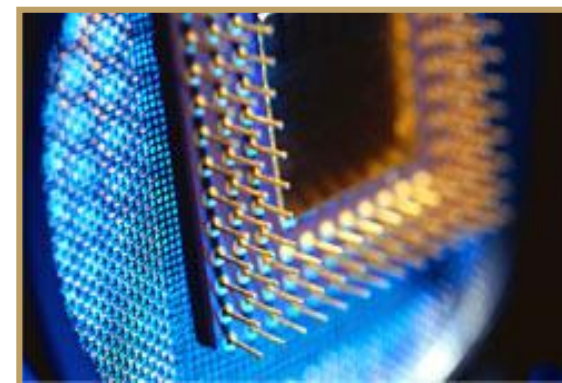
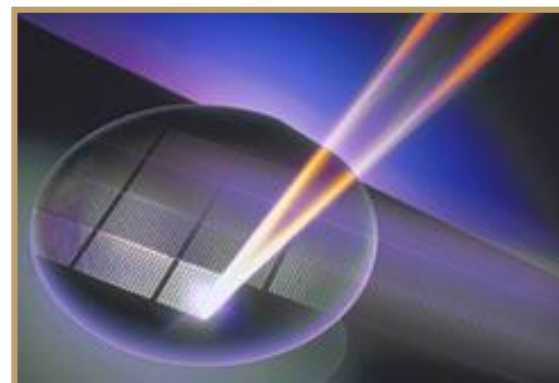
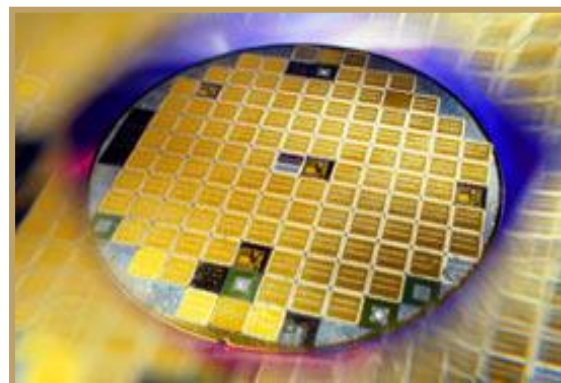




《VLSI数字通信原理与设计》课程

主讲人 贺光辉

实验1： IIR数字滤波器设计与实现





目录

01 实验目的&平台

02 IIR滤波器介绍

03 用MATLAB设计IIR滤波器

04 性能验证

05 报告要求



实验目的

- 学习IIR滤波器的原理和设计方法;
- 学会使用ModelSim、Vivado等工具;
- 深入理解流水线、并行处理、折叠结构等在实际电路中的作用。
- **本次实验1到3人一组。**

- MATLAB:
 - 利用FDA Designer工具箱完成滤波器的设计
 - 选取滤波器的类型，确定阶数、定点位宽等
- ModelSim
 - 完成相应RTL程序的仿真验证工作
- Vivado
 - 完成相应RTL程序的仿真验证工作
 - 综合设计，讨论时序、面积信息



目录

01 实验目的&平台

02 IIR滤波器介绍

03 用MATLAB设计IIR滤波器

04 性能验证

05 报告要求



应用场景

IIR滤波器常用于计算/存储资源受限、低延迟、非线性相位要求的场景

- 模拟滤波器的数字仿真
- 音频均衡
- IoT智能传感器
- 高速电信/射频应用
- 生物医学传感器信号处理
- 数据通信中的时钟恢复
- 抗混叠滤波器
- 信号监测

FIR滤波器常用于有大量的计算/存储资源可用且线性相位要求重要(高保真)的场景

- 音频/生物医学信号增强
- 精密仪器的信号采集/滤波
- 数字无线电中的ADC模块
- 视频/图像处理

IIR滤波器后可接全通网络进行相位均衡，优化群延时

IIR滤波器介绍

与FIR滤波器不同，IIR滤波器具有无限的脉冲响应，即 $\forall k, h(k) \neq 0$

IIR理论时域表达式： $y(n) = \sum_{k=0}^{\infty} h(k)x(n-k)$

上述长度为无穷的卷积无法直接实现，可引入一定形式的反馈（因果系统）

带反馈的IIR时域表达式： $y(n) = \sum_{i=0}^N b_i x(n-i) - \sum_{j=1}^M a_j y(n-j)$

传输函数： $H(z) = \frac{B(z)}{A(z)} = \frac{\sum_{i=0}^N b_i z^{-i}}{1 + \sum_{j=1}^M a_j z^{-j}}$ （或 $H(z) = \frac{\sum_{i=0}^N b_i z^{-i}}{\sum_{j=0}^M a_j z^{-j}}, a_0 = 1$ ）

Example：考虑一个传输函数为 $H(z) = \frac{1}{1-az^{-1}}$ 的IIR滤波器，其中 $0 < |a| < 1$

脉冲响应： $h(n) = a^n u(n)$ （ $u(n)$ 为单位阶跃函数）

时域表达式： $y(n) = \sum_{k=0}^{\infty} a^k x(n-k) = x(n) + ay(n-1)$

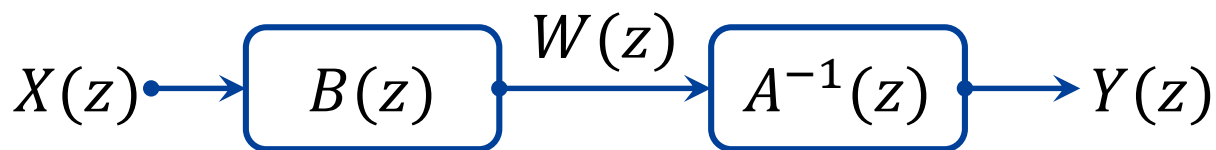
IIR滤波器的基本结构：直接 I 型

标准的N阶IIR滤波器($M=N$):

- $2N+1$ 个抽头系数($b_0, \dots, b_N, a_1, \dots, a_N$)
- $2N+1$ 个乘法器 & $2N$ 个两输入加法器

直接型结构：乘法器的系数等于传输函数的系数

以三阶IIR滤波器举例

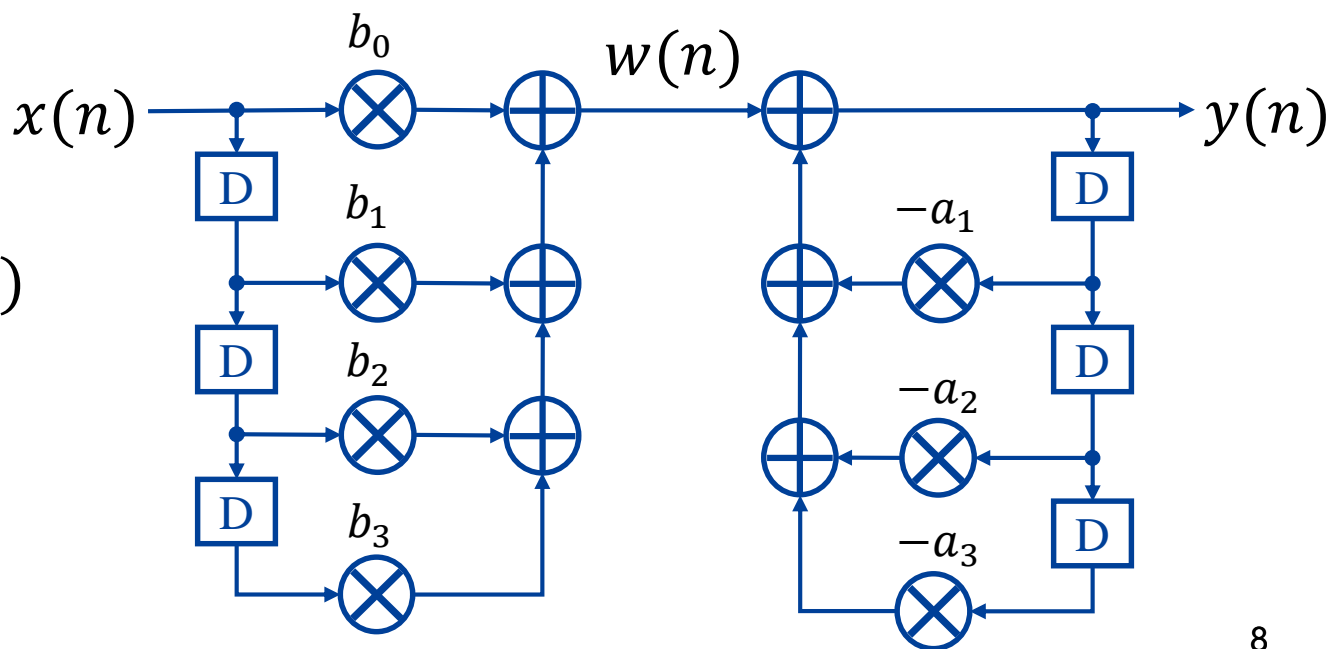


直接 I 型IIR滤波器实现方案

传输函数

$$H(z) = \frac{B(z)}{A(z)} = \frac{\sum_{i=0}^N b_i z^{-i}}{1 + \sum_{j=1}^N a_j z^{-j}}$$

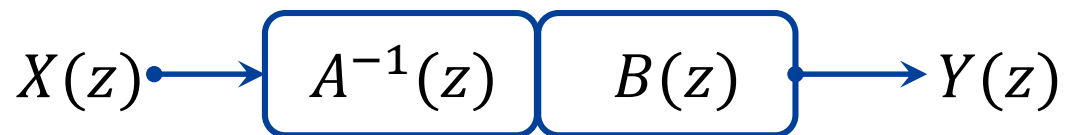
直接 I 型结构, $N=3$



IIR滤波器的基本结构：直接Ⅱ型

直接型结构：乘法器的系数等于传输函数的系数

直接Ⅱ型：在直接Ⅰ型的基础上，交换串联结构的顺序，并复用延迟单元(更常用)

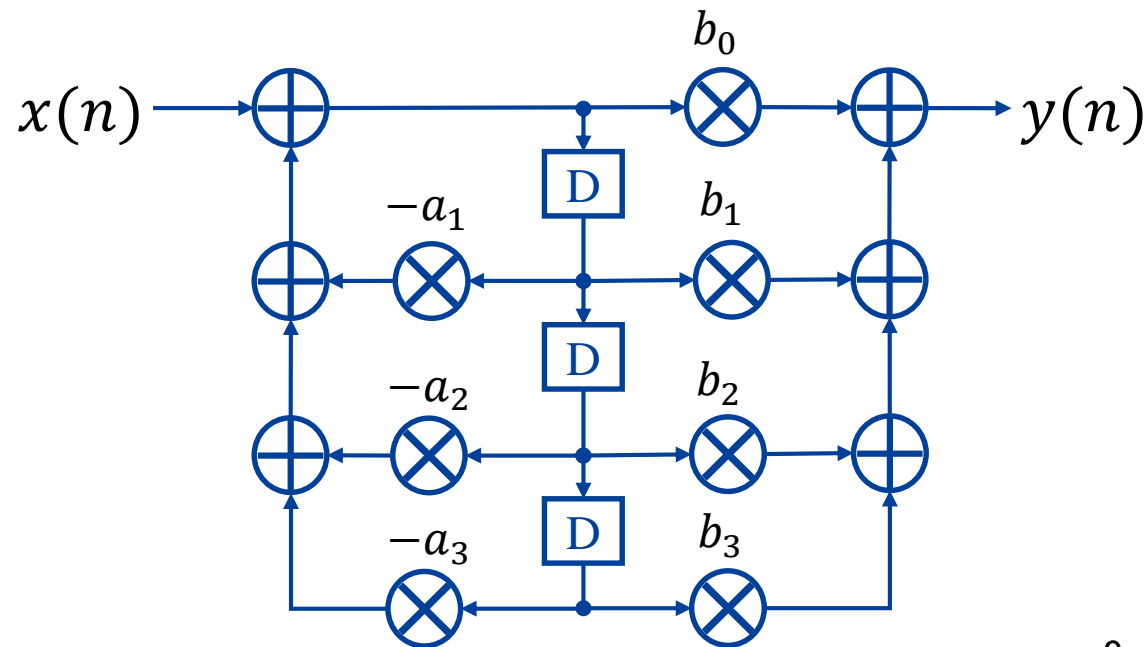


直接Ⅱ型IIR滤波器实现方案

传输函数

$$H(z) = \frac{B(z)}{A(z)} = \frac{\sum_{i=0}^N b_i z^{-i}}{1 + \sum_{j=1}^N a_j z^{-j}}$$

直接Ⅱ型结构, $N=3$



IIR滤波器的基本结构：级联型

级联型结构

将 $H(z)$ 的分子分母多项式表示为若干低次多项式的积，然后用低阶IIR滤波器级联实现

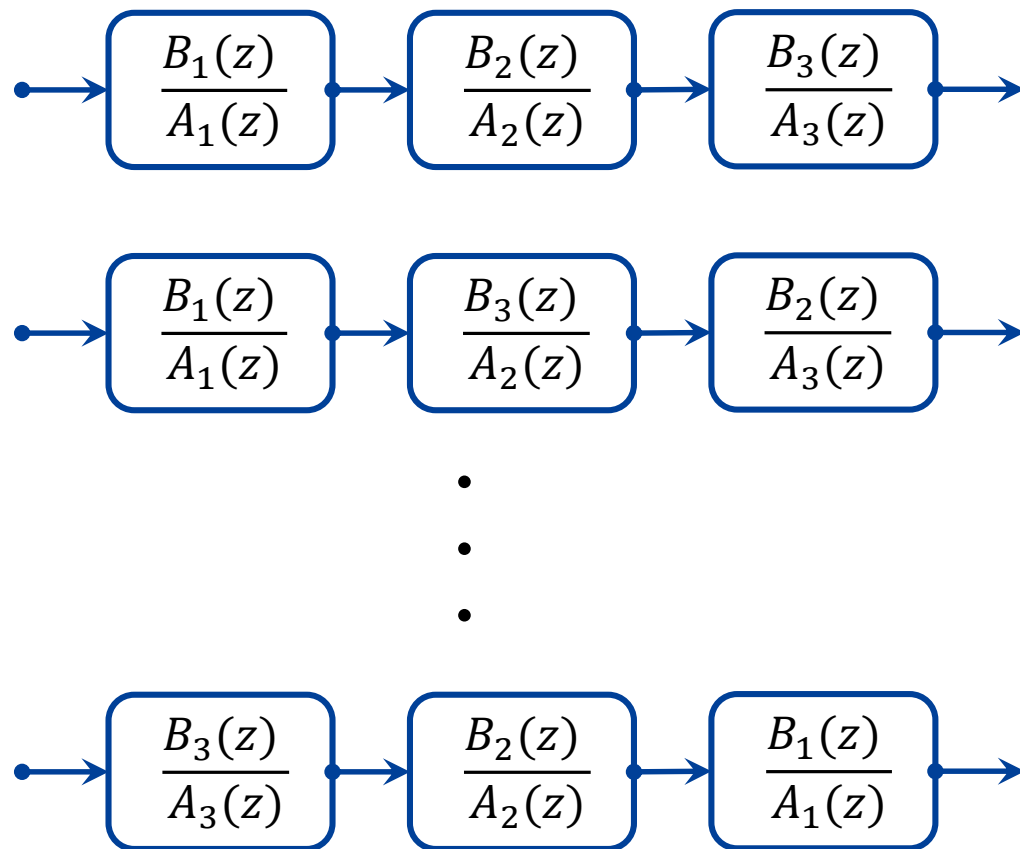
根据不同极零点的配对，有多种实现方案

例如，对于如下因式分解，有36种级联方案

$$H(z) = \frac{B(z)}{A(z)} = \frac{B_1(z)B_2(z)B_3(z)}{A_1(z)A_2(z)A_3(z)}$$

每个低阶子模块可采用直接II型设计

由于有限字长效应，不同级联方案的输出结果可能不同



不同的级联型IIR滤波器实现方案

IIR滤波器的基本结构：级联型实现举例

例 8.4 级联型实现举例

我们生成式(8.26)的三阶 IIR 传输函数的级联实现。对 $H(z)$ 的分子和分母进行式(8.28)形式的因式分解, 得到

$$\begin{aligned} H(z) &= \frac{0.44z^2 + 0.362z + 0.02}{(z^2 + 0.8z + 0.5)(z - 0.4)} \\ &= \left(\frac{0.44 + 0.362z^{-1} + 0.02z^{-2}}{1 + 0.8z^{-1} + 0.5z^{-2}} \right) \left(\frac{z^{-1}}{1 - 0.4z^{-1}} \right) \end{aligned}$$

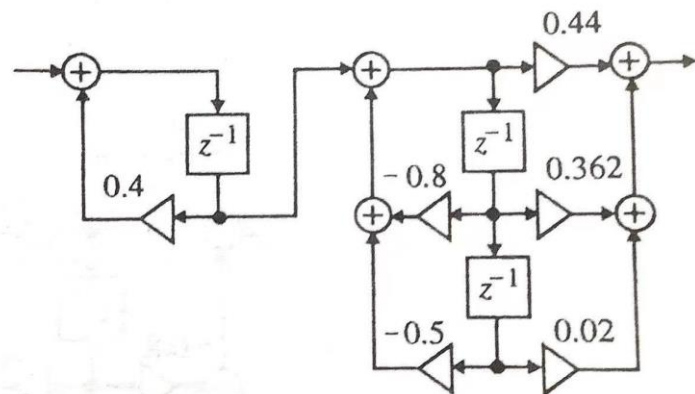


图 8.19 基于每个节的直接 II 型实现的式(8.26)给出的 IIR 传输函数的级联实现

IIR滤波器的基本结构：并联型

并联型结构

将 $H(z)$ 进行部分分式展开，以并联方式实现

并联Ⅰ型

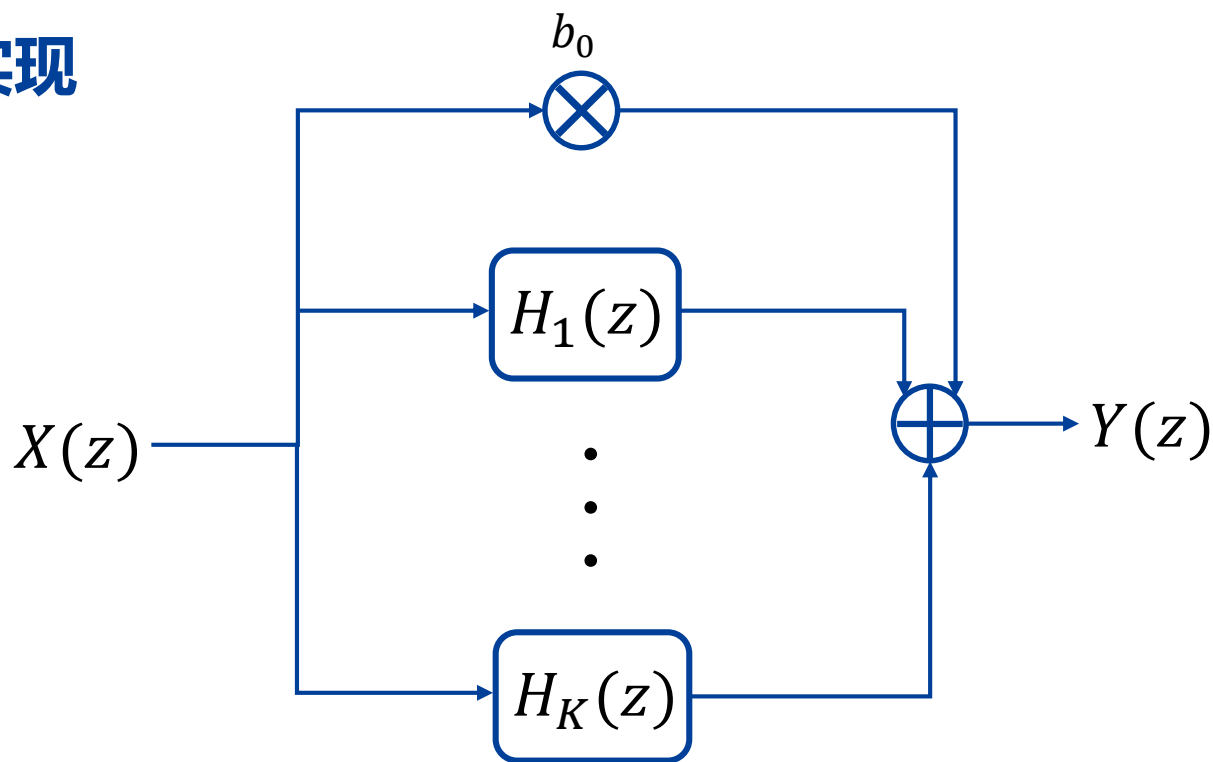
$$H(z) = b_0 + \sum_k \frac{b_{0k} + b_{1k}z^{-1}}{1 + a_{1k}z^{-1} + a_{2k}z^{-2}}$$

对于实极点，有 $b_{1k} = a_{2k} = 0$

并联Ⅱ型

$$H(z) = b_0 + \sum_k \frac{b_{1k}z^{-1} + b_{2k}z^{-2}}{1 + a_{1k}z^{-1} + a_{2k}z^{-2}}$$

对于实极点，有 $b_{2k} = a_{2k} = 0$



并联型IIR滤波器实现方案

IIR滤波器的基本结构：并联型实现举例

例 8.5 并联型实现举例

我们生成式(8.26)的三阶 IIR 传输函数的两种不同的并联实现。为此我们按照式(6.40)中表示为 z^{-1} 的多项式比值的形式对 $H(z)$ 进行部分分式展开, 形成

$$H(z) = -0.1 + \frac{0.6}{1 - 0.4z^{-1}} + \frac{-0.5 - 0.2z^{-1}}{1 + 0.8z^{-1} + 0.5z^{-2}}$$

得到并联 I 型实现, 如图 8.21(a) 所示。

最后, 将 $H(z)$ 表示为 z 的多项式的比值的直接部分分式展开为

$$H(z) = \frac{0.24}{z - 0.4} + \frac{0.2z + 0.25}{z^2 + 0.8z + 0.5} = \frac{0.24z^{-1}}{1 - 0.4z^{-1}} + \frac{0.2z^{-1} + 0.25z^{-2}}{1 + 0.8z^{-1} + 0.5z^{-2}}$$

得到并联 II 型实现方式, 如图 8.21(b) 所示。

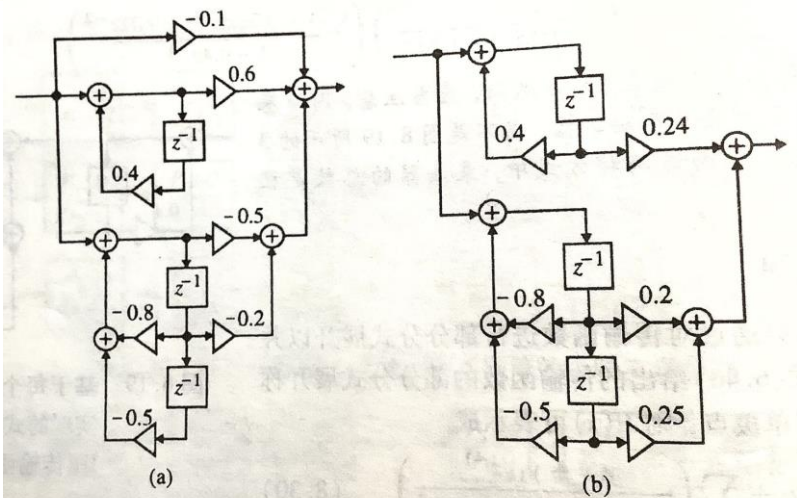


图 8.21 例 8.5:(a) 并联 I 型实现;(b) 并联 II 型实现

■ 相同传输函数时, 直接 I / II 型、级联型、并联型 IIR 滤波器的对比

- 关键路径/吞吐
- 计算/存储资源开销
- 敏感度(受有限字长效应影响程度)
- 可调性(方便调整极/零点)

■ 相同滤波器指标时, FIR vs IIR

- | | |
|-----------------|---------|
| • 计算/存储资源开销(阶数) | • 延迟/吞吐 |
| • 线性相位/群延时 | • 定点化性能 |
| • BIBO 稳定性 | • 其他 |

相同设计指标下FIR与IIR滤波器的对比

性质	FIR	IIR
定义	有限脉冲响应	无限脉冲响应
结构	非递归(无反馈)	递归(有反馈)
因果性	因果	非因果
传输函数	仅有零点	有极/零点
计算资源消耗	高	低
延迟	高	低
存储需求	高	低
通带/过渡带/阻带	更波动/更宽/旁瓣大	更平坦/更窄/旁瓣小
模拟等效性	无	有
最小相位设计	难	易
线性相位	满足	不满足
稳定性	高(始终稳定)	低(极点在单位圆内时稳定)
敏感度	低	高
设计/分析/量化难度	低	高



目 录

01 实验目的&平台

02 IIR滤波器介绍

03 用MATLAB设计IIR滤波器

04 性能验证

05 报告要求



IIR滤波器设计的MATLAB设计方法

filterDesigner工具箱采用模拟频率变换法设计IIR滤波器

例如，设计一个采样频率 $f_s=10000\text{Hz}$ ，通带为 $1000\text{-}2000\text{Hz}$ ，截止频率为 $500\text{、}2500\text{Hz}$ ，通带波纹至多 1 dB ，阻带衰减至少 60 dB 的IIR带通滤波器，采用巴特沃斯模拟滤波器变换设计，MATLAB代码如下所示

```
[n,Wn] = buttord([1000 2000]/5000,[500 2500]/5000,1,60);  
[b,a] = butter(n,Wn);
```

其中， a 、 b 为直接II型IIR滤波器的抽头系数

可视化设计

为了方便定点化设计，在命令行窗口输入filterDesigner  `>> filterDesigner`

直接II型，二阶节

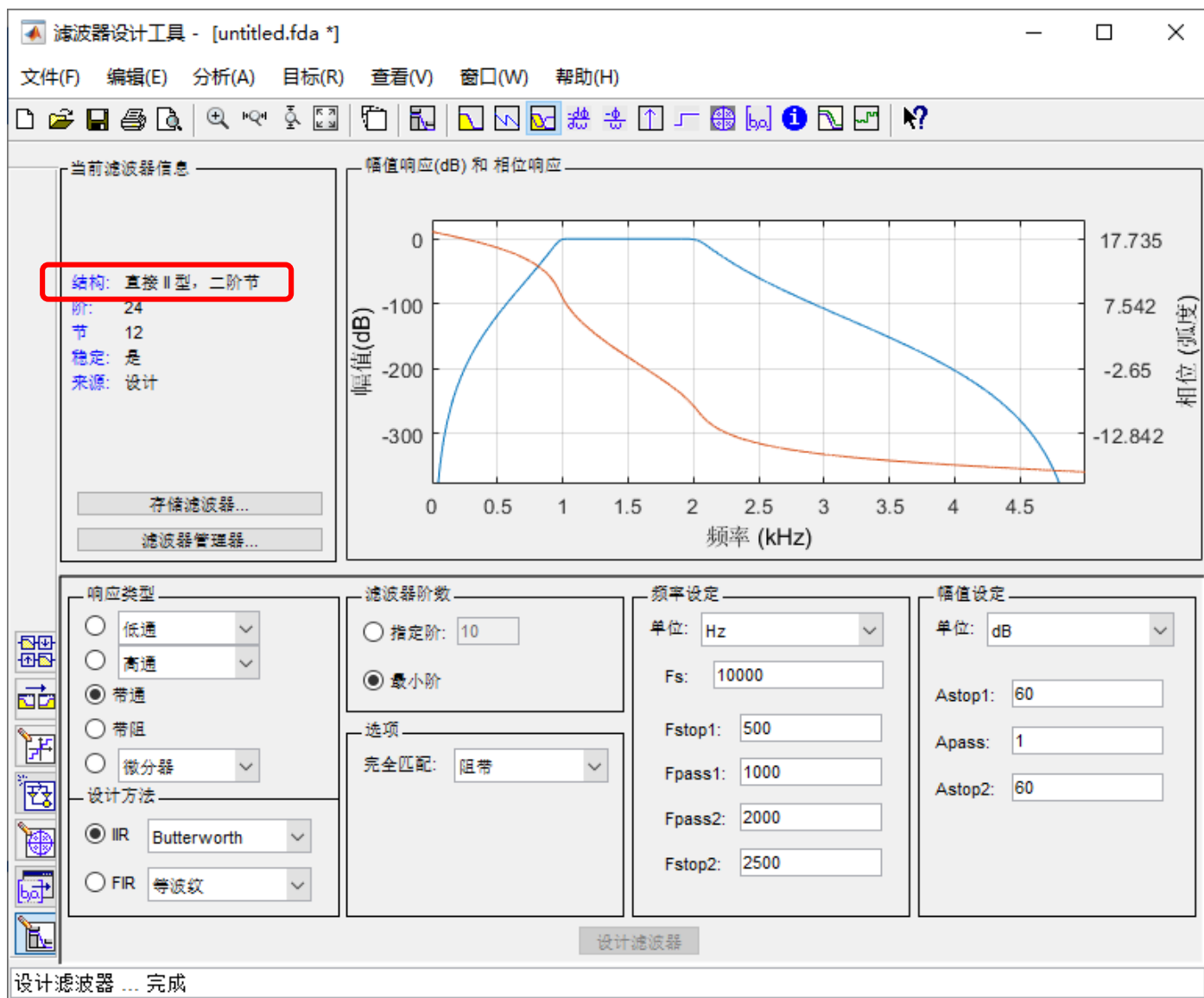


IIR滤波器设计的MATLAB设计方法

该工具箱支持多种类型的IIR滤波器设计，可在左上角的编辑中选择单节/二阶节以及各种结构

默认结构为直接II型，二阶节，即由12节二阶IIR滤波器级联而成

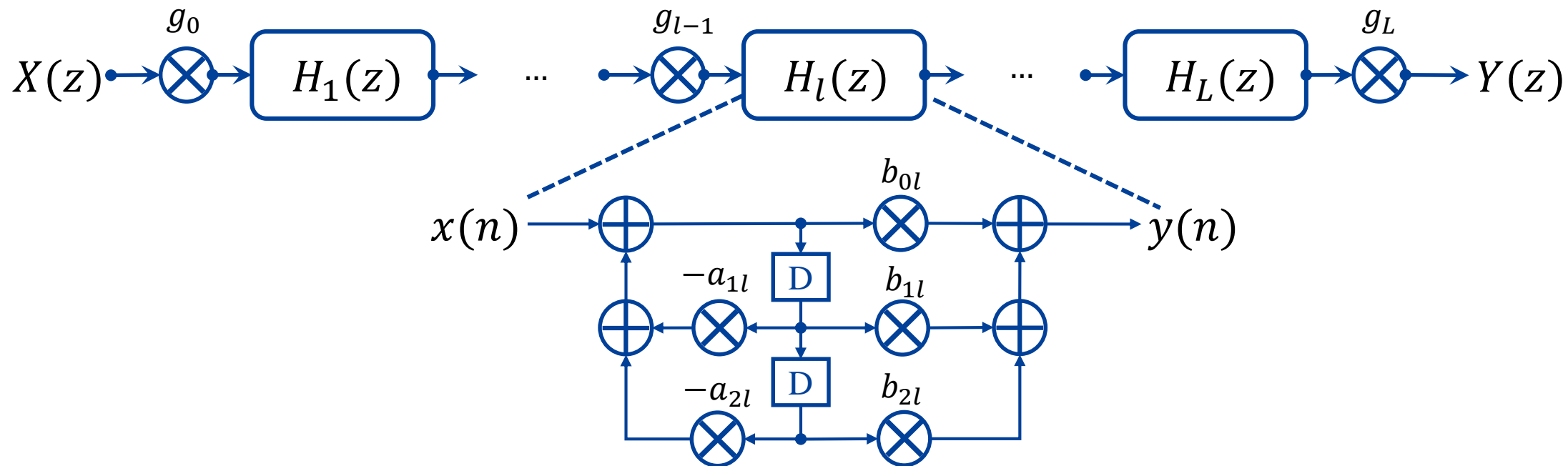
该结构胜在稳定，对有限字长效应最不敏感，且方便调节极零点



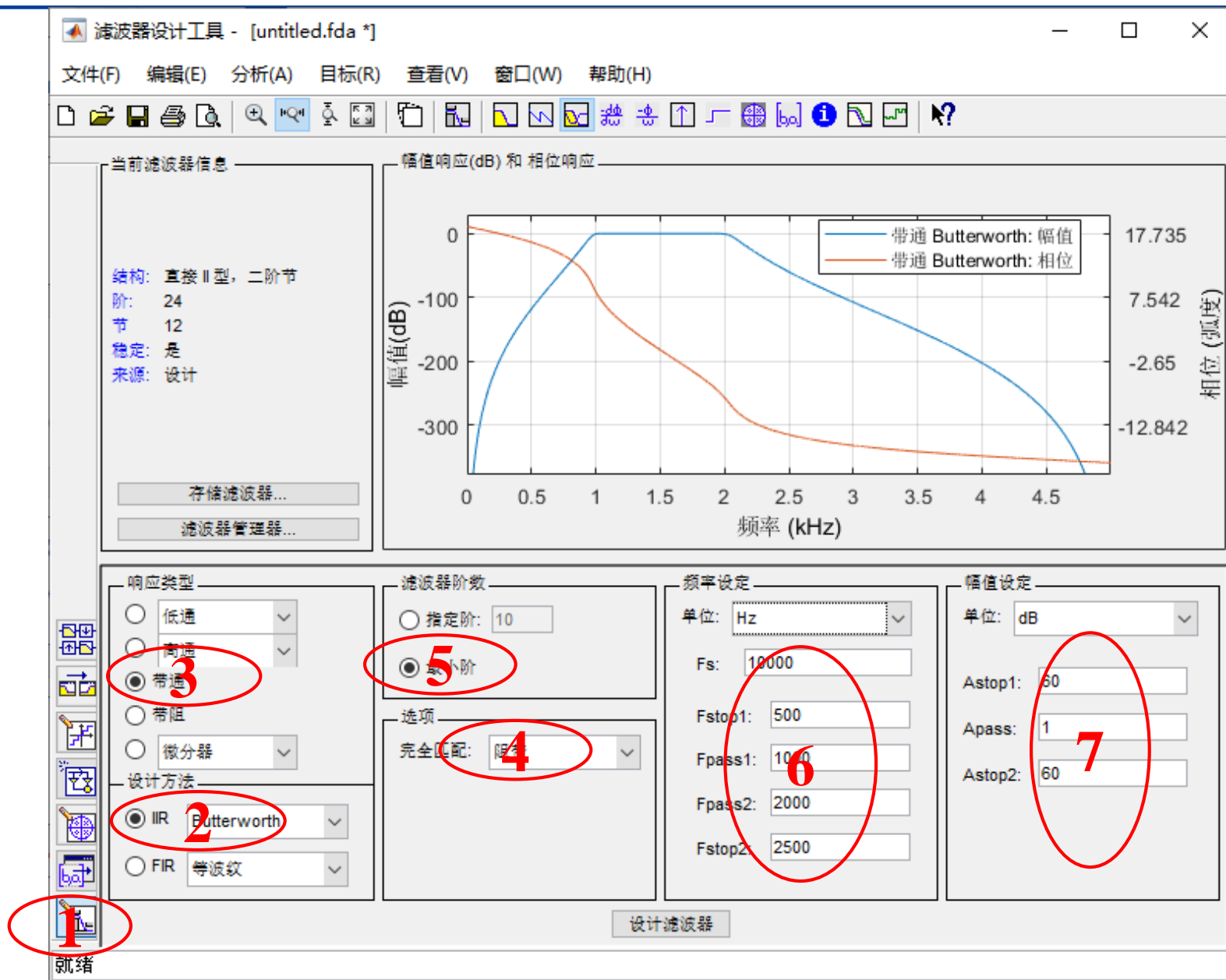
IIR滤波器设计的MATLAB设计方法——直接II型，二阶节

直接II型，二阶节结构的抽头系数以SOS(second-order sections)矩阵和定标向量G(gain)的形式输出。对于节数为L，其输出与对应滤波器结构如下所示

$$SOS = \begin{bmatrix} b_{01} & b_{11} & b_{21} & 1 & a_{11} & a_{12} \\ b_{02} & b_{12} & b_{22} & 1 & a_{12} & a_{22} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ b_{0L} & b_{1L} & b_{2L} & 1 & a_{1L} & a_{2L} \end{bmatrix}, \quad G = \begin{bmatrix} g_0 \\ \vdots \\ g_L \end{bmatrix}$$



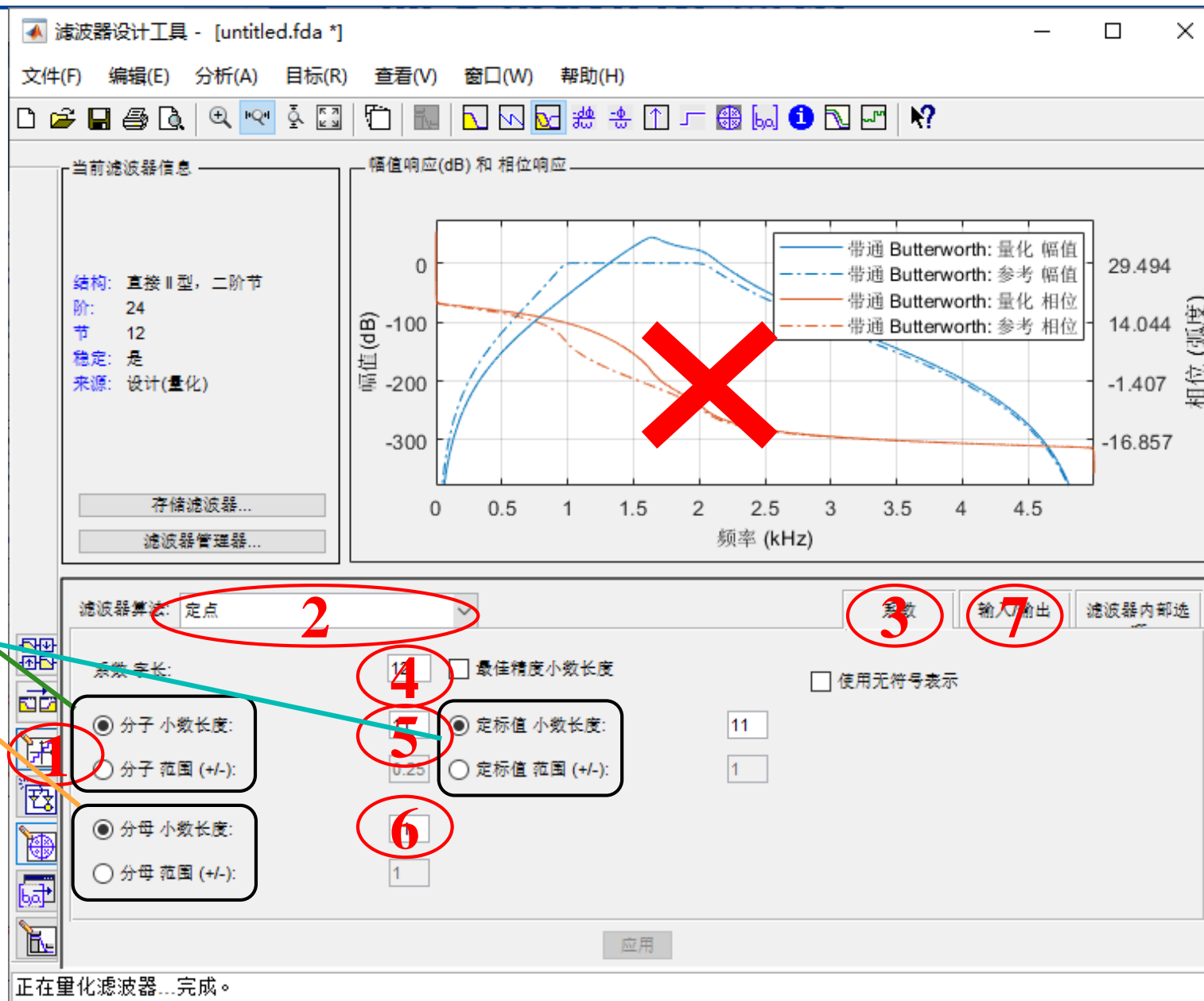
设置目标参数——直接II型，二阶节



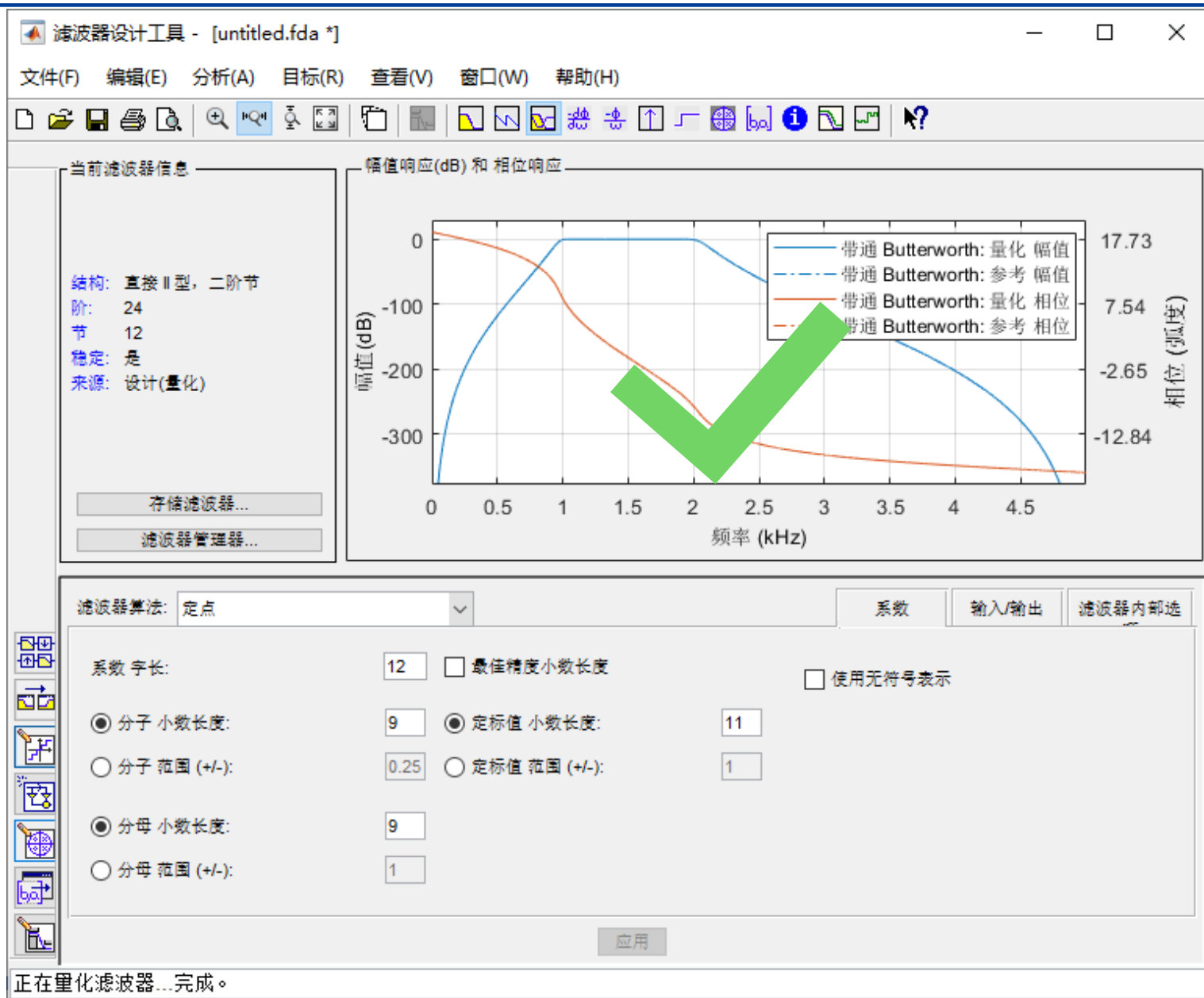
抽头系数的定点化——直接II型，二阶节

$$g_{l-1}H_l(z) = \frac{\sum_{i=0}^3 b_{il}z^{-i}}{\sum_{j=0}^3 a_{jl}z^{-j}}, a_{0l} = 1$$

$l = 1, \dots, L, g_L = 1$



抽头系数的定点化——直接II型，二阶节



导出抽头系数——直接II型，二阶节

文件->导出：导出量化后的SOS矩阵和定标向量G到工作区



SOS矩阵转化为非级联的抽头系数(直接II型)

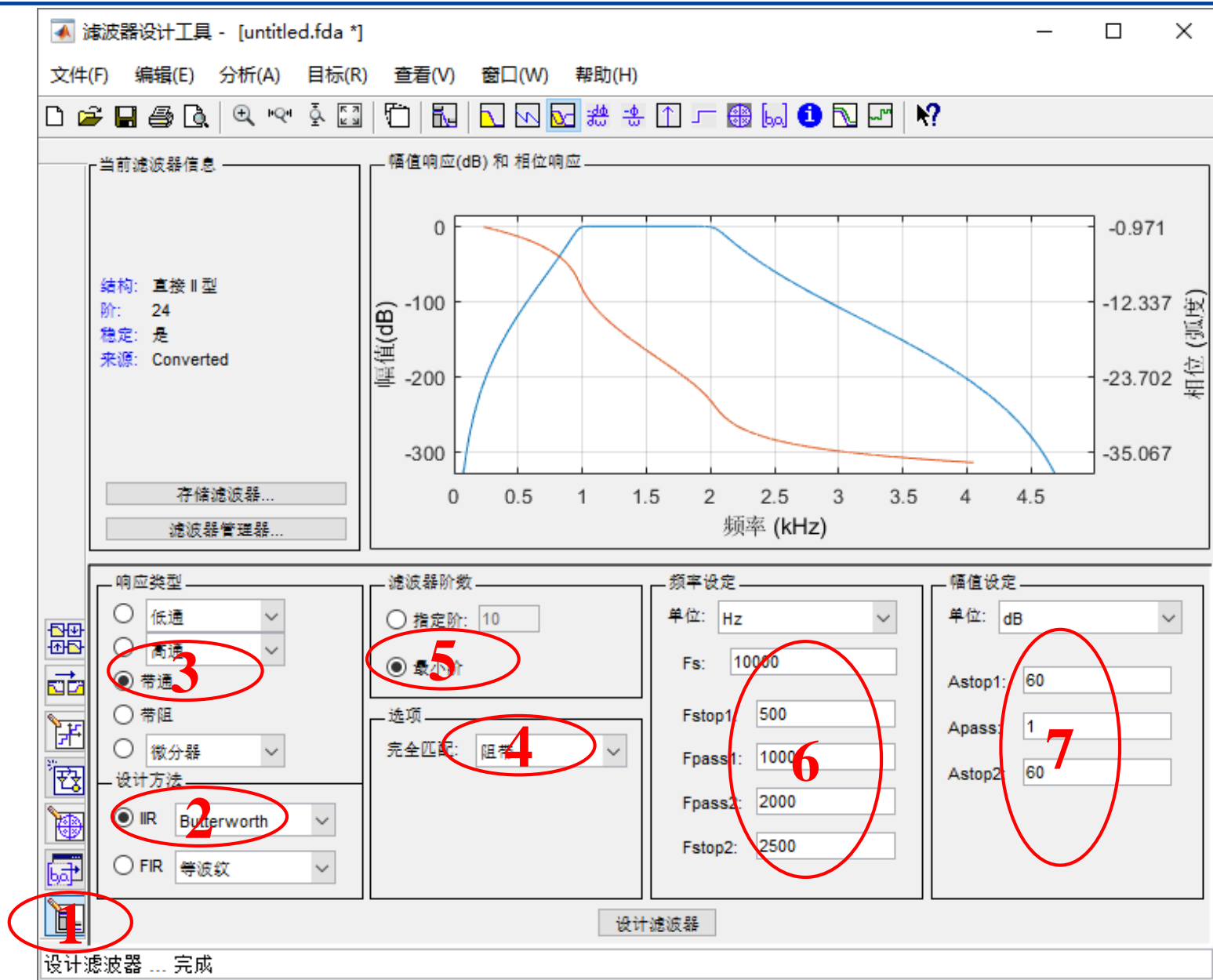
$[b \ a] = \text{sos2tf}(\text{SOS}, G);$

$$g_L \prod_{l=1}^L g_{l-1} H_l(z) \longrightarrow \frac{\sum_{i=0}^N b_i z^{-i}}{1 + \sum_{j=1}^N a_j z^{-j}}$$

直接II型，单节

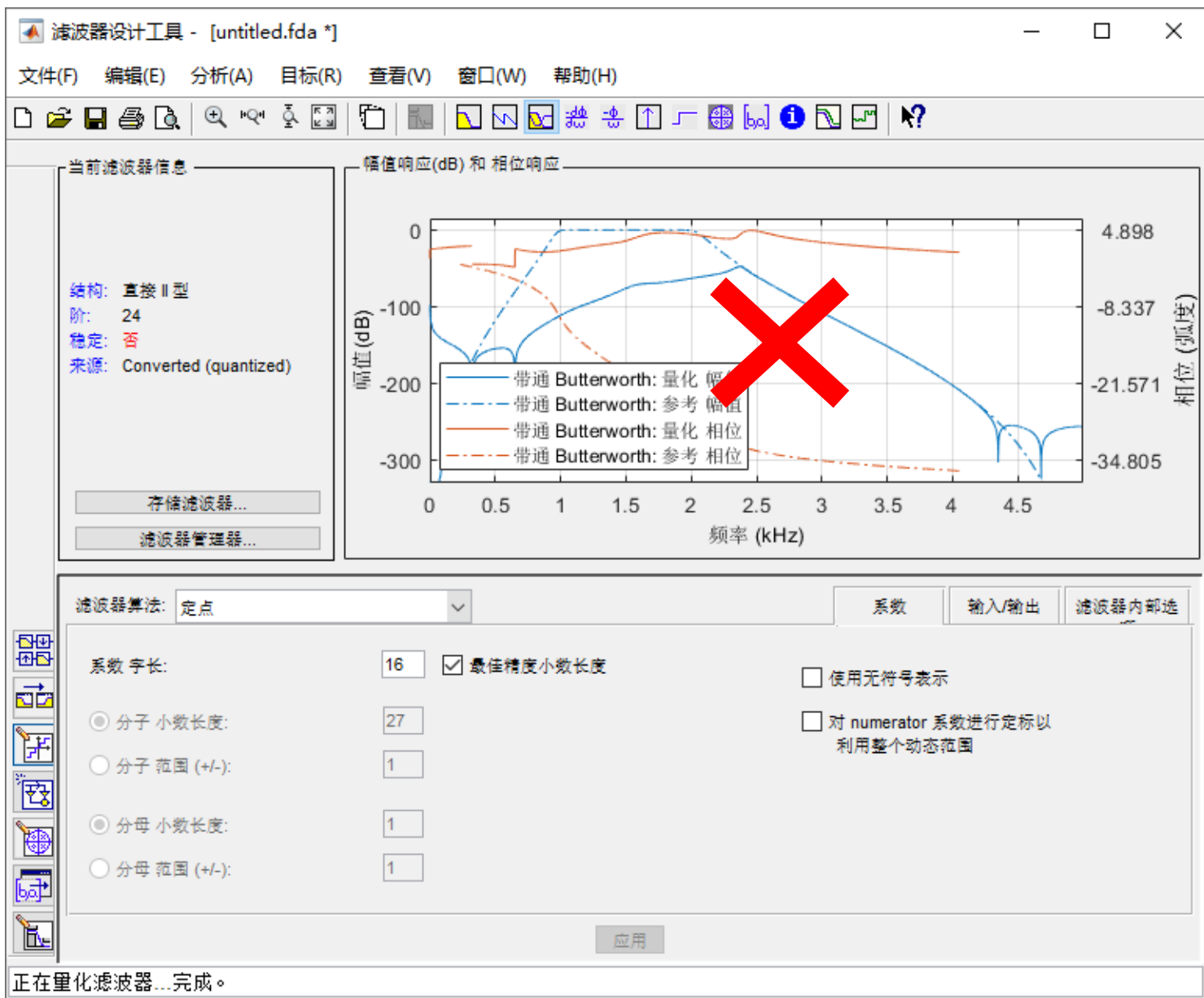


设置目标参数——直接II型，单节



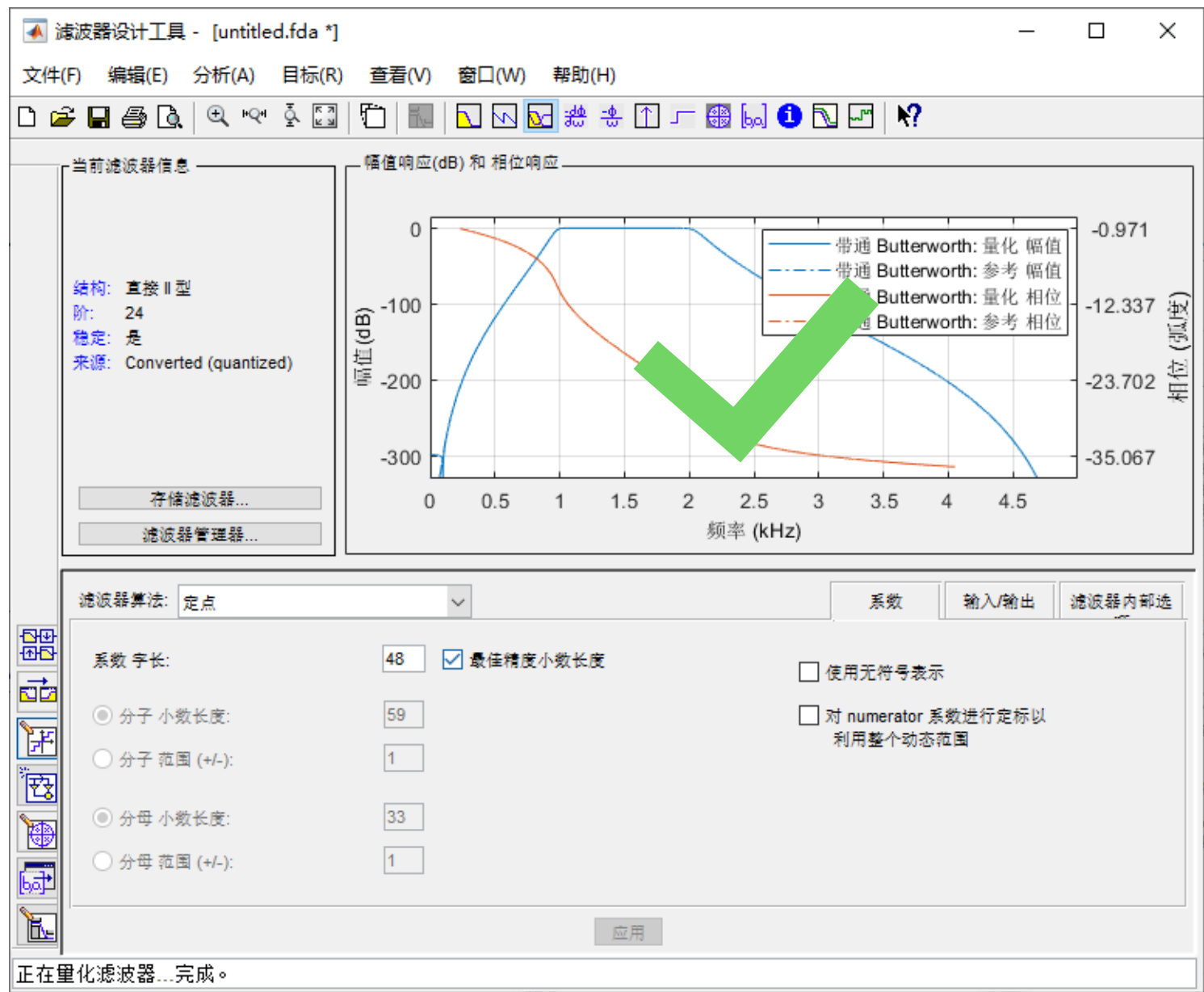
抽头系数的定点化——直接II型，单节

阶数较高时，IIR滤波器受有限字长效应影响严重，定点化性能较差，且易不稳定！！



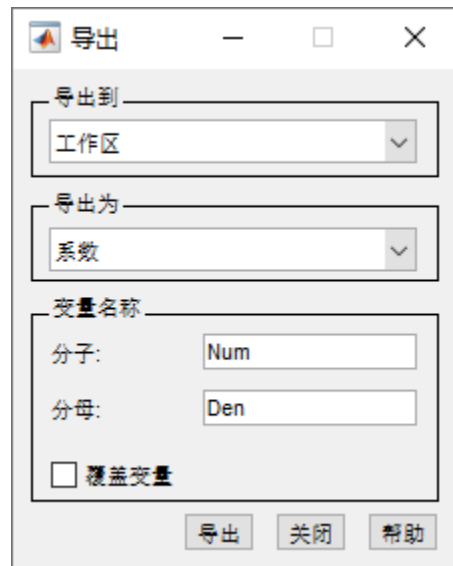
抽头系数的定点化——直接II型，单节

定点字长足够时，方能满足要求



导出抽头系数——直接II型，单节

文件->导出：导出量化后的抽头系数[b, a]到工作区



抽头系数转化为二阶节级联的SOS矩阵

$$[\text{SOS}, G] = \text{tf2sos}(b, a); \quad \frac{\sum_{i=0}^N b_i z^{-i}}{1 + \sum_{j=1}^N a_j z^{-j}} \quad \longrightarrow \quad g_L \prod_{l=1}^L g_{l-1} H_l(z)$$



目录

01 实验目的&平台

02 IIR滤波器介绍

03 用MATLAB设计IIR滤波器

04 性能验证

05 报告要求



验证滤波效果(以直接型为例)

生成一组包含100/1500/3000Hz的余弦信号

```
% Generating signals
Fs = 1e4; % Fs=10kHz
f1 = 100; % f1=100Hz
f2 = 1500; % f2=1500Hz
f3 = 3000; % f3=3000Hz
t = 0 : 1/Fs : 1;
s1 = cos((2 * pi * f1) .* t); % Generating cosine signal with f1=100 Hz
s2 = cos((2 * pi * f2) .* t); % Generating cosine signal with f2=1500 Hz
s3 = cos((2 * pi * f3) .* t); % Generating cosine signal with f3=3000 Hz
s = (s1 + s2 + s3) ./ 3; % add them
```

绘制其时域信号，并对其做FFT，绘制其幅度频谱

```
figure(1); % plot
subplot(2,1,1); stem(s(1:300)); grid; % plot original
title('Original signal in time-domain');
xlabel('k');
ylabel('s(k)');
% FFT on signal s
len = 4096; % FFT length
f = Fs .* (0 : len/2 - 1) ./ len;
y = fft(s, len); % do len-points FFT transform on the signal s
```

```
subplot(2,1,2);
plot(f, abs(y(1:len/2))); grid; % plot original signal s on the frequency domain
title('Original signal spectrum in frequency domain');
xlabel('f (Hz)');
ylabel('S(f)');
```

验证滤波效果(以直接型为例)

载入抽头系数，并进行滤波得到滤波后时域信号，做FFT

% load the IIR tap coefficient and filter the signal

load('.\a.mat');

load('.\b.mat');

s_bp = filter(b, a, s); % do IIR filtering to get bandpass signal s_bp
y_bp=fft(s_bp, len); % do len-points FFT transform on the signal s_bp



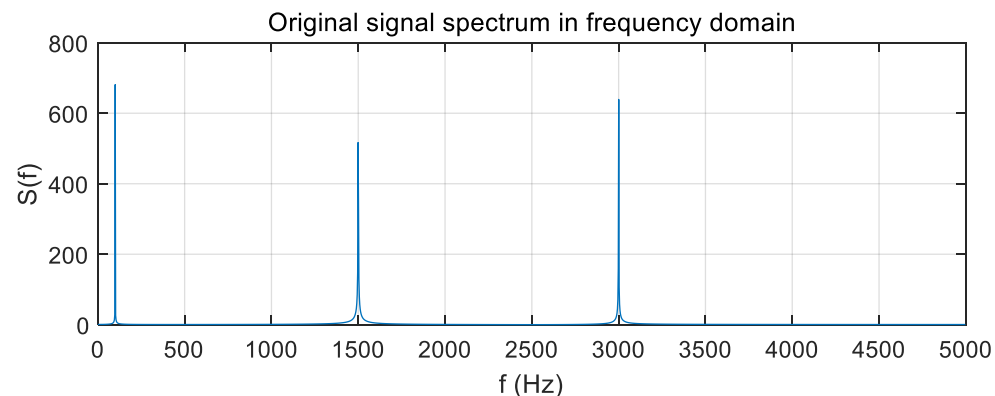
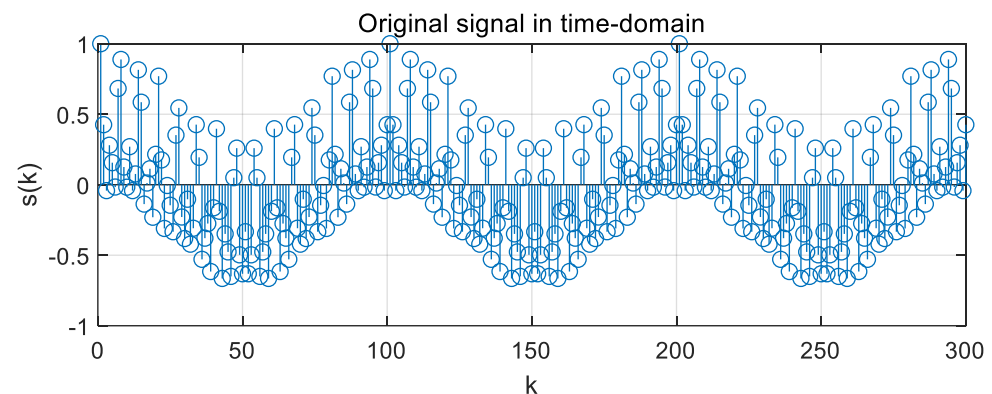
验证Verilog正确性:

将s作为testbench中的输入，Verilog代码输出作为s_bp，用MATLAB读入s_bp，再进行FFT变换，查看频谱

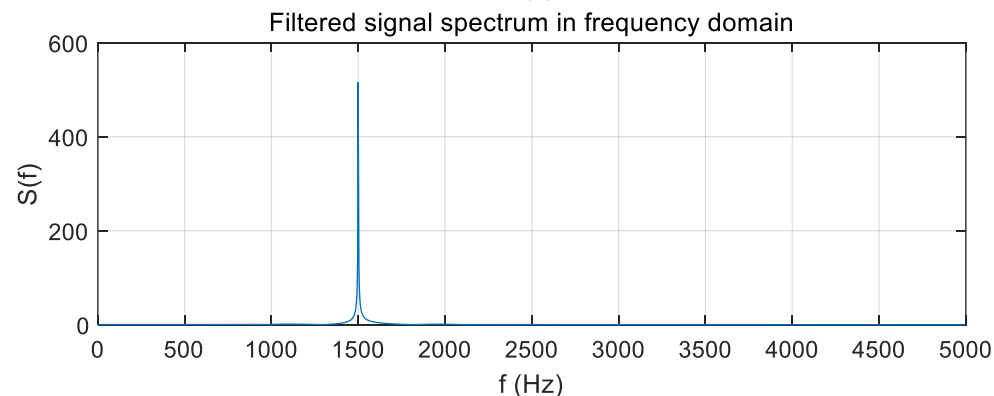
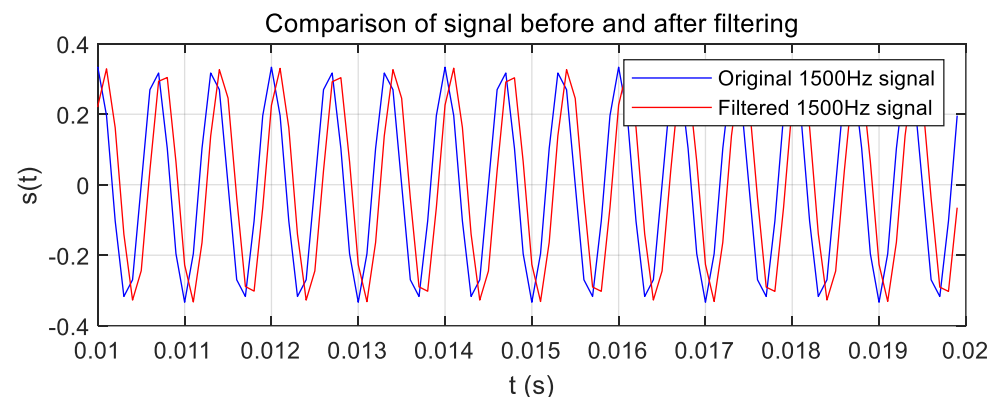
绘制滤波后时域信号对比(观察群延时)，以及滤波后频谱

```
figure(2);  
subplot(2,1,1);  
s2 = s2 ./ 3;  
plot(t(100:200), s2(100:200), 'blue', t(100:200), s_bp(100:200), 'red'); grid; xlim([0.01 0.02])  
legend('Original 1500Hz signal','Filtered 1500Hz signal');  
title('Comparison of signal before and after filtering');  
xlabel('t (s)');ylabel('s(t)');  
subplot(2,1,2);  
plot(f, abs(y_bp(1:len/2)));grid;  
title('Filtered signal spectrum in frequency domain');  
xlabel('f (Hz)');ylabel('S(f)');
```

滤波后1500Hz信号有群延时；100/3000Hz信号被滤去



滤波前



滤波后

一些级联/并联型IIR滤波器辅助设计的MATLAB函数

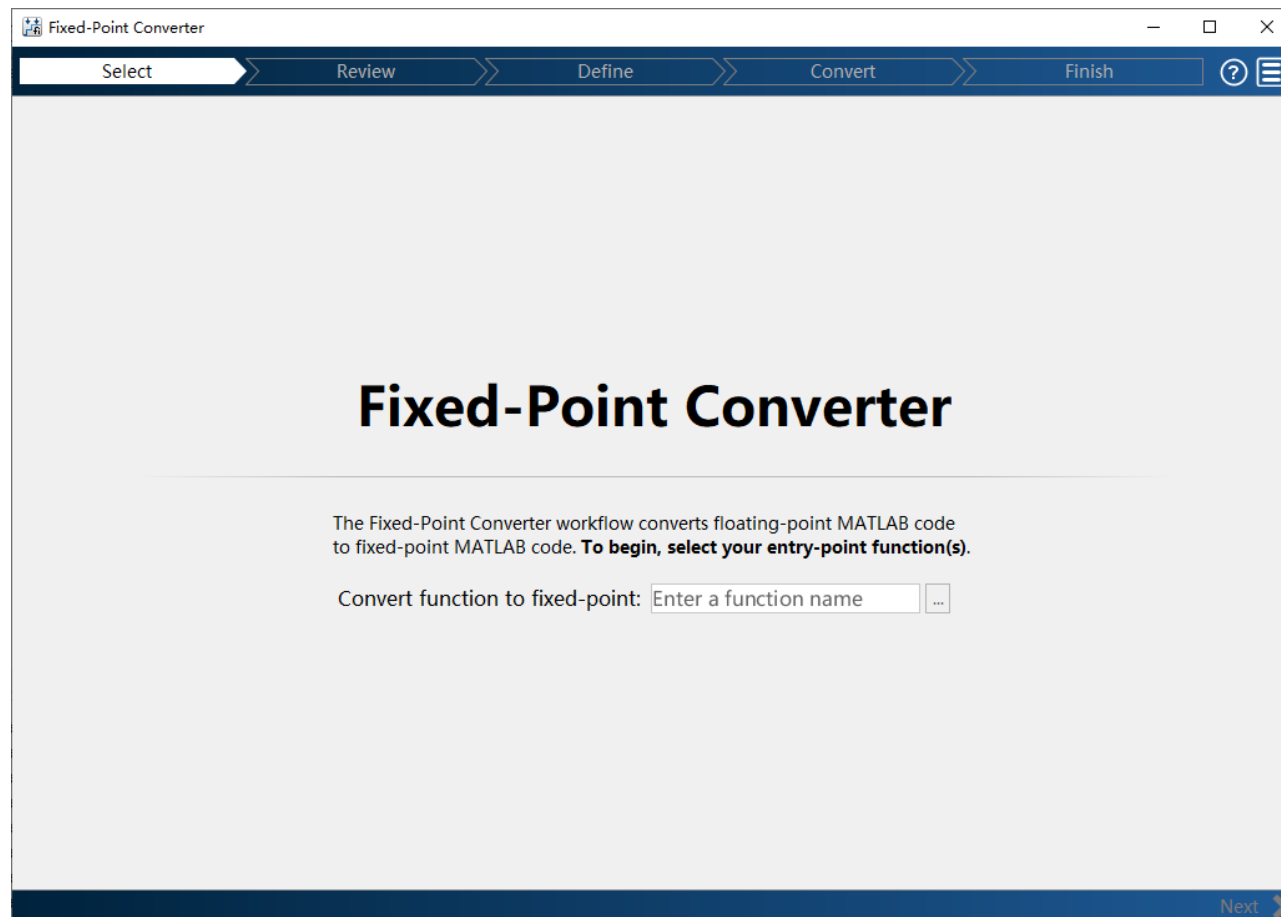
级联型

```
r = roots(h); % 求根, h为系数向量, r为根向量  
[z, p, k] = tf2zpk(b,a); % 求极零点, z为零点向量, p为极点向量, k为系数  
  
% 因式分解: factor()函数  
syms x  
F = factor(x^6-1); %输出F = [x - 1, x + 1, x^2 + x + 1, x^2 - x + 1]
```

并联型

```
[r, p, k] = residuez(b, a); % 并联I型  
[r, p, k] = residue(b, a); % 并联II型
```

更细化的定点化设计——Fixed-Point Converter工具箱



具体流程及使用方法以及MATLAB定点化仿真方法见文档

<https://ww2.mathworks.cn/help/fixedpoint/ref/fixedpointconverter-app.html>



目录

01 实验目的&平台

02 IIR滤波器介绍

03 用MATLAB设计IIR滤波器

04 性能验证

05 报告要求



■ 目标

- F_s : 10MHz ; 采样频率
- F_{stop1} : 0.5MHz; 通带截止频率1
- F_{pass1} : 1.5MHz; 通带下界频率
- F_{pass2} : 3.2MHz; 通带上界频率
- F_{stop2} : 4MHz ; 通带截止频率2
- A_{pass} : 1dB; 通带波纹
- A_{stop1} : 60dB ; 阻带衰减
- A_{stop2} : 60dB ; 阻带衰减

根据以上目标，以**巴特沃斯**模拟滤波器为原型，利用MATLAB的滤波器设计工具确定IIR数字滤波器的**阶数**、**定点**抽头系数、极零点等

用直接II型结构的IIR滤波器实现上页的参数

1. 根据MATLAB的设计结果画出对应的电路图（乘法器、加法器、寄存器）
2. 定点量化，验证软件实现的滤波效果；编写相应的Verilog代码，并验证其正确性
3. 用Vivado综合该设计（V7系列，直接在boards栏选取VC709型号），时钟频率需超过**125MHz**，给出对应的资源开销以及最大时钟频率（精确到10MHz），该设计记为 A_1
4. 利用VLSI技术对 A_1 进行优化，设计新架构 A_2 和 A_3 ，使其分别满足以下任意**两种**需求：
 - 时钟频率提升到原来的2倍以上；**
 - 吞吐率提升到原来的3倍以上；**
 - 面积减少到原来的一半**

- 关于实验报告的完成要求：
 - 所要求递交的实验报告应该至少包含以下内容：
 - 优化前的和优化后的IIR滤波器电路图
 - 主要功能模块的接口信号描述及实现
 - MATLAB和Verilog的验证结果
 - A_1 、 A_2 、 A_3 的设计思路与综合结果
 - 实验结果分析以及实验工作分工等
 - **注：**所有数据需配上对应的截图
 - 提交时间：**04月24日**23：59之前

谢谢!

