

基于 MATLAB 窗函数法设计 FIR 滤波器

蒋梦影 占伟星 陶敬荣

浙江八达电子仪表有限公司, 浙江 金华 321018

[摘要] FIR 滤波器的应用十分普遍, 主要是因为它的输出可以保持良好的线性特征。而传统应用中的数字滤波器由于复杂的设计过程、需要辅以大量的计算并且调整其滤波特性极其困难等原因相比较而言就没得到那么广泛的应用。在实际应用中, 信号是多变的, 信号的处理既要有极强的实时性又要兼具较高的灵活性, 而 MATLAB 的出现恰好解决了这个难题。随着 MATLAB 技术的发展, MATLAB 软件包功能强大、使用方便, 用 MATLAB 来实现 FIR 滤波器, 既具有实时性, 又兼顾了一定的灵活性。利用 MATLAB 设计数字滤波器, 有利于滤波器设计的优化, 能为硬件实现提供数据基础。

[关键词] FIR 数字滤波器; MATLAB; 窗函数法

DOI: 10.33142/sca.v5i4.6706

中图分类号: TN911

文献标识码: A

Design FIR Filter Based on MATLAB Window Function Method

JIANG Mengying, ZHAN Weixing, TAO Jingrong

Zhejiang Bada Electronic Instrument Co., Ltd., Jinhua, Zhejiang, 321018, China

Abstract: FIR filter is widely used, mainly because its output can maintain good linear characteristics. However, digital filters in traditional applications have not been widely used because of the complex design process, the need to be supplemented by a large number of calculations and the extremely difficult to adjust their filtering characteristics. In practical application, the signal is changeable, and the signal processing should have both strong real-time and high flexibility, and the emergence of MATLAB just solves this problem. With the development of MATLAB technology, MATLAB software package is powerful and easy to use. Using MATLAB to realize FIR filter has both real-time and flexibility. Using MATLAB to design digital filter is conducive to the optimization of filter design, and can provide data basis for hardware implementation.

Keywords: FIR digital filter; MATLAB; window function method

1 本课题的研究意义

信号的处理问题在许多领域里都有应用, 其表现形式为电、磁、机械以及热、光、声等各种形式。在应用工程如何在强噪声背景中提取不失真的真实信号或信号特征是极为关键的。信息是由许多不同的信号来传达, 而不同的信号所承载的讯息也是不一样的。但是在信号传递过程中可能会受到外界干扰而产生变化, 因此, 我们要从这些信号中, 提取出我们要的信号, 也就是说, 我们要做的, 就是过滤掉那些不必要的噪音干扰, 从而获得我们要的信号。滤波器需要完成的使命也就在此。滤波器的作用就是去除那些我们不需要的信号, 划分频带、让信号频谱成形。

2 FIR 数字滤波器基础

数字滤波器实现的功能是对输入的信号进行处理, 也就是把我们不需要的信号尽可能都除掉。离散时间非线性时不变系统输出数据是基于输入数据经过精密计算的而得出的, 非常精确, 所以数字滤波器精度很高稳定性很好灵活性很大。由于社会生产力的发展日新月异, 工作节奏越来越快, 大家对滤波器的设计方法提出了需要更快更便捷的要求。

2.1 数字滤波器的原理

数字滤波器的系统函数可以用式 (2.1) 表示:

$$H(z) = \frac{\sum_{k=0}^m b_k z^{-k}}{1 - \sum_{k=1}^n a_k z^{-k}} = \frac{Y(z)}{X(z)} \quad (1)$$

变换后得出表示输入输出关系的方程为:

$$y(n) = \sum_{k=1}^n a_k y(n-k) + \sum_{k=0}^m b_k x(n-k) \quad (2)$$

通过这两个式子可以看出, 数字滤波器是通过 (2) 式将输入序列变换成输出序列的, 又因为 FIR 数字滤波器的系统函数仅有零点, 而且因为系数为零, 代入式 (2) 可得:

$$y(n) = \sum_{k=0}^m b_k x(n-k) \quad (3)$$

用脉冲响应序列 $h(n)$ ($n=0, 1, \dots$) 来表示数字滤波器, 如图 1 所示^[2]。

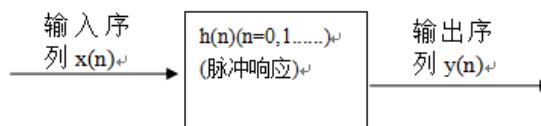


图 1 数字滤波器的图示

因为数字滤波器的输入、输出以及单位脉冲响应 $h(n)$

都是离散的，所以在线性时不变系统中，数字滤波器的时域输入输出关系是：

$$y(n) = x(n) * h(n) \quad (4)$$

输出的频域关系是(x(n), y(n)的傅立叶变换都存在)：

$$Y(e^{j\omega}) = X(e^{j\omega})H(e^{j\omega}) \quad (5)$$

根据输入信号可通过卷积和计算输出信号，FIR 滤波器的相关公式如式(6)所示。

$$y(n) = \sum_{k=0}^{K-1} h(k)x(n-k) \quad (6)$$

其中：

$x(n-k)$ ：延时 k 个抽头的输入信号。

K ：FIR 滤波器的抽头数；

$h(k)$ ：第 k 级的滤波器系数；

由于 FIR 滤波器可以在保证精确严格的线性相位特性的情况下设计任意幅度频率特性滤波器，而且 FIR 滤波器可以设计成因果稳定的系统，所以 FIR 滤波器比 IIR 滤波器应用更广泛。

3 基于 MATLAB 的窗函数法设计 FIR 滤波器

设计数字滤波器主要通过以下步骤来实现：1. 输入输出指标的确定；2 通过计算尽可能地逼近理想值 3. 选用相应的函数来实现。通常在设计滤波器之前，应该先根据具体的应用确定滤波器的性能指标。然后就可以根据公式算出相应的值，最后可以根据这个函数来进行实现，滤波器就这样设计完成了。

FIR 滤波器的设计主要是基于对理想中的滤波器的无限逼近。由于窗函数法在应用中简单、物理意义明确，因此被采用的比较多。

3.1 窗函数设计法

FIR 滤波器设计中最简单快捷的方法就是采用窗函数法^[3]。因为窗函数法是基于时域设计的，所以需要通过对傅里叶逆变换将 FIR 滤波器的理想频率响应 $H_d(e^{j\omega})$ 从频域转换成时域——它的单位冲击响应 $h_d(n)$ ，然后再用 FIR 滤波器的实际单位冲激响应 $h(n)$ 去无限逼近这个理论值。假设理想的滤波器的频率响应 $H_d(e^{j\omega})$ 的表示为：

$$H_d(e^{j\omega}) = \sum_{n=-\infty}^{\infty} h_d(n)e^{-j\omega n} \quad (7)$$

通过傅里叶逆变换后，滤波器的频率响应可以得到相应的单位冲激响应序列：

$$h_d(n) = \frac{1}{2\pi} \int_{-\pi}^{\pi} H_d(e^{j\omega})e^{j\omega n} d\omega \quad (8)$$

经过上述计算得到 $h_d(n)$ 后，再将 $h_d(n)$ 通过傅里叶变换来获得即理想滤波器（离散）系统传递函数：

$$H_d(z) = \sum_{n=-\infty}^{\infty} h_d(n)z^{-n} \quad (9)$$

因为 $h_d(n)$ 为无限长的序列，因此在实际运算设计中 $H_d(z)$ 也是不可能达到的，为了使设计可以实现的，并且

效果达到最好的状态，我们需要使实际的 FIR 滤波器频率响应无限贴近理想滤波器的频率响应，换句话说就是两者之间不一样的地方越少越好，可以采用合适的窗函数将无限脉冲响应截取一段来近似表示 $h_d(n)$ ，可得

$$h(n) = h_d(n)w(n) \quad (10)$$

运用这种简单的截短方法会出现一个新的问题，它会使逼近的频率响应具有一定百分量的尖冲振幅和波动波纹，这就是吉布斯(Gibbs)现象。加窗来修正是解决这一现象的一个最有效的方法，现实中的 FIR 滤波器 $H(\omega)$ 只受窗口的振幅频率特征的影响。FIR 滤波器在实际应用中的振幅频率特征是由理想中的低通滤波器和我们所用的窗函数振幅频率特征组成的复卷积。由于复卷积会使 $H(\omega)$ 产生过冲和起伏，因此，在加入窗函数之后，其理想性能会受到如下的影响^[1]：

(1) 过渡区的带宽宽度与窗函数的主瓣宽度相等。因此，窗函数的主瓣宽度与过渡区宽度呈正比关系。

(2) 窗函数的副瓣会引起滤波器振幅频率特性的变化，副瓣的相对振幅与于其影响产生的波动的振幅呈正比例关系。而副瓣的值又会影响到产生的波动性的大小。

(3) 如若增大窗函数的长度，也仅能使窗函数的振幅频率特征 $W(\omega)$ 的主瓣的宽度减少，而不能使主瓣与副瓣之间的大小比值产生变化，这个值是由窗口函数的形状所决定，也就是说，加大截取函数的长度 N 仅能使过渡带减少一定比例的值，而不会影响滤波器的波动性。

我们可以选择不同的窗函数来满足各种实际应用中的要求，而窗函数的选择原则是^[2]：

旁瓣的振幅幅度选择较小的。

旁瓣的振幅降低的越快越好，从而使阻带的衰减相应增大，使滤波效果更好。

将主瓣的宽度缩小，使过渡区带宽变得更窄。

设计思想：我们从时域着手，使我们设计的 $h(n)$ 接近于理想中的 $h_d(n)$ 。将理想滤波器 $H_d(e^{j\omega})$ 的单位脉冲响应设定为 $h_d(n)$ ^[4]。如果是低通线性特性的 FIR 数字滤波器，那么：

$$H_d(e^{j\omega}) = \sum_{n=-\infty}^{\infty} h_d(n)e^{-j\omega n} \quad (11)$$

$$h_d(n) = \frac{1}{2\pi} \int_{-\pi}^{\pi} H_d(e^{j\omega})e^{j\omega n} d\omega \quad (12)$$

窗函数设计法的原理可以用图 2 的框图表示^[5]：

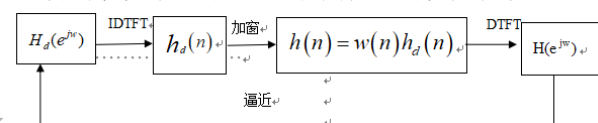


图 2 窗函数设计法原理

$h_d(n)$ 一般是无限长非因果序列，在设计中不能实现所以不能直接把它作为 FIR 滤波器的单位脉冲响应。用公式 $h(n) = h_d(n)w(n)$ 截断 $h_d(n)$ 就可以得到一个因果的有限

长的滤波器系数 $h(n)$ ，并用最符合设计条件的窗函数经过计算作为 FIR 滤波器的单位脉冲响应。用 $h(n)$ 去无限逼近 $h_d(n)$ 而得到的 FIR 滤波器，那么他们俩的频响 $H(e^{j\omega})$ 和理想频响 $H_d(e^{j\omega})$ 也肯定是类似的。所以窗口设计就是加窗去截断 $h_d(n)$ 的过程。

3.2 窗函数的几种类型及相应的选取得原理

MATLAB 中用窗函数法设计 FIR 滤波器的比较常用的窗函数有矩形窗、汉宁窗、海明窗等。不同的窗函数适用于参数不同的滤波器的设计中。

在 MATLAB 中，实现矩形窗的函数为 `boxcar` 和 `rectwn`，其调用格式分别如下： $w = \text{boxcar}(N)$ ， $w = \text{rectwin}(N)$ ，在上述式子中， N 是我们所选用的窗函数的长度，通过函数计算得到的 w 是一个 N 阶的矢量，其要素是窗函数的数值。 $w = \text{boxcar}(N)$ 等价于 $w = \text{ones}(N,1)$ 。

与矩形窗相比较，三角窗函数的的主瓣宽度 ($8\pi/N$) 是矩形窗的主瓣宽度的两倍，但其旁瓣却比矩形窗的小很多。在 matlab 中，调用 $w = \text{triang}(N)$ 来实现，三角窗函数与巴特里特函数的窗函数非常相像，一般情况下巴特里里窗函数的起始和结尾一般为 0，而三角窗函数在正常情况下是非零的，只有在窗函数的长度值 N 为偶数时， $\text{triang}(N-2)$ 等价于 $\text{bartlett}(N)$ 。

Hanning 窗口的最大副瓣比主瓣值要小 31 db，而主瓣的宽度则是矩形窗函数的的主瓣宽度的两倍大。在 MATLAB 中，调用 $w = \text{hann}(N)$ ， $w = \text{hann}(N, 'sflag')$ 来实现汉宁窗，`hann` 函数中的参数 `sflag` 为采样方式，其值可取 `symmetric` (默认值) 或 `periodic`。当 `sflag = symmetric` 时，为对称采样；当 `sflag = periodic`，为周期采样，此时 `hann` 函数计算 $N+1$ 个点的窗，但是仅返回前 N 个点。

与汉宁窗函数进行比较，海明窗函数与汉宁窗函数的的主瓣宽度是一致的，旁瓣比汉宁窗函数的要小，最大旁瓣的值低于主瓣的值 41db，海明窗的函数的实现使用 `hamming`，其调用格式如下： $w = \text{hamming}(N)$ ， $w = \text{hamming}(N, 'sflag')$ ，函数说明和矩形窗的一样。

当然我们也可以直接用直接法设计滤波器，用这个方法设计通常可用的函数主要有三种：`fir1`，`fir2` 和 `kaiserord`。`fir1` 函数是一种可用于标准带通滤波器的设计，是一种非常典型的 FIR 滤波器的设计函数，`fir1` 函数的具体算法是：如果窗函数为 $w(n)$ ，理想滤波器的单位脉冲响应为 $h(n)$ ，则所设计的滤波器的系数为 $b(n) = w(n)h(n)$ ，其中 $1 \leq n \leq N$ 。 $b = \text{fir1}(N, Wn)$ 中的 b ，是由 $N+1$ 个元素构成的向量，这些向量是由滤波器传递函数的 $N+1$ 个系数所构成的，参数 N 为滤波器的阶次， Wn 为归一化的截止频率。

我设计了低通的 fir 滤波器和高通的 fir 滤波器，设计参数如下表 1 所示：

首先，我采样窗函数法设计 fir 低通滤波器，根据最小阻带衰减的特性，以及表 1，得出只有海明窗和布莱克

曼窗可提供大于 50db 的衰减，我选择海明窗，其过渡带为 $6.6\pi/N$ ，具有较小的阶次。调用 `ideal_lp(wc, N)` 计算理想低通滤波器的单位冲激响应，加窗后我再调用一个子函数 `[db, mag, pha, w] = \text{freqz}_m2(h, [1])` 来计算现实中滤波器的幅值响应。

表 1 设计参数

滤波器类型	通带截止频率	阻带截止频率	通带波纹	阻带衰减
低通滤波器	0.2π	0.3π	0.25db	50db
高通滤波器	0.8π	0.7π	0.25db	50db

最后画出相应的图进行比较。

图 3 图 4 是窗函数法设计的仿真图，图 3 采用的是海明窗，图 4 采用的是凯塞窗，由图可以看出，不管是海明窗或者凯塞窗设计都满足 $w_p = 0.2\pi$ ， $w_s = 0.3\pi$ ， $A_s = 50\text{db}$ 。

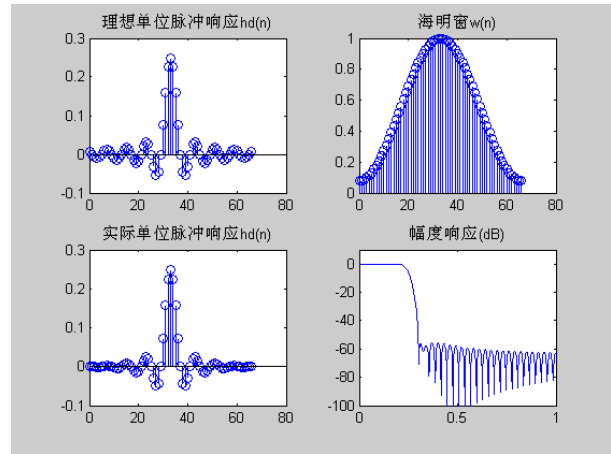


图 3 窗函数设计法(海明窗)

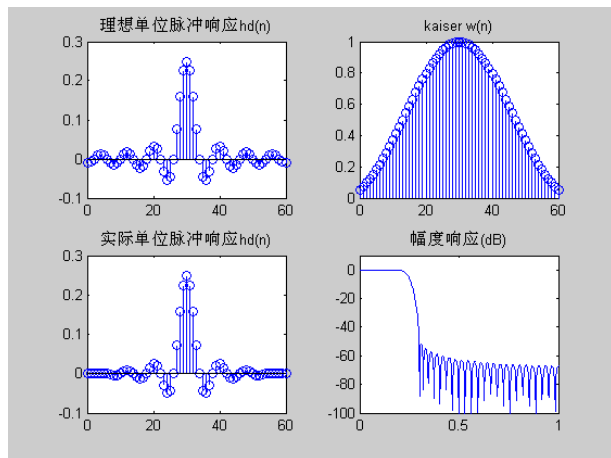


图 4 窗函数设计法(凯塞窗)

由于高通滤波器通带截止频率 w_p 为 0.8π ，阻带的截止频率 w_s 为 0.7π ，阻带损耗 A_s 等于 50db，通带的波纹 A_p 等于 0.25db，查表可知汉宁窗最符合要求，因此这个设计我选用汉宁窗来实现，然后根据已选定的窗函数求出滤波器的阶次，用 `hd = \text{ideal}_hpl(wc, N)` 求出一个理想

中的高通滤波器的单位冲激响应，并调用 $[db, mag, pha, w]=freqz_m2(h, [1])$ 来求得实际滤波器的幅值响应。

图 5 是采用基于窗函数设计中的汉宁窗来设计实现的滤波器，从图片中可得滤波器的通带阻带截止频率及阻带的损耗都已满足要求。

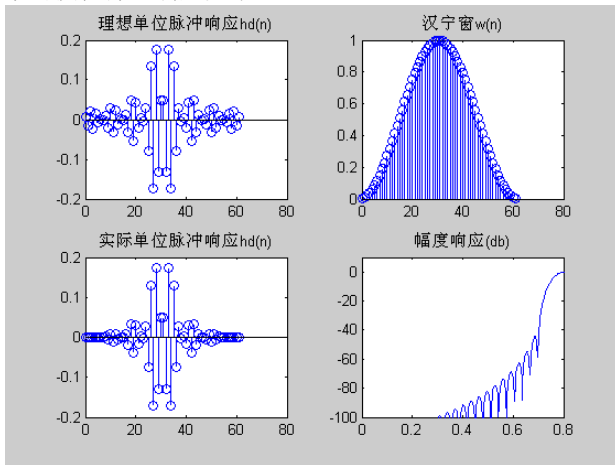


图 5 窗函数法设计法(汉宁窗)

4 总结

运用 MATLAB 窗函数法设计 FIR 滤波器比较快捷方便，生成的 FIR 滤波器具有严格的线性相位，而且比较容易进行修改，并且所生成的滤波器系数还可以成为硬件设计的理论依据。通过设计仿真结果可知：窗函数的设计方法相

比较而言是很简单的，只需要选取适当的窗函数即可，但由于其不够灵活，很容易导致通带波纹过小或阻带衰减过大等问题。此外，由于窗口函数频谱与所需响应间存在卷积作用的问题，使其无法准确地确定通带边界频率和阻带边界频率的值。

【参考文献】

- [1]梁辰. 基于 MATLAB 的 FIR 数字滤波器的设计[J]. 机械设计与制造, 2010(12): 87-89.
 - [2]卢莉蓉, 周晋阳, 牛晓东. 基于 MATLAB 的 FIR 数字滤波器的设计与仿真[J]. 科教前沿, 2010(13): 25-27.
 - [3]朱敏. MATLAB 数字信号处理工具箱的开发和应用——数字滤波器 FIR 的设计[J]. 信息与电脑, 2010(2): 154-156.
 - [4]周伟林, 周鲜成. FIR 滤波器的软件仿真与硬件实现[J]. 微计算机信息, 2009(25): 222-225.
 - [5]李明, 罗霖, 汤胜龙. 基于窗函数的 FIR 数字滤波器的优化及 Matlab 实现[J]. 应用技术与研究学术讨论, 2009(10): 61-65.
 - [6]张葵, 刘伟. 窗函数法设计 FIR 带通滤波器的实现与性能研究[J]. 中国科技信息, 2010(15): 142-145.
 - [7]飞思科技产品研发中心. MATLAB7 辅助信号处理技术与应用[M]. 北京: 电子工业出版社, 2005.
- 作者简介: 蒋梦影(1989-)女, 本科, 中国计量学院现代科技学院, 通信工程, 浙江八达电子仪表有限公司, 工程师, 从事智能电能表开发测试工作。