

CZT 和 ZFFT 频谱细化性能分析及 FPGA 实现

马可, 张远安, 张开生

(西安电子工程研究所, 西安 710100)

摘要: 快速傅里叶变换在信号的频谱分析中应用广泛, 而在工程实际中往往只对信号频谱中一段区间感兴趣, 需要对频谱进行细化分析; 常用的频谱细化方法有线性调频 Z 变换 (Chirp-Z Transform, CZT) 算法和基于复调制的细化 (Zoom-FFT, ZFFT) 算法, 在给出了两种频谱细化方法的计算流程后, 通过 MATLAB 仿真说明两种算法都能提高信号频率的测量精度, 最后给出了这两种算法在 FPGA 具体工程实现中的实现步骤。

关键词: 线性调频 Z 变换; 复调制细化; 频谱细化

Performance Analysis for CZT and ZFFT Spectrum Zoom and Its FPGA Realization

Ma Ke, Zhang Yuanan, Zhang Kaisheng

(Xi'an Electronic Engineering Research Institute, Xi'an 710100, China)

Abstract: Fast Fourier Transform is widely used in the signal spectrum analysis. However, it usually gets interesting in a part of the whole signal spectrum in engineering, so it needs spectrum zoom analysis. It is the common method for spectrum zoom which including complex modulation Zoom-FFT (ZFFT) and Chirp-Z Transform (CZT). After the calculating methods of two kinds spectrum zoom are given, both ZFFT and CZT can improve the measurement accuracy by simulation in MATLAB. Lastly, this paper gives the realization steps of the two methods in FPGA engineering.

Keywords: Chirp-Z transform; zoom-FFT; spectrum zoom

0 引言

在信号处理中, 经常需要分析信号的频谱, 传统的频谱分析方法一般采用快速傅里叶变换 (fast fourier transform, FFT) 算法, FFT 算法得到的是整个采样频率上的粗略的“全景频谱”, 而在实际的应用中, 往往只需要对感兴趣的某一段频谱区间进行细微观察和分析, 这就需要通过提高所选区间的频率分辨率^[1]。

频率分辨率表示频谱中能够分辨的两个频率分量的最小间隔^[2], 用频率间隔 Δf 来表示。而频率间隔等于采用频率 f_s 除以采样点数 N 。为了提高频率分辨率, 可以采用的方法: 一是降低采样频率, 这会使频率分析范围缩小, 并且采样频率还受采样定律限制, 不可能太小; 二是增加分析的采样点数, 这意味着计算机的存储量和计算量大大增加, 由于实际系统软件、硬件方面的限制, 这样做在工程实践中并不总是可能的。

为了对窄带频谱区间进行细致观察, 提出了频谱细化的概念, 其基本思路就是对信号频谱中需要分析的频段进行局部放大, 也即在某一频率附近局部增加谱线密度, 实现选带分析。常用的频谱细化方法有线性调频 Z 变换 (Chirp-Z Transform, CZT) 算法和基于复调制的细化 (Zoom-FFT, ZFFT) 算法。文献 [3-6] 中都介绍了这两种算法的原理和性能, 不过对于 ZFFT 算法的具体现场可编程门阵列 (field program-

mable gate array, FPGA) 实现, 并没有太多介绍。本文首先简单介绍这两种算法原理并对比分析它们的性能, 然后给出了这两种算法在 FPGA 中的具体实现步骤。

1 算法分析

1.1 CZT 算法

CZT 可以对感兴趣的某一段频谱区间进行细化, 它通过对单位圆上任意一段曲线上的 Z 变换进行计算。对有限长序列 $x(n)$ 做 CZT, 主要是确定需要细化频谱区间的起始频率、细化频率间隔以及取样点数, 其线性滤波计算步骤如图 1 所示^[7-8]。

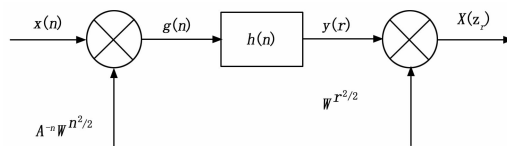


图 1 CZT 的线性滤波计算步骤

对于长度为 M 的有限长序列 $x(n)$ 的 CZT 是:

$$X(z_r) = CZT[x(n)] = \sum_{n=0}^{M-1} x(n) A_0^{-n} e^{-j\theta_0 n} W_0^n e^{-j\varphi_0 nr} \quad (1)$$

其中: θ_0 为初始幅角, φ_0 为单位圆等间隔增量。当 $A_0 = W_0 = 1$ 时, 即在单位圆上进行采样时, 式 (2) 变为:

$$X(r) = \sum_{n=0}^{M-1} x(n) \exp[-j(\theta_0 + \varphi_0 r)n] \quad (2)$$

式中, $\theta_0 = 2\pi f_0 / f_s$, $\varphi_0 = 2\pi f_L / (M f_s)$, 其中 f_0 为分析频谱的起始频率, f_L 为分析频谱的长度, f_s 为采样频率。这样,

收稿日期: 2015-09-08; 修回日期: 2015-09-29。

基金项目: 武器装备预先研究基金资助项目 (40405040201)。

作者简介: 马可 (1988-) 男, 陕西西安人, 工程师, 主要从事雷达总体工程与雷达信号处理方向的研究。

CZT 算法频率分辨率为 $\Delta f = f_L / M$ 。

1.2 ZFFT 算法

ZFFT 算法通过指定的采样频率分析整个频谱上感兴趣的一段区间。对有限长序列 $x(n)$ 进行 ZFFT, 主要需要确定需要细化频带的频率中心 f_c , 细化倍数 D , 其主要的运算流程如图 2 所示^[9]。

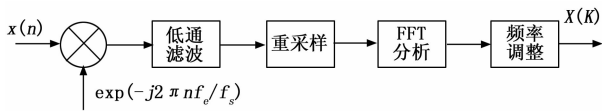


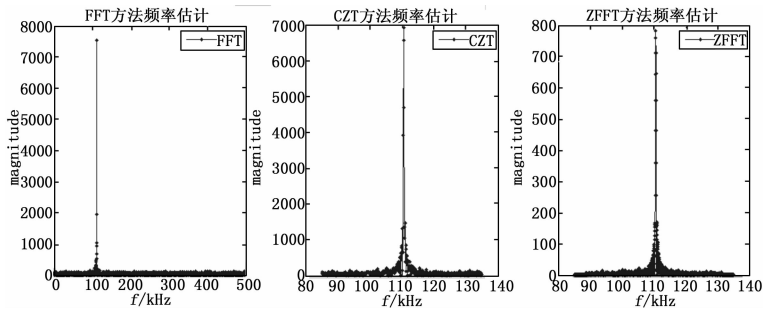
图 2 ZFFT 运算流程

在确定了细化频带的中心频率 f_c 、细化倍数 D 以及采样频率 f_s 之后, 就可以确定细化频带的下限频率 $f_l = f_c - f_s / (2D)$ 以及上限频率 $f_h = f_c + f_s / (2D)$ 。重采样之后如果进行 N 点 FFT 处理, 则频率分辨率为 $\Delta f = f_s / (DN)$ 。FFT 处理之后, 对结果进行频率翻转, 得到最后的处理结果。

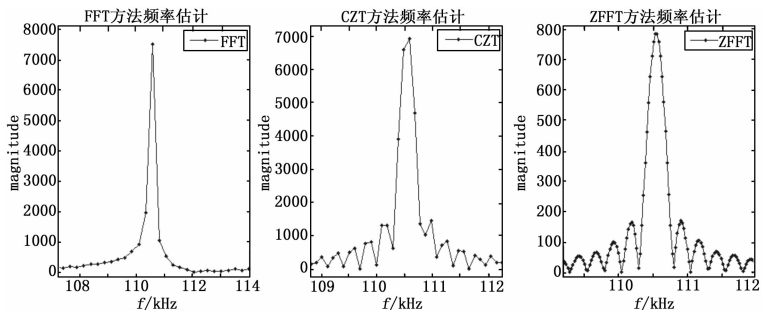
2 CZT 和 ZFFT 频谱仿真

针对上述 CZT 算法和 ZFFT 算法, 我们进行计算机仿真。在仿真分析 CZT 和 ZFFT 算法对信号频谱的细化情况时, 先分析单频率成分的测量精度, 再分析多频率成分的测量精度。

本文仿真中, 假定信号频率 f 为 110.55 kHz, 采样频率 f_s 为 500 kHz, 信号幅度 $A=4$, 噪声为服从正态分布的随机噪声, 方差为 1.6, 信噪比 SNR 为 10 dB。用 CZT 算法进行分析时, 细化的频率区间也设定为 85 kHz~135 kHz, 细化点数为 512。用 ZFFT 算法分析时, 细化频谱区间的中心频率 f_c 为 110 kHz, 细化倍数 $D=10$, 则细化的频率区间也为 85 kHz~135 kHz。用功率加权求平均的方法进行频谱校正, 做 100 次 Monte Carlo 实验得到仿真结果如图 3 所示。



(a) 3 种算法频谱分析图



(b) 3 种算法频谱分析局部放大图

图 3 仿真结果图

由图 3 可以得出, FFT 方法仿真估计出的结果 $f_{FFT} = 110.60$ kHz, 误差 $\sigma_{FFT} = 0.05$ kHz; CZT 方法仿真估计出的结果 $f_{CZT} = 110.59$ kHz, 误差 $\sigma_{CZT} = 0.04$ kHz; ZFFT 方法仿真估计出的结果 $f_{ZFFT} = 110.56$ kHz, 误差 $\sigma_{ZFFT} = 0.01$ kHz。从图 3 的仿真结果可以看出, CZT 算法和 ZFFT 算法比传统的 FFT 算法在频谱测量精度上, 要提高很多。

除了分析单频信号, 这里假设一个多频率成分的信号, 频率分别为 110.16 kHz, 110.55 kHz, 110.93 kHz, 幅度分别为 3, 5, 4, 其他参数同上面仿真设计, 其频谱细化的仿真结果如图 4 所示。

从图 4 中可以看到, 在多频率成分时, 当信号的多个频率成分很接近, FFT 算法无法区分信号的 3 个频率, 而 CZT 以及 ZFFT 算法可以区分 3 个频率成分的谱峰, 具体的测量结果

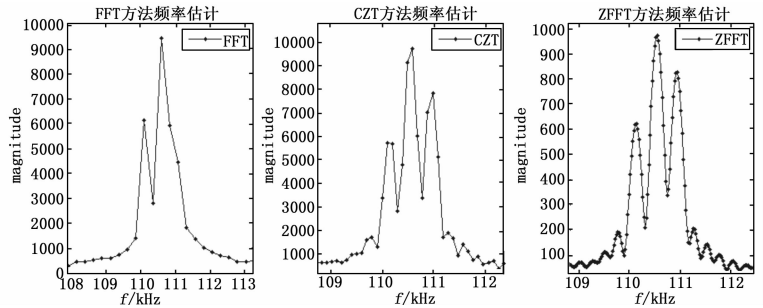


图 4 多频率下的仿真结果

如下表 1 所示。

表 1 3 种算法频谱谱峰测量值

	理论频率 /kHz	FFT		CZT		ZFFT	
		测得值	误差	测得值	误差	测得值	误差
频率成分 1	110.16	110.11	0.05	110.10	0.06	110.15	0.01
频率成分 2	110.55	110.60	0.05	110.59	0.04	110.56	0.01
频率成分 3	110.93			110.98	0.05	110.95	0.02

从仿真中可以看到, ZFFT 算法在频谱细化中, 比 FFT 算法和 CZT 算法精度要高很多。从频率分辨率上分析, FFT 算法的频率分辨率 $\Delta f_{FFT} = f_s / N = 0.24$ kHz, CZT 算法的频率分辨率 $\Delta f_{CZT} = 50 / M = 0.10$ kHz, ZFFT 算法的频率分辨率 $\Delta f_{ZFFT} = f_s / (ND) = 0.02$ kHz。所以从这一点上也可以看出 ZFFT 算法的频率分辨率要高些。

3 CZT 和 ZFFT 的 FPGA 实现

图 1 中介绍了 CZT 算法的流程, 其中比较复杂的计算步骤是 $y(n)$ 和 $h(n)$ 的卷积, 在 FPGA 中具体实现 CZT 算法时, 常采用图 5 中的计算过程。

实现时, 首先利用 MATLAB 产生图 5 中三组系数的 coe 文件, 通过设定所需分析频谱的起始频率和分析频段的长度, 细化点数, 产生三组浮点型的系数, 换算成 16 bit 的整型有符号数, 存入 Xilinx 的 Block RAM IP 核中。运算时, 通过设计 FPGA 内部程序, 使得系数 1 读出时, 与输入信号 $x(n)$ 对齐, 调用复数乘法器 IP 核进行运算, 对输出结果 $y'(n)$ 调用 FFT

(下转第 303 页)