0.2π 和 0.8π ，衰减小于 2dB；阻带频率 0.4π 和 0.6π ，衰减大于 25dB

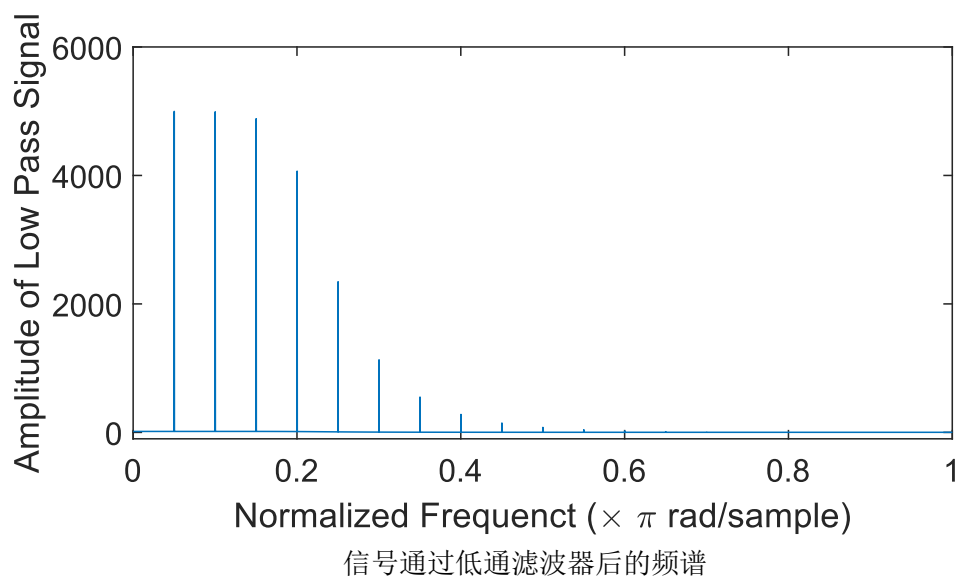
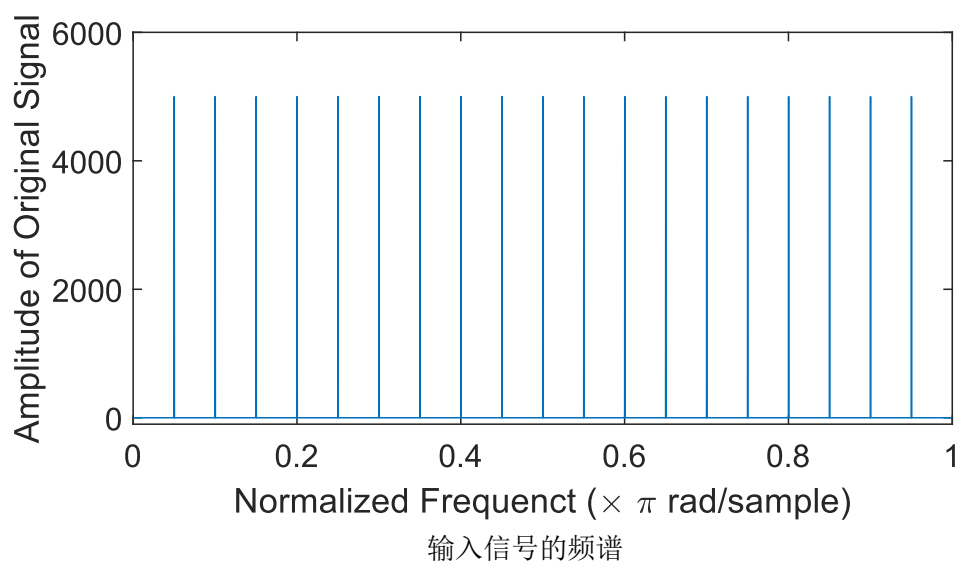
3.5. 根据传递函数计算输出信号

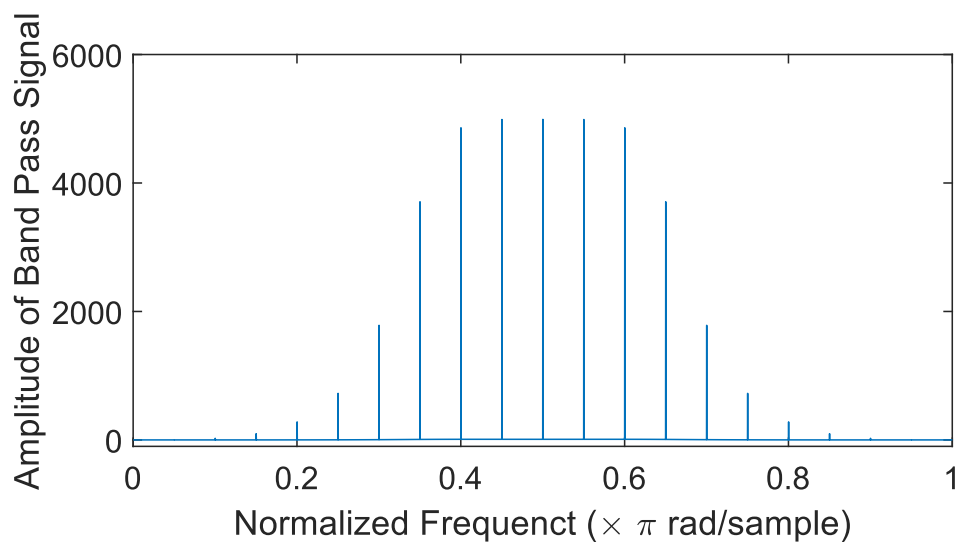
输入信号采样率为 10kHz，时长 1s。信号为频率从 250Hz 开始，以 250Hz 为步进增加到 4750Hz 的 19 个等振幅的正弦信号叠加而成。计算该信号分别通过前文中设计的低通、带通和高通滤波器后的频谱。

使用自定义函数和库函数分别计算高通滤波器的输出，并求差值，结果如下。

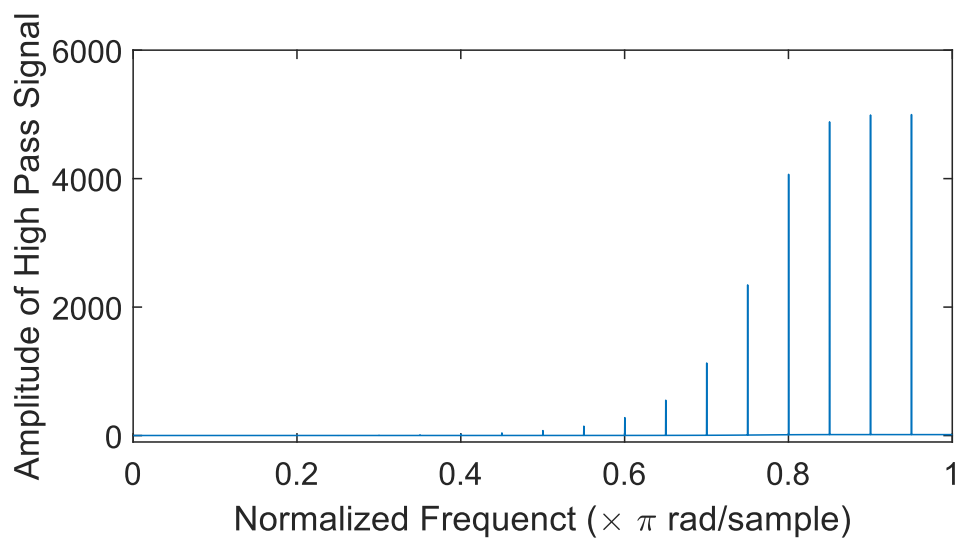


信号通过不同滤波器后的频谱数据为：

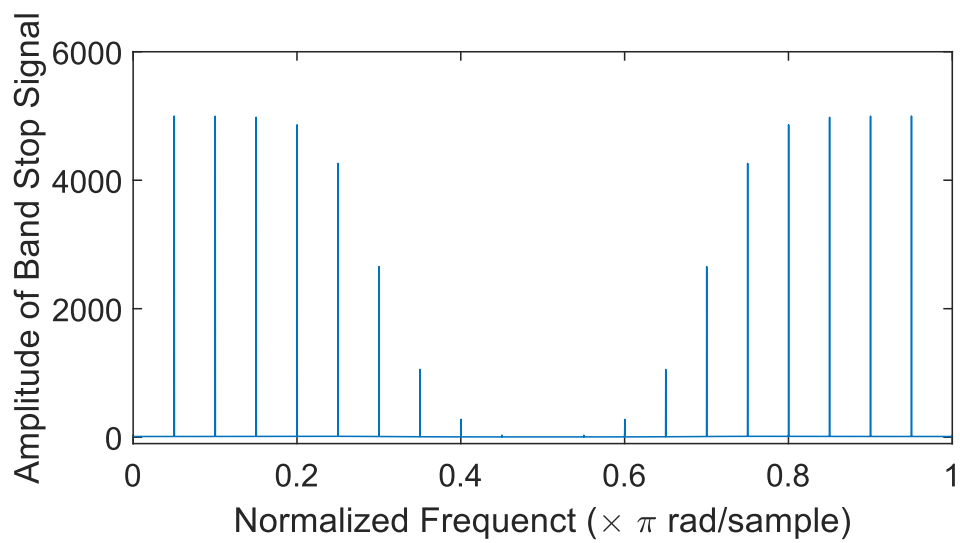




信号通过带通滤波器后的频谱



信号通过高通滤波器后的频谱



信号通过带阻滤波器后的频谱

4. 实验总结

本次实验中，我实现了数字巴特沃斯滤波器的系统函数计算，以及根据传递函数计算输出结果。

推导并完成计算滤波器的系统函数的计算过程时，我参考了 MATLAB 的库函数和数字信号处理相关书籍中的运算流程，根据其中的内容逐步推导计算公式并验算结果。计算公式的推导和实现部分也参考了 `scipy.signal.filter_design` 库中的相关内容的代码和参考资料。相较于 MATLAB 中的源代码，python 中的源代码更适合在学习时进行参考。

较前一次课程作业，本次实验中，我增加了实现带通滤波器和带阻滤波器的内容。并重新推导了模拟滤波器去归一化的计算公式，修改了前次实验中一些没有仔细推导验证的计算公式。

通过本次实验，我对数字滤波器的设计和实现方法有了一定的了解。从模拟滤波器出发，利用双线性变换最终得到数字滤波器的系统函数是一种较为常用的方法。同时，在数字信号处理的教材中还涉及到了其他的设计方法，特别是将模拟滤波器直接映射为不同类型的数字滤波器的方法。这一直接映射方法可能会更加简单直接。