

基于 MATLAB 的数字滤波器设计

MATLAB 电子技术应用期末作业

王海飞

2022 年 2 月 17 日

目录

1	数字滤波器介绍	2
2	IIR 滤波器设计	3
2.1	模拟原型低通滤波器	3
2.2	模拟频域频带变换	5
2.3	冲激响应不变法	7
2.4	双线性变换法	9
3	FIR 滤波器设计	10
3.1	窗函数法设计思想	10
3.2	不同窗函数的性能	11
3.3	FIR 的线性相位特点	12
3.4	FIR 和 IIR 的比较	12
4	数字滤波器设计的 MATLAB 实现	12
4.1	IIR 滤波器设计的 MATLAB 实现	12
4.2	FIR 滤波器设计的 MATLAB 实现	15
4.3	使用完全设计函数	17
4.4	使用 Filter Designer (前 FDATool) 进行滤波器设计	17
5	基于 MATLAB 的数字滤波器的硬件实现	19
5.1	用 HDL Coder 工具生成 FIR 滤波器的 HDL 代码	19
5.2	基于 Xilinx System Generator 的 FIR 滤波器硬件设计	19
5.3	使用 Xilinx FIR Compiler IP 核	23
6	结论	26



1 数字滤波器介绍

数字滤波器是这样一个离散线性时不变系统, 它的系统函数具有如下形式:

$$H(z) = \frac{\sum_{k=0}^M b_k z^{-k}}{1 + \sum_{k=1}^N a_k z^{-k}} \quad (1)$$

对其作逆 z 变换, 可得

$$y[n] = \sum_{k=0}^M b_k x[n-k] - \sum_{k=1}^N a_k y[n-k] \quad (2)$$

由式 (2) 可见, 在时域上, 数字滤波器的响应是当下和过去一段时间的激励以及过去一段时间的响应的线性组合.

根据系数 a_k 和 b_k 的情况可以将数字滤波器进一步分为以下几类:

1. 若系统的响应只与当下和过去一段时间的激励有关, 即 $a_k = 0 (k = 1, \dots, N)$, 则该系统称为移动平均 (MA, Moving Average) 系统. 这类系统的系统函数没有 $|z| > 0$ 的极点, 故也称为全零点系统.
2. 若系统的响应与过去一段时间的激励无关, 但与过去一段时间的响应有关, 即 $b_k = 0 (k = 1, \dots, N)$, 则该系统称为自回归 (AR, Autoregressive) 系统. 这类系统的系统函数没有 $|z| > 0$ 的零点, 故也称为全极点系统.
3. 若 $a_k, b_k (k = 1, \dots, N)$ 不全为零, 则该系统称为自回归移动平均 (ARMA) 系统. 这类系统的系统函数有 $|z| > 0$ 的零点和极点, 故也称为零-极点系统.

将上述的1. 称为有限冲激响应滤波器 (Finite Impulse Response Filter, FIR Filter), 将2. 和3. 称为无限冲激响应滤波器 (Infinite Impulse Response Filter, IIR Filter). 这是因为前者的单位冲激响应是有限长的, 而后者的单位冲激响应是无限长的.

生活中常见的数字滤波器之一就是股票的均线, 它等于过去 N 个交易日的股价的平均数, 显然, 它是一个移动平均 (MA) 系统, 或 FIR 滤波器. 例如, MA60 就是 60 日均线, 它的时域差分方程为 $y[n] = \sum_{k=0}^{59} x[n-k]$. 均线可以在一定程度上过滤掉市场的噪声, 体现出一段时间内的趋势变化. 因此可以想象到, 均线是一个低通数字滤波器.

数字滤波器并非只是用来处理股票价格这样的离散信号的, 它还可以用来处理连续的模拟信号. 即, 将模拟信号经过 ADC (模数转换) 变为离散的数字信号, 经过数字滤波, 再通过 DAC (数模转换) 和滤波变回模拟信号.

与模拟滤波器相比, 数字滤波器在体积, 精度等各方面都显示出明显的优点, 而且, 数字滤波器除利用硬件电路实现之外, 还可用计算机软件实现. 因此, 随着数字技术的发展, 模拟滤波器的应用领域已逐步减少, 模拟信号也通常用数字滤波器来处理.



2 IIR 滤波器设计

数字滤波器设计就是根据所需滤波器的性能指标 (如通带和阻带的范围, 带内最大和最小衰减和采样频率等) 确定滤波器参数 (a_k 和 b_k) 的过程.

IIR 滤波器的设计方法分为直接法和间接法. 直接法要求解联立方程组, 需要计算机辅助设计. 间接法是借助模拟滤波器来设计数字滤波器. 由于模拟滤波器发展比较成熟, 因此间接法更加常用. 间接法包括冲激响应不变法, 阶跃响应不变法和双线性变换法.

本文介绍间接法中的冲激响应不变法和双线性变换法.

2.1 模拟原型低通滤波器

间接法需借助模拟滤波器来设计数字滤波器, 因此有必要了解模拟滤波器.

所谓低通原型滤波器, 就是将低通通带边沿频率归一化为 1. 各种类型的滤波器如低通, 带通, 高通, 带阻滤波器, 都可以由原型低通滤波器通过变量替换的方法得到.

2.1.1 巴特沃斯 (Butterworth) 低通滤波器

巴特沃斯滤波器的幅频特性模平方为

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

其中 N 是滤波器的阶数, Ω_c 是滤波器的截止角频率, 下标 a 表示模拟 (analog). 可以证明, 幅频特性在 $\Omega = 0$ 处的前 $(2N - 1)$ 阶导数等于零, 因此它也叫最平幅度特性滤波器.

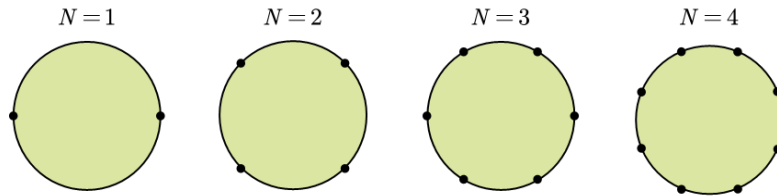


图 1: Butterworth 滤波器的幅频特性模平方的极点

由于希望冲激响应 $h_a(t)$ 是实的, 则有

$$H_a(-s) = \int_0^\infty h_a(t)e^{st}dt = \int_0^\infty [h_a(t)e^{-st}] = H_a^*(s) \quad (4)$$

因此

$$|H_a(s)|^2 = H_a^*(s)H_a(s) = H_a(s)H_a(-s) \quad (5)$$

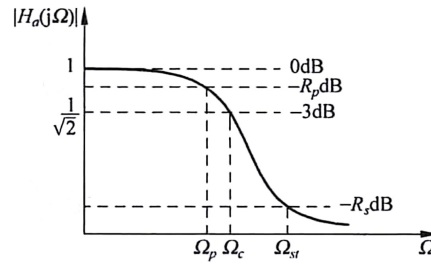


图 2: Butterworth 滤波器的幅度特性和性能指标

这说明 $|H_a(s)|^2$ 的零极点的分布关于虚轴镜像对称. 这些零极点一半是 $H_a(s)$ 的, 一半是 $H_a(-s)$ 的. 如果希望系统稳定, 应取左半平面的极点为 $H_a(s)$ 的极点. 如果希望系统是最小相位的, 应取左半平面的零点为 $H_a(s)$ 的零点.

$|H_a(jΩ)|^2$ 的极点分布在半径为 $Ω_c$ 的圆上, 但不会落在虚轴上 (见图1), 零点在 $∞$ 处. Butterworth 滤波器的极点就是位于左半平面的那一半极点.

Butterworth 滤波器原型设计的 MATLAB 实现为:

$$[z, p, k] = \text{buttapp}(n)$$

其中 n 为模拟滤波器的阶数, z, p, k 分别是设计出的原型滤波器的零点, 极点和增益.

2.1.2 根据技术指标确定所需的最低阶数

见图 (2), $Ω_p$ 为所需的通带截止频率, $R_p(\text{dB})$ 为 $Ω_p$ 处的衰减, 它应小于通带最大衰减. $Ω_s$ 为所需的阻带截止频率, $R_s(\text{dB})$ 为 $Ω_s$ 处的衰减, 它应大于阻带最小衰减:

$$\begin{aligned} R_p &\geq -20 \lg |H_a(jΩ_p)| \\ R_s &\leq -20 \lg |H_a(jΩ_s)| \end{aligned} \quad (6)$$

代入 Butterworth 原型滤波器的幅频响应 (式3) 即可以求得所需 Butterworth 滤波器的最低阶数.

MATLAB 实现如下:

$$[n, Wn] = \text{buttord}(Wp, Ws, Rp, Rs, 's')$$

其中 w_p 为通带截止频率, w_s 为阻带截止频率, R_p 为通带最大衰减, R_s 为阻带最小衰减, n 为所需的最低阶数, w_n 为截止频率 (衰减为 3dB 处的频率, 不一定等于通带截止频率 w_p).

2.1.3 切比雪夫 (Chebyshev) 低通滤波器

Butterworth 滤波器的通带内与理想滤波器的误差分布是不均匀的, 越靠近通带边缘, 误差越大. Chebyshev 滤波器可以使通带 (Chebyshev I 型) 或阻带 (Chebyshev II 型) 内的误差分布均匀. 如果希望同时做到通带和阻带内的误差分布均匀, 则需要椭圆函数滤波器. 限于篇幅, 不再详细介绍. 图 (3) 展示了 Chebyshev 低通滤波器的幅度特性.

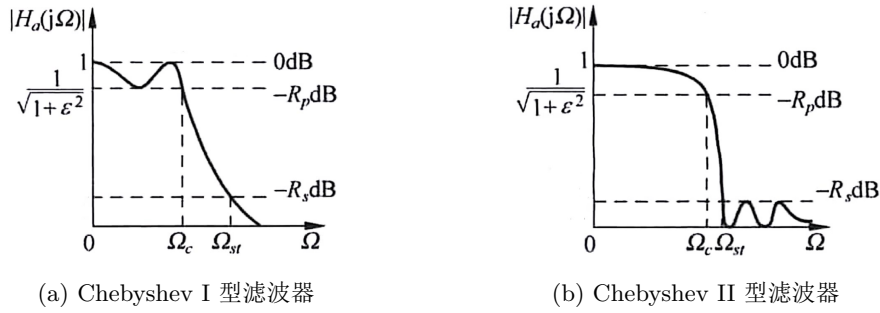


图 3: Chebyshev 滤波器的幅度特性和性能指标

MATLAB 实现如下:

1. Chebyshev I 型滤波器原型设计的 MATLAB 实现为:

$$[z, p, k] = \text{cheblap}(n, R_p)$$

其中 n 为模拟滤波器的阶数, R_p 为指定通带内最大衰减, z, p, k 分别是设计出的滤波器的零点、极点和增益.

2. Chebyshev II 型滤波器原型设计的 MATLAB 实现为:

$$[z, p, k] = \text{cheb2ap}(n, R_s)$$

其中 R_s 为指定阻带内最小衰减, 其余参数含义同上.

3. 椭圆滤波器滤波器原型设计的 MATLAB 实现为:

$$[z, p, k] = \text{ellipap}(n, R_p, R_s)$$

参数含义同上.

2.2 模拟频域频带变换

可以由原型低通滤波器通过变量替换的方法得到其他类型的滤波器, 即将原型滤波器的系统函数的 s 换成 $G(s)$:

$$(j\bar{\Omega} = \bar{s} = G(s) = G(j\Omega)) \quad (7)$$

具体方法如下:

1. 低通滤波器: 令

$$\bar{s} = s/\Omega_p, \quad \bar{\Omega} = \Omega/\Omega_p \quad (8)$$

其中 Ω_p 为通带截止频率, 即通带边缘处, 频响为通带最大衰减 R_p 的频率, 见图 (2).



2. 高通滤波器: 令

$$\bar{s} = \Omega_p/s, \quad \bar{\Omega} = -\Omega_p/\Omega \quad (9)$$

其中 Ω_p 为通带截止频率.

3. 带通滤波器: 令

$$\begin{aligned} \bar{s} &= \frac{s^2 + \Omega_{p1}\Omega_{p2}}{(\Omega_{p2} - \Omega_{p1})s} \\ \bar{\Omega} &= \frac{\Omega^2 - \Omega_{p1}\Omega_{p2}}{(\Omega_{p2} - \Omega_{p1})\Omega} \\ B_w &= \Omega_{p2} - \Omega_{p1}, \quad \Omega_{p0} = \sqrt{\Omega_{p2}\Omega_{p1}} \end{aligned} \quad (10)$$

其中 Ω_{p2} 为通带上截止频率, Ω_{p1} 为通带下截止频率, B_w 为带宽, Ω_{p0} 为通带的中心频率.

4. 带阻滤波器: 令

$$\begin{aligned} \bar{s} &= \frac{\bar{\Omega}_s(\Omega_{s2} - \Omega_{s1})s}{s^2 + \Omega_{s1}\Omega_{s2}} \\ \bar{\Omega} &= \frac{\bar{\Omega}_s(\Omega_{s2} - \Omega_{s1})\Omega}{\Omega_{s1}\Omega_{s2} - \Omega^2} \end{aligned} \quad (11)$$

其中 Ω_{s2} 为阻带带上截止频率, Ω_{s1} 为阻带下截止频率. 阻带的截止频率即阻带边缘处, 频响为阻带最小衰减 R_s 的频率, 见图 (1). $\bar{\Omega}_s$ 是原型滤波器的阻带截止频率, 需要进一步分析确定.

总之, 各种类型的滤波器可以由原型低通滤波器通过模拟频域频带变换的方法得到.

在 MATLAB 中, 模拟频域频带变换可以用如下函数实现:

1. lp2lp: 从滤波器原型到低通模拟滤波器的转换函数

$[bt, at] = lp2lp(b, a, wo)$. wo 为指定截止频率, b 和 a 为原滤波器原型的传递函数的分子系数和分母系数, bt 和 at 为所需低通滤波器的传递函数的分子系数和分母系数.

2. lp2hp: 从滤波器原型到高通模拟滤波器的转换函数

$[bt, at] = lp2hp(b, a, wo)$. 各参数含义同 $lp2lp$.

3. lp2bp: 从滤波器原型到带通模拟滤波器的转换函数

$[bt, at] = lp2bp(b, a, wo, bw)$. wo 为指定通带的中心频率, bw 为指定带宽, 其余参数含义同 $lp2lp$.

4. lp2bs: 从滤波器原型到带阻模拟滤波器的转换函数.

$[bt, at] = lp2bs(b, a, wo, bw)$. 参数含义同 $lp2bp$.

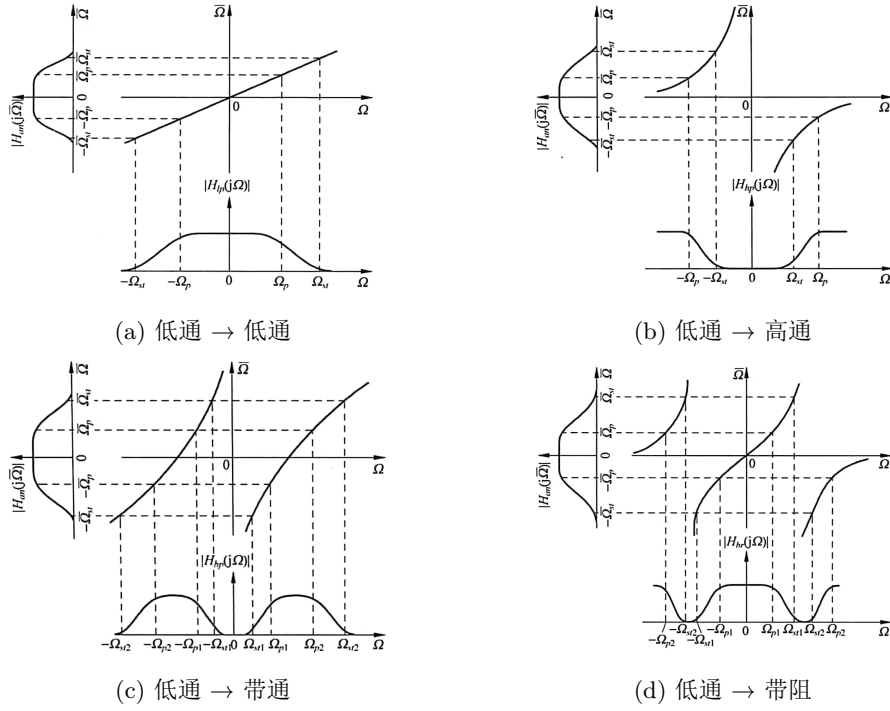


图 4: 模拟频域频带变换示意图

2.3 冲激响应不变法

冲激响应不变法 (Impulse Invariance) 的思想是对模拟滤波器的单位冲激响应进行取样。设模拟滤波器的系统函数为

$$H_a(s) = \sum_{k=1}^N \frac{A_k}{s - s_k} \quad (12)$$

对其做逆拉普拉斯变换得到单位冲激响应

$$h_a(t) = \sum_{k=1}^N A_k e^{s_k t} u(t) \quad (13)$$

其中 $u(t)$ 为阶跃函数。再对上式以周期 T 抽样得

$$h[n] = h_a(t)|_{t=nT} = \sum_{k=1}^N A_k e^{s_k nT} u[n] \quad (14)$$

对上式做 z 变换得:

$$H(z) = \sum_{n=0}^{\infty} \left(\sum_{k=1}^N A_k e^{s_k nT} u[n] \right) z^{-n} = \sum_{k=1}^N \frac{A_k}{1 - e^{s_k T} z^{-1}} \quad (15)$$



可见, 冲激响应不变法只需将模拟滤波器的系统函数 (部分分式形式) 的分母 $(s - s_k)$ 换成 $(1 - e^{s_k T} z^{-1})$ 就得到对应数字滤波器的系统函数。

下面讨论 s 域和 z 域的映射关系. 当 $z = e^{sT}$ 时, 抽样序列的 z 变换等于抽样信号 $\sum_{n=-\infty}^{\infty} h_a(t) \delta(t - nT)$ 的拉普拉斯变换:

$$\begin{aligned} \mathcal{L} \left[\sum_{n=-\infty}^{\infty} h_a(t) \delta(t - nT) \right] &= \sum_{n=-\infty}^{\infty} \int_0^{\infty} h_a(t) \delta(t - nT) e^{-st} dt \\ &= \sum_{n=-\infty}^{\infty} h_a(nT) e^{-snT} \\ &= \sum_{n=-\infty}^{\infty} h[n] z^{-n} \\ &= \mathcal{Z}[h[n]] = H(z) \end{aligned} \quad (16)$$

对 $\sum_{n=-\infty}^{\infty} \delta(t - nT)$ 做傅里叶级数展开得

$$\sum_{n=-\infty}^{\infty} \delta(t - nT) = \sum_{n=-\infty}^{\infty} \left(\frac{1}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} \delta(t) e^{-jk \frac{2\pi}{T} t} dt \right) e^{jk \frac{2\pi}{T} t} = \frac{1}{T} \sum_{n=-\infty}^{\infty} e^{jk \frac{2\pi}{T} t} \quad (17)$$

因此

$$\begin{aligned} \mathcal{L} \left[\sum_{n=-\infty}^{\infty} h_a(t) \delta(t - nT) \right] &= \mathcal{L} \left[h_a(t) \frac{1}{T} \sum_{n=-\infty}^{\infty} e^{jk \frac{2\pi}{T} t} \right] \\ &= \frac{1}{T} \sum_{n=-\infty}^{\infty} H_a \left(s - j \frac{2\pi}{T} k \right) \end{aligned} \quad (18)$$

因此数字滤波器的频率响应为

$$\begin{aligned} H(z)|_{z=e^{j\omega}} &= \frac{1}{T} \sum_{n=-\infty}^{\infty} H_a \left(s - j \frac{2\pi}{T} k \right) \Big|_{s=j\omega/T} \\ &= \frac{1}{T} \sum_{n=-\infty}^{\infty} H_a \left(j \frac{\omega}{T} - j \frac{2\pi}{T} k \right) \end{aligned} \quad (19)$$

上式就是数字滤波器的频率响应 $H(e^{j\omega})$ 和模拟滤波器的频率响应 $H_a(j\Omega)$ 的关系. 可见数字滤波器的频响是模拟滤波器的频响的周期性叠加, 见图 (5). 如果模拟滤波器的频响带限于折叠频率 π/T 之内, 则频谱不会混叠. 如果频谱混叠严重, 频率特性高频处会严重失真, 因此冲激响应不变法只适用于低通和带限的场合。

冲激响应不变法的 MATLAB 实现如下:

$$[bz, az] = \text{impinvar}(b, a, fs)$$

其中 b 和 a 为模拟滤波器传递函数的分子和分母系数, fs 为采样频率. bz 和 az 为冲激响应不变法设计出的数字滤波器传递函数的分子和分母系数。

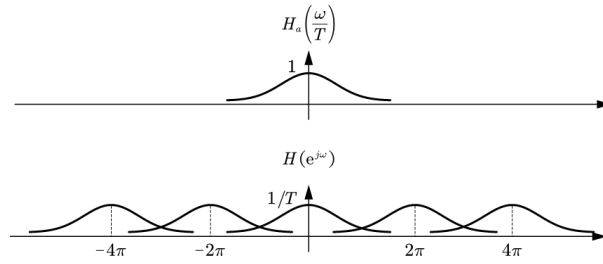


图 5: 冲激响应不变法: 数字滤波器的频响是模拟滤波器的频响的周期性叠加

2.4 双线性变换法

为了克服频谱混叠失真现象, 希望在 s 和 z 之间建立单值映射关系. 多值性是由指数函数 $z = e^{sT}$ 引起的, 为了消除多值性, 当 T 较小时, 可将 e^{sT} 近似为

$$e^{sT} \approx 1 + sT \approx \frac{1 + sT/2}{1 - sT/2} \quad (20)$$

这就是双线性变换法 (Bilinear Transformation) 的思路.

再来考察这种近似是否合理, 设 $s = \sigma + j\Omega$, $z = re^{j\omega}$, 则

$$\begin{aligned} re^{j\omega} &= \frac{1 + \frac{T}{2}(\sigma + j\Omega)}{1 - \frac{T}{2}(\sigma + j\Omega)} \\ &= \sqrt{\frac{(1 + \frac{T}{2}\sigma)^2 + (\frac{\Omega T}{2})^2}{(1 - \frac{T}{2}\sigma)^2 + (\frac{\Omega T}{2})^2}} \exp \left\{ j \left[\arctan \left(\frac{\frac{\Omega T}{2}}{1 + \frac{T}{2}\sigma} \right) + \arctan \left(\frac{\frac{\Omega T}{2}}{1 - \frac{T}{2}\sigma} \right) \right] \right\} \end{aligned} \quad (21)$$

由上式可见, 当 $\sigma < 0$ 时, $r < 1$, 该映射将 s 域的左半平面映射到 z 域的单位圆内, 因而滤波器的稳定性可以保证.

令 $\sigma = 0$ 得频率的映射关系为

$$\omega = 2 \arctan \left(\frac{\Omega T}{2} \right) \quad (22)$$

单值映射可以避免频谱混叠, 但是由上式可知, 频率的映射是非线性的, 会引起相频特性的失真. 因此在设计滤波器时, 需要采用“频率预畸” (Frequency Prewarping) 的办法, 即给定数字滤波器的截止频率 ω_i , 应根据式 (22) 将模拟滤波器的频率预畸为 $\Omega_i = \frac{2}{T} \tan \left(\frac{\omega_i}{2} \right)$.

双线性变换法的 MATLAB 实现如下:

$$[bd, ad] = \text{bilinear}(b, a, fs)$$

其中 b 和 a 为模拟滤波器传递函数的分子和分母系数. fs 为采样频率. bd 和 ad 为双线性变换法设计出的数字滤波器传递函数的分子和分母系数.

在 MATLAB 中, 除了用传递函数的分子和分母系数表示外, 还可以用零极点增益的形式:

$$[zd, pd, kd] = \text{bilinear}(z, p, k, fs)$$



3 FIR 滤波器设计

FIR 滤波器和 IIR 滤波器的设计方法是不同的. 设计 FIR 滤波器主要有三种方法: 窗函数法, 频率抽样法和最优化方法. 本文讨论最常用的窗函数法.

FIR 滤波器的单位冲激响应是有限长的. 若长度为 N , 则 $h(n)$ 在 $0 < n \leq N - 1$ 范围非零, 其系统函数为

$$H(z) = \sum_{n=0}^{N-1} h(n)z^{-n} \quad (23)$$

它是 z^{-1} 的 $(N - 1)$ 阶多项式. 若 FIR 滤波器的系统函数是 z^{-1} 的 n 阶多项式, 则称 FIR 滤波器的阶数为 n . 可见滤波器长度等于滤波器阶数加一.

3.1 窗函数法设计思想

窗函数设计法的思想如下:

1. 根据所需滤波器的类型 (低通, 带通, 高通, 带阻), 确定理想的频率响应 $H_d(e^{j\omega})$.
2. 将理想频率响应变换到时域. 求理想频率响应变换的离散时间傅里叶逆变换 (IDTFT), 得到理想滤波器的单位冲激响应 $h_d[n]$:

$$h_d[n] = \text{IDTFT}[H_d(e^{j\omega})] = \frac{1}{2\pi} \int_{-\pi}^{\pi} H_d(e^{j\omega}) e^{j\omega n} d\omega \quad (24)$$

3. 理想滤波器的单位冲激响应 $h_d(n)$ 是无限长且非因果的. 要得到有限长的冲激响应, 可以将 $h_d(n)$ 乘以一个长度为 N 的窗函数 (Window function) $w_N[n]$, $n = 0, 1, \dots, N - 1$:

$$h[n] = h_d[n]w_N[n] \quad (25)$$

窗的长度 N 和窗的形状决定了滤波器最终的性能.

4. 对加窗后的有限长冲激响应做离散时间傅里叶变换 (DTFT), 变回频域, 得到实际的频率响应:

$$\begin{aligned} H(e^{j\omega}) &= \text{DTFT}[h[n]] = \text{DTFT}[h_d[n]w[n]] \\ &= \frac{1}{2\pi} [H_d(e^{j\omega}) * W(e^{j\omega})] = \frac{1}{2\pi} \int_{-\pi}^{\pi} H_d(e^{j\theta}) W(e^{j(\omega-\theta)}) d\theta \end{aligned} \quad (26)$$

5. 检验技术指标是否被满足, 如果不满足则需改变 N 或窗函数类型, 重新计算直到满足要求为止.



3.2 不同窗函数的性能

由 DTFT 的卷积定理知, 最终的频率响应是理想频率响应和窗函数的频谱的卷积. 以理想低通滤波器和矩形窗频谱的卷积为例, 矩形窗的频谱的幅度函数为 sinc 函数, 卷积的过程见图 (6).

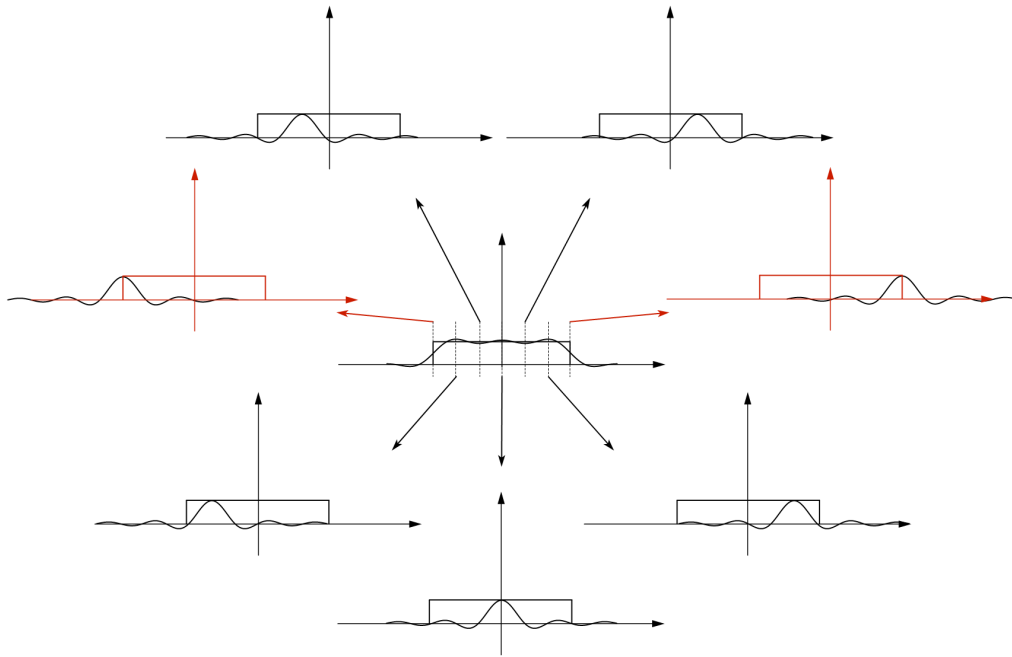


图 6: 理想频率响应 (图中矩形) 和矩形窗函数的频谱 (图中 sinc 函数) 的卷积.

一般来说, 窗函数的频谱都有类似于 sinc 函数那样的主瓣 (Mainlobe) 和旁瓣 (Sidelobe). 由图 (6) 可见, 主瓣的宽度决定了过渡带的宽度, 旁瓣的峰值决定了通带最大衰减和阻带最小衰减.

不同的窗函数拥有不同的频谱, 因此最终的频率响应的形状和窗函数的形状是密切相关的. 常用的窗函数有矩形窗, 三角形窗 (Bartlett 窗), Hanning 窗 (升余弦窗), Hamming 窗 (改进的升余弦窗), Blackman 窗 (二阶升余弦窗) 以及 Kaiser 窗. 它们各有不同的性能, 矩形窗的主瓣宽度最小, 但旁瓣峰值最大. 其他非矩形窗的旁瓣峰值小, 但主瓣宽度大. Kaiser 窗函数包含一个参数 β , 通过调整它, 可以实现主瓣宽度和旁瓣峰值的取舍.

窗函数法的截止频率定义为衰减为 -6dB 处的频率, 即幅度为一半, 而非 -3dB . 由图 (6) 可见, 截止频率在过渡带的中点处.



3.3 FIR 的线性相位特点

一般来说,窗函数 $w[n]$ 关于 $\frac{N-1}{2}$ 对称. 如果理想滤波器的冲激响应 $h_d[n]$ 也关于 $\frac{N-1}{2}$ 对称,那么实际滤波器的冲激响应 $h[n] = h_d[n]w[n]$ 也关于 $\frac{N-1}{2}$ 对称. 此时可以证明,滤波器的系统函数 $H(e^{j\omega}) = \sum_{n=0}^{N-1} h[n]e^{-j\omega n}$ 可以写成 $H(e^{j\omega}) = A_{\text{even}}(e^{j\omega})e^{-j\omega \frac{N-1}{2}}$ 的形式,其中 A_{even} 是 ω 的实偶函数. 这说明,滤波器具有线性相位特性. 线性相位特性的好处在于通带内所有频率的信号的延时是相同的.

非线性相位滤波器用 IIR 滤波器实现需要的阶数更小,因此 FIR 主要用于设计线性相位滤波器.

另外,如果理想滤波器的冲激响应 $h_d[n]$ 关于 $\frac{N-1}{2}$ 反对称,则滤波器的系统函数可以写成 $H(e^{j\omega}) = jA_{\text{odd}}(e^{j\omega})e^{-j\omega \frac{N-1}{2}}$ 的形式,其中 A_{odd} 是关于 ω 的实奇函数. 此时的滤波器除了具有线性相位,还具有 $\frac{\pi}{2}$ 的相移,可用于设计离散希尔伯特变换器和微分器.

3.4 FIR 和 IIR 的比较

1. IIR 数字滤波器的相位非线性,而 FIR 数字滤波器可以做到严格的线性相位.
2. IIR 滤波器有历史的输出参与反馈,同 FIR 相比在相同阶数时取得更好的滤波效果.
3. IIR 数字滤波器采用递归型结构,由于运算中的舍入处理,使误差不断累积,有时会对信号产生微弱的寄生振荡. 而 FIR 滤波器没有反馈回路,传递函数没有极点,总是稳定的.

4 数字滤波器设计的 MATLAB 实现

借助 MATLAB 可以方便地进行数字滤波器的设计.

4.1 IIR 滤波器设计的 MATLAB 实现

IIR 滤波器设计的 MATLAB 实现步骤如下:

1. 由 $\omega = \Omega T$ 将所需设计数字滤波器频率指标 ω 转换为相应模拟滤波器频率指标 Ω . 如果给出的指标已经是模拟频率指标,则可以省略这一步. 如果使用双线性变换法,需要将模拟滤波器的频率预畸为 $\Omega_i = \frac{2}{T} \tan\left(\frac{\omega_i}{2}\right)$.
2. 选择模拟滤波器类型,并根据技术指标求其阶数,使用的函数有 `buttord`, `cheblord`, `cheb2ord`, `elliord` 等. 见第2.1.2节.
3. 设计模拟低通滤波器原型,使用的函数有 `buttap`, `cheblap`, `cheb2ap`, `ellipap` 等. 见第2.1节.



4. s 域变量代换得到低通, 高通, 带通, 带阻模拟滤波器, 使用的函数有 `lp2lp`, `lp2hp`, `lp2bp`, `lp2bs` 等. 见第2.2 节.
5. 使用冲激响应不变法或双线性变换法完成 s 域到 z 域的映射, 得到数字滤波器. 分别使用函数 `impinvar` 和 `bilinear`. 见第2.3, 2.4 节.

习题 根据下列技术指标完成 IIR 数字滤波器设计.

1. 通带范围 300—500Hz, 带内最大衰减 $R_p = -3\text{dB}$
2. 阻带范围 $\leq 250\text{Hz}$, $\geq 550\text{Hz}$, 带内最小衰减 $R_s = -40\text{dB}$
3. 采样频率 $F_s = 2000\text{Hz}$

给出的指标已经是模拟频率指标, 对应的数字频率指标为通带 0.3π — $0.5\pi\text{rad/s}$, 阻带 0 — $0.25\pi\text{rad/s}$, 0.55π — $\pi\text{rad/s}$.

4.1.1 冲激响应不变法

采用冲激响应不变法, 选择 Butterworth 原型滤波器. MATLAB 代码如下:

```
1  clc;
2  clear;
3
4  Fs = 2000; % sampling frequency
5  Wp = [300 * 2 * pi 500 * 2 * pi]; % passband
6  Ws = [250 * 2 * pi 550 * 2 * pi]; % stopband
7  Rp = 3; % maximum attenuation in passband
8  Rs = 40; % minimum attenuation in stopband
9
10 [n, Wn] = buttord(Wp, Ws, Rp, Rs, 's'); % lowest order
11
12 % zeros, poles and gain of an n-order Butterworth analog lowpass filter
   prototype
13 [z, p, k] = buttap(n);
14
15 % convert zero-pole-gain form to denominator-numerator form
16 [b0, a0] = zp2tf(z, p, k);
17
18 % lowpass to bandpass
19 [b, a] = lp2bp(b0, a0, sqrt(Wn(1) * Wn(2)), Wn(2) - Wn(1));
20
```



```
21 [bz, az] =impinvar(b, a, Fs); % impulse invariance method
22
23 freqz(bz, az); % plot the frequency response
```

其中需要用到函数 `zp2tf`, 它将零极点增益形式转化为传递函数形式。
传递函数形式如下:

$$H(s) = \frac{B(s)}{A(s)} = \frac{b_1 s^{n-1} + \cdots + b_{n-1} s + b_n}{a_1 s^{m-1} + \cdots + a_{m-1} s + a_m} \quad (27)$$

零极点增益形式如下:

$$H(s) = \frac{Z(s)}{P(s)} = k \frac{(s - z_1)(s - z_2) \cdots (s - z_m)}{(s - p_1)(s - p_2) \cdots (s - p_n)} \quad (28)$$

代码运行结果如图 (7) 所示. 工作区变量 `n` 的值为 15, 即所需滤波器的阶数为 15.

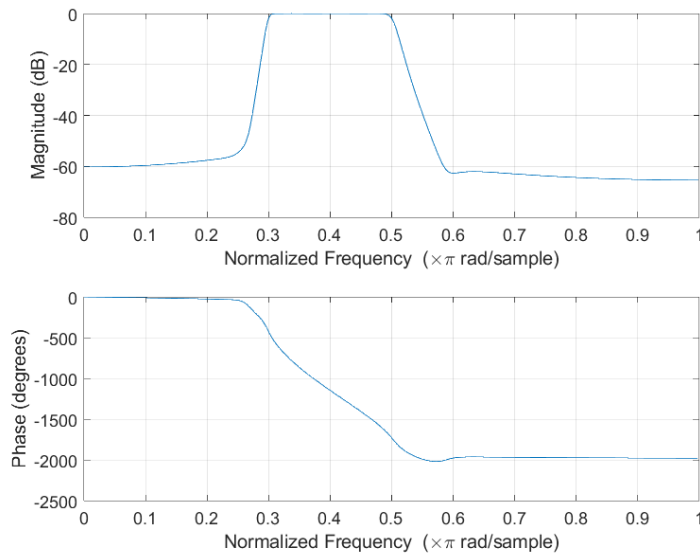


图 7: 冲激响应不变法设计出的 IIR 滤波器的频率特性

4.1.2 双线性变换法

采用双线性变换法, 选择 Butterworth 原型滤波器. MATLAB 代码如下:

```
1 clc;
2 clear;
3
4 Fs = 2000; % sampling frequency
```



```
5 Wp = [2*Fs*tan(300*pi/Fs) 2*Fs*tan(500*pi/Fs)]; % passband, notice that
    frequency prewarping is performed
6 Ws = [2*Fs*tan(250*pi/Fs) 2*Fs*tan(550*pi/Fs)]; % stopband, notice frequency
    prewarping is performed
7 Rp = 3; % maximum attenuation in passband
8 Rs = 40; % minimum attenuation in stopband
9
10 [n, Wn] = buttord(Wp, Ws, Rp, Rs, 's'); % lowest order
11
12 % zeros, poles and gain of an n-order Butterworth analog lowpass filter
    prototype
13 [z, p, k] = buttap(n);
14
15 % convert zero-pole-gain form to transfer-function form
16 [b0, a0] = zp2tf(z, p, k);
17
18 % lowpass to bandpass
19 [b, a] = lp2bp(b0, a0, sqrt(Wn(1) * Wn(2)), Wn(2) - Wn(1));
20
21 [bz, az] = bilinear(b, a, Fs); % impulse invariance method
22
23 freqz(bz, az); % plot the frequency response
```

代码运行结果如图 (8) 所示. 工作区变量 n 的值为 12, 即所需滤波器的阶数为 12.

4.2 FIR 滤波器设计的 MATLAB 实现

采用窗函数法, 选择 Kaiser 窗函数. MATLAB 代码如下:

```
1 clc;
2 clear;
3
4 Fs = 2000; % sampling frequency
5 f = [250 300 500 550]; % band edges
6 mags = [0 1 0]; % band amplitude
7 % 10^(-40/20) = 0.01
8 devs = [0.01 10^(-3/20) 0.01]; % maximum allowable deviation
9
10 % filter order n, normalized frequency band edges Wn, and shape factor beta
    that specify a Kaiser window for use with the fir1 function.
11 [n, Wn, beta, ftype] = kaiserord(f, mags, devs, Fs);
12
```

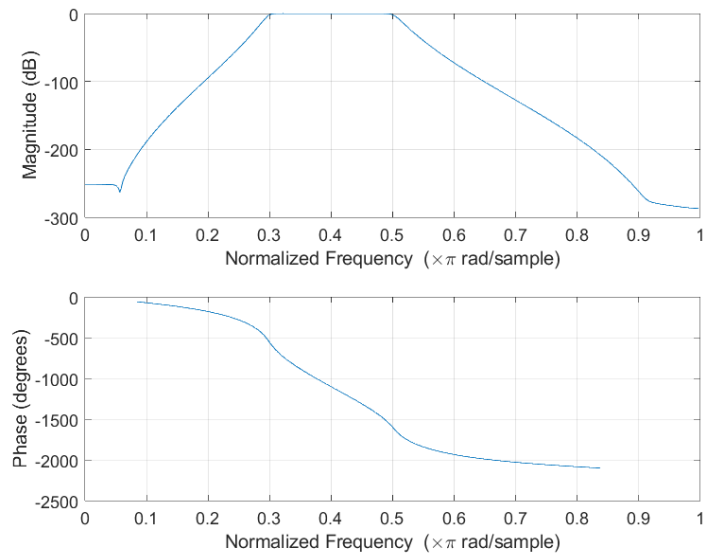


图 8: 双线性变换法设计出的 IIR 滤波器的频率特性

```
13 % designs a lowpass, highpass, bandpass, bandstop, or multiband filter
    depending on the value of ftype and the number of elements of Wn.
14 b = fir1(n, Wn, ftype, kaiser(n + 1, beta));
15
16 % plot the frequency response
17 freqz(b);
```

其中, `kaiserord` 可以根据技术指标求出所需的 Kaiser 窗函数阶数 n 和参数 `beta`。此外, 它还返回了 W_n 和 `ftype`, W_n 表示归一化截止频率 $W_n(\pi \cdot \text{rad/s})$ 。截止频率在过渡带的中点, 此例中 $W_n=[0.275, 0.525]$ 。`ftype` 为滤波器类型, 有 'low', 'bandpass', 'high', 'stop' 等。

`kaiser(L, beta)` 返回一个参数为 `beta`, 窗口长度为 L 的 Kaiser 窗。注意窗口长度比滤波器阶数 n 大一, 因为 $0 \leq k \leq n$ 中有 $(n+1)$ 个 k 。

`fir1(n, Wn, ftype, window)` 函数根据滤波器阶数 n , 截止频率 W_n , 滤波器类型 `ftype` 和窗函数类型 `window`, 返回设计出的 FIR 滤波器的系数 b_n 。

代码运行结果如图 (9) 所示。可见通带内相位特性是线性的。查看工作区变量 n 的值为 90, 说明在相同技术指标下 FIR 滤波器所需要的阶数更高 (IIR 双线性变换法阶数仅为 12)。

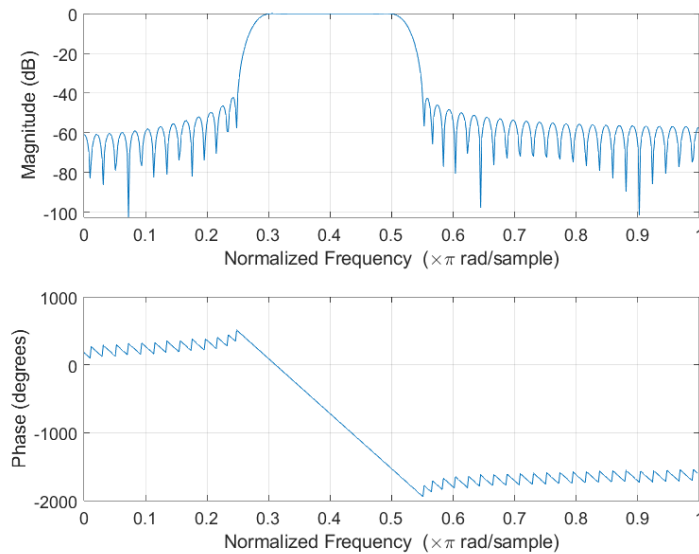


图 9: 窗函数法设计出的 FIR 滤波器的频率特性

4.3 使用完全设计函数

使用 Signal Processing Toolbox 中的 IIR 数字滤波器设计函数, 可以一次性完成 IIR 滤波器的设计. 这些函数包括 `butter`, `cheby1`, `cheby2`, `ellip`.

4.4 使用 Filter Designer (前 FDATool) 进行滤波器设计

数字滤波器的设计还可以借助 MATLAB 提供的 Filter Designer (旧版本的 FDATool) 快速实现.

在 MATLAB 命令行中输入 `filterDesigner`, 在弹出的图形界面窗口中 (见图10) 可以方便地进行设计.

以 FIR 设计为例, 如图 (10) 所示, 在 Response Type 一栏中选择 Bandpass, 在 Design Method 一栏中选择 FIR 并在下拉菜单中选择 Window, 即选择窗函数法进行 FIR 带通滤波器设计. 在 Filter Order 一栏中选择 Minimum order, 在 Options 一栏中选择窗函数类型为 Kaiser (Scale Passband 的作用是使得通带中心频率的增益为 0dB). 在 Frequency Specifications 一栏中选择 Units 为 Hz, 并输入通带和阻带截止频率, 在 Magnitude Specifications 一栏中设置通带最大衰减和阻带最小衰减. 完成以上设置后点击下方 Design Filter 即可完成滤波器设计.

完成设计后可以将设计结果导出到工作区: 打开 File 菜单栏, 点击 Export 即可将滤波器的系数导出到工作区.

对比发现, Filter Designer 工具导出的滤波器系数和第4.2节得到的系数完全相同.

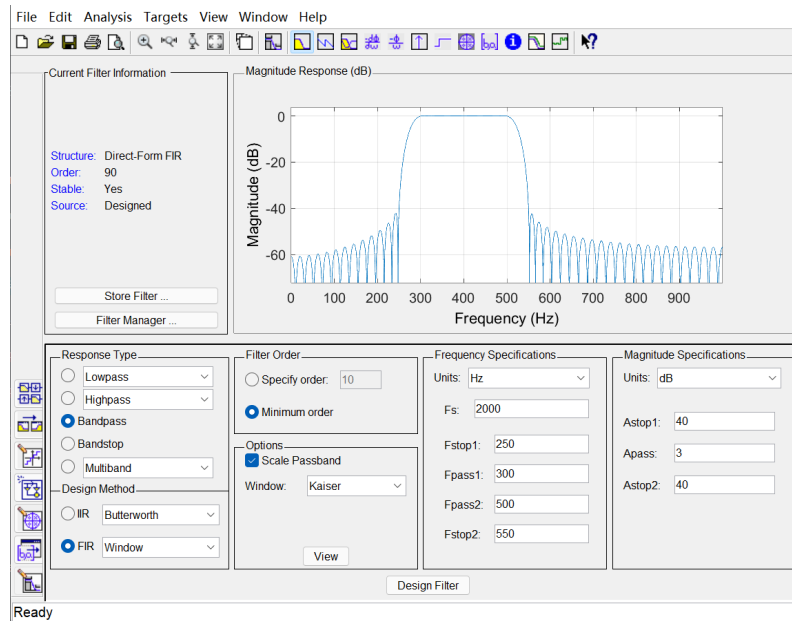


图 10: MATLAB 提供的 Filter Designer 滤波器设计工具

利用 Filter Designer 工具还可以生成 Matlab 代码: 打开 File 菜单栏, 点击 Generate MATLAB Code, 生成的代码如下:

```
1 function Hd = firgen
2 %FIRGEN Returns a discrete-time filter object.
3
4 % MATLAB Code
5 % Generated by MATLAB(R) 9.10 and DSP System Toolbox 9.12.
6 % Generated on: 22-Dec-2021 13:13:40
7
8 % FIR Window Bandpass filter designed using the FIR1 function.
9
10 % All frequency values are in Hz.
11 Fs = 2000; % Sampling Frequency
12
13 Fstop1 = 250; % First Stopband Frequency
14 Fpass1 = 300; % First Passband Frequency
15 Fpass2 = 500; % Second Passband Frequency
16 Fstop2 = 550; % Second Stopband Frequency
17 Dstop1 = 0.01; % First Stopband Attenuation
18 Dpass = 0.17099735734; % Passband Ripple
19 Dstop2 = 0.01; % Second Stopband Attenuation
20 flag = 'scale'; % Sampling Flag
```



```
21
22 % Calculate the order from the parameters using KAISERORD.
23 [N,Wn,BETA,TYPE] = kaiserord([Fstop1 Fpass1 Fpass2 Fstop2]/(Fs/2), [0 ...
24 1 0], [Dstop1 Dpass Dstop2]);
25
26 % Calculate the coefficients using the FIR1 function.
27 b = fir1(N, Wn, TYPE, kaiser(N+1, BETA), flag);
28 Hd = dfilt.dfir(b);
29
30 % [EOF]
```

5 基于 MATLAB 的数字滤波器的硬件实现

数字信号处理 (Digital Signal Processing, DSP) 可在通用计算机上用软件实现, 但速度较慢, 一般用于 DSP 算法的模拟. 在对速度要求高的场合, DSP 应当用硬件实现. 本节探索部署于 ASIC/FPGA 的数字滤波硬件实现方案.

5.1 用 HDL Coder 工具生成 FIR 滤波器的 HDL 代码

MATLAB 提供了 Filter Design HDL Coder 工具, 可以生成定点滤波器的可综合 HDL (Hardware Description Language, 硬件描述语言) 代码. 此外, 还可以生成用于验证的 Test Bench. 在 Filter Designer 中设计好 FIR 滤波器后 (见第4.4 节), 选择菜单栏中的 Targets -> Generate HDL ... 即可打开 Filter Design HDL Coder, 如图 (11) 所示.

在 Filter Design HDL Coder 中可以设置各种参数, 例如架构 (并行, 半并行, 串行), 复位类型 (同步, 异步), 字长, 加法结构 (加法链, 加法树), 流水线级数, 生成的代码的命名规则, Test Bench 激励类型 (冲激, 阶跃, 斜坡, 扫频) 等.

将 Filter Design HDL Coder 导出的 Verilog 设计和仿真源文件 (filter.v, filter_tb.v) 导入 Modelsim 或 Vivado 进行仿真, 扫频部分的仿真结果如图 (12) 所示. 可见在扫频信号激励下, FIR 带通滤波器可以滤除低频和高频信号.

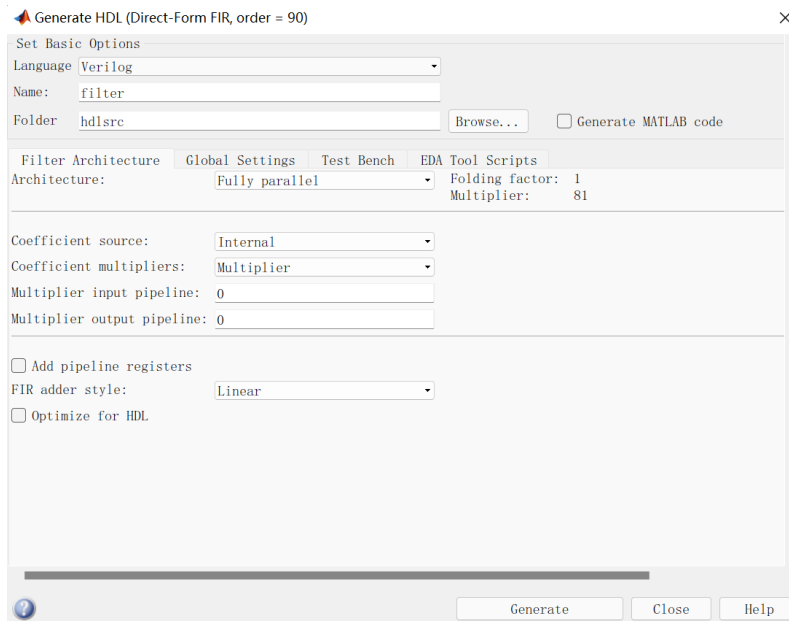
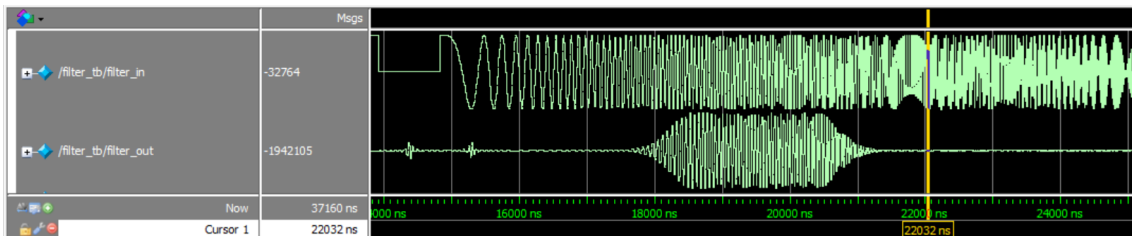
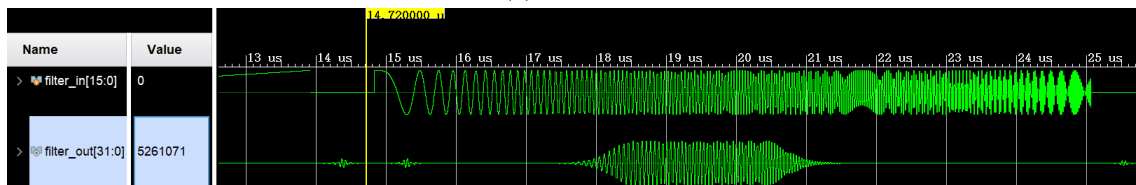


图 11: Filter Design HDL Coder



(a) Modelsim



(b) Vivado Simulator

图 12: FIR 滤波器仿真结果