

分裂基算法的讨论

何 璞, 张 平

(西安电子科技大学 电子工程学院 陕西 西安 710071)

摘 要: 介绍了分裂基 FFT 的原理及其在 Matlab 上的实现方法, 并将基 2 算法和分裂基算法进行了性能比较。

关键词: FFT; 分裂基算法; L 型蝶形; Matlab

中图分类号: TP301.6

文献标识码: B

文章编号: 1004-373X (2003) 18-031-03

Discussion on Split-radix FFT

HE Pu, ZHANG Ping

(Department of Electronics Engineering, Xidian University, Xi'an, 710071, China)

Abstract: This article introduces the principle of split-radix FFT and implements it on matlab. A comparison of split-radix fft with radix-2 FFT is also made.

Keywords: FFT; split-radix FFT; L-shaped butterfly; Matlab

1 引 言

快速傅里叶变化 FFT 是离散傅里叶变换的一种快速算法。自 1965 年 Cooley-Tukey 的算法提出之后, 新的算法不断涌现。总的说来, 这几十年的快速傅里叶变换的发展方向有 2 个; 一是针对 N 等于 2 的整数次幂的算法, 如基 2 算法、基 4 算法、实因子算法和分裂基算法等; 另一个是 N 不等于 2 的整数次幂的算法, 他是以 Winograd 为代表的一类算法 (素因子算法, Winograd 算法)。1984 年提出的分裂基算法 (Split-radix FFT) 同时使用基 2 和基 4 算法, 被认为是目前对 N 等于 2 的整数次幂中各类算法中最为理想的一种。Winograd 算法 (WFTA) 和上述算法在理论上有着根本的差别, 他是建立在下标映射和数论上的一套完全新颖的算法。

本文将重点介绍分裂基算法的原理和实现方法及其性能的改善, 实验结果表明该算法与常用的基 2 算法相比, 速度有明显的改善。

2 分裂基算法

分裂基算法^[1]又称基 2/4 算法。众所周知, 基 4 算法比基 2 算法更有效, 从原理上讲, 用较大的基还可以进一步减少复数乘法次数, 但他是以增加复数加法次数和以程序 (或硬件) 变得更为复杂为代价的, 所

以取大于 8 的基数是没有实际意义的。分裂基算法的基本思路就是对偶序号输出使用基 2 算法, 对奇序号输出使用基 4 算法, 将基 2 分解和基 4 分解组合在一起。分裂基算法在目前已知的所有针对的算法中具有最少的乘法次数和加法次数, 并且具有好的结构, 运算程序也较短, 因此被认为是一种实用的高效算法。后来的研究表明, 该算法最接近理论上所需乘法次数的最小值^[2]。

算法的推导:

对 $N = 2^M$ 点 DFT:

$$X(k) = \sum_{n=0}^{N-1} x(n) W_N^{kn}, \\ 0 \leq k \leq N-1$$

DIF 的偶序号输出项, 即:

$$X(2r) = \sum_{n=0}^{\frac{N}{2}-1} [x(n) + x(n + \frac{N}{2})] W_N^{nr}, \\ r = 0, 1, \dots, \frac{N}{2} - 1 \quad (1)$$

对 k 的奇序号项用基 4 算法, 即:

$$X(4r+1) = \sum_{n=0}^{\frac{N}{4}-1} [(x(n) - x(n + \frac{N}{2})) - \\ j(x(n + \frac{N}{4}) - x(n + 3\frac{N}{4}))] W_N^r W_{N/4}^{nr} \quad (2)$$

$$j(x(n + \frac{N}{4}) - x(n + 3\frac{N}{4}))] W_N^{3n} W_{N/4}^n, \quad r = 0, 1, \dots, N/4 - 1 \quad (3)$$

式(3)构成了分裂基算法的L型算法结构,如图1所示。

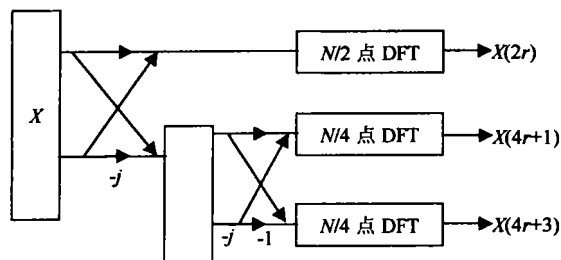


图1 L型算法结构

从图中可以看出一个 N 点序列 $x(n)$ 的 N 点DFT可以分解成1个 $N/2$ 点DFT和2个 $N/4$ 点DFT。这种分解既有基2部分,又有基4部分。基2部分 $X(2r)$ 的奇数点部分又进一步按基4抽取分解,而基4部分的偶数点部分又进一步按基2抽取分解。

第一次分解完成后,所得 $N/2$ 点DFT又可用 $N/8$ 个L型蝶形运算分解成1个 $N/4$ 点DFT和2个 $N/8$ 点DFT,所得2个 $N/4$ 点DFT也可以分别用 $N/16$ 个L型蝶形运算分解成一个 $N/8$ 点DFT和2个 $N/16$ 点DFT。以此类推的进行下去,直至分解为4点或2点DFT,完成序列 $x(n)$ 的 N 点DFT的快速运算。

现以 $N=8$ 为例,推导其算法,并给出信号流程图如图2所示。

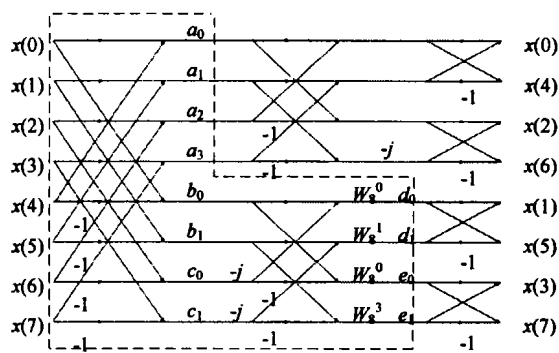


图2 8点分裂基算法信号流程图

令:

$$a(n) = x(n) + x(n+4), \quad n = 0, 1, 2, 3$$

$$b(n) = x(n) - x(n+4), \quad n = 0, 1,$$

$$c(n) = x(n+2) - x(n+6), \quad n = 0, 1$$

$$d(n) = [b(n) - jc(n)] W_8^n, \quad n = 0, 1$$

$$e(n) = [b(n) + jc(n)] W_8^{3n}, \quad n = 0, 1$$

由式(1), 式(2)和式(3)得:

$$X(2r) = \sum_{n=0}^3 a(n) W_8^{rn}, \quad r = 0, 1, 2, 3$$

$$X(4r+1) = \sum_{n=0}^1 d(n) W_8^{rn}, \quad r = 0, 1$$

$$X(4r+3) = \sum_{n=0}^1 e(n) W_8^{rn}, \quad r = 0, 1$$

图2中的虚线框部分就是一个L型蝶形运算结构。

3 分裂基算法的实现及与基2算法的比较

3.1 分裂基算法在Matlab中的实现

这里采用的是基于频域抽取(decimation-in-frequency)的原位(in-place)分裂基算法^[3]。

程序共分3个部分:

第1部分进行L型蝶形运算,由一个3层循环嵌套构成。最外层的循环对于 N 点FFT($N=2^M$)控制循环 M 回,只需在初始时给定 N 的值,就可以进行相应各点的FFT。中间的循环要进行 a_k 次,这里的 a_k 等于 $N/2^{k+1}$ ($0 \leq k \leq M-1$),他表示在第 k 回循环分成了 n_k ($n_k = \frac{1}{3} [2^k + (-1)^{k-1}]$)组中每一组L型蝶形运算的数目,同在一组具有相同的结构及旋转因子分布。里层的循环用来计算相邻的蝶型运算和他们的旋转因子。

第2部分单独计算基2蝶型运算。

最后一部分用来进行整序运算,在FFT中由于不断地对输入序列进行奇偶抽取,导致序列最后按称之为倒序的规律排列,所以最后要重新排序。该程序适用于实数输入(复数的实部和虚部分开运算)。这里要注意在Matlab中用数组表示时,下标是从1开始的,而不是从习惯的0开始。

3.2 分裂基算法和基2算法的比较

基2算法是最常用的FFT算法,分裂基算法有着和基2算法一样的规则结构。把分裂基和基2算法中所有无关紧要的旋转因子(像 $W^0, W^{\frac{1}{2}}, W^{\frac{1}{4}}$ 这样的旋转因子称为无关紧要的旋转因子)都考虑在内,那么两者所需的计算量其实是一样的。分裂基算法的特点是合理的安排了算法结构,使无关紧要的旋转因子最大程度的减小。

在Matlab上仿真实现了基2算法和分裂基算法,并进行了性能对比。图3显示了在Matlab中分别用基2算法和分裂基算法完成 N (N 取32, 64, 128, 256, 512, 1024, 2048, 4096)点FFT所用的时间。图中的实线表示基2算法,虚线表示分裂基算法。基2算法^[4]采用的是DIF-FFT,程序适用于实数输入(复数的实部和虚部分开运算)。在仿真时给定了一组输入数

据 $x(n)$ ($1 \leq n \leq N$), $x(n)$ ($1 \leq n \leq N/2$) 的实部取 10, 虚部取 10, 而 $x(n)$ ($N/2 + 1 \leq n \leq N$) 的实部取 0, 虚部取 10。可以发现, 随着取样点 N 的增大, 分裂基算法的优越性越来越明显。

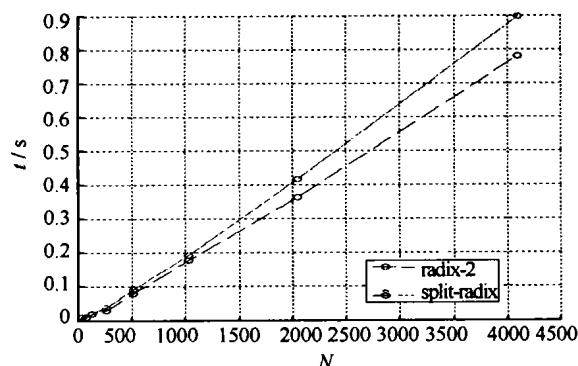


图3 基2算法和分裂基算法的比较

4 结 语

分裂基算法具有与传统的 Cooley-Tukey 算法同样好的结构, 可以同址运算, 这在用 IC 芯片来实现

这些算法时是特别重要的。本文在 Matlab 中实现了分裂基算法, 经过仿真实验, 分裂基算法与基 2 算法相比速度提高了大约 12%, 是一种高效易行的算法。

参 考 文 献

- [1] 胡广书. 数字信号处理—理论, 算法与实现 [M]. 北京: 清华大学出版社, 1997.
- [2] Duhamel P. Algorithms meeting the lower bounds on the multiplicative complexity of length- 2^N DFTs and their connection with practical algorithms [J]. IEEE Trans. on ASSP, 1990, 38 (9): 1504-1511.
- [3] Squirensen H V, Heideman M T, Burrus C S. On computing the split-radix FFT [J]. IEEE Trans., 1986, ASSP-34: 152-156.
- [4] 董绍平, 王洋, 陈世耕. 数字信号处理基础 [M]. 哈尔滨: 哈尔滨工业大学出版社, 1989.

作者简介 何 璞 女, 1978 年出生, 宁夏银川人, 西安电子科技大学电子工程学院硕士研究生。

张 平 男, 1947 年出生, 江苏人, 西安电子科技大学硕士生导师, 研究方向为信息处理。

(上接第 30 页)

图 4 所示为 PID-Fuzzy 控制系统的仿真曲线, 与图 1 比较, 可以发现输出值 y 超调很小, 过渡时间较短, 响应速度提高, 但稳态值与理想值之间有差距。同时,

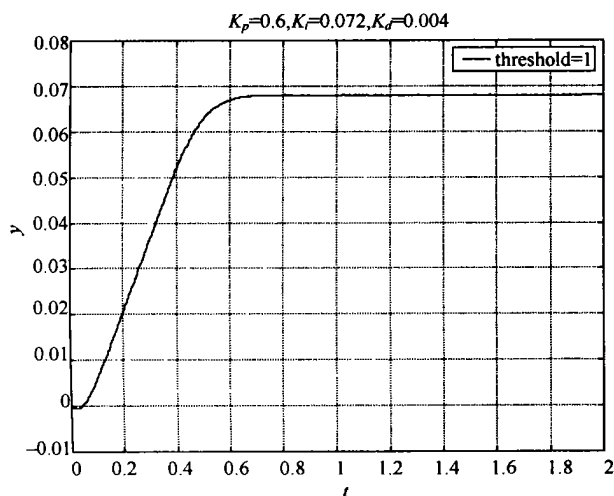


图4 PID-Fuzzy 控制系统的仿真曲线

对门限值大小的变化、论域等分与不等分、以及不同增益情况下系统的响应波形进行了研究; 他们对系统的响应速度、输入误差、超调量和过渡时间具有决定性的作用。

4 结 语

Fuzzy-PID 复合控制取代常规 PID 控制方法, 各项性能指标均有较大提高, 而且可以显著地提高了系统的可靠性及抗干扰能力。

参 考 文 献

- [1] 章卫国, 杨向忠. 模糊控制理论与应用 [M]. 西安: 西北工业大学出版社, 1999.
- [2] 孔令志, 孔令慧. 基于 GA 参数寻优的复合式 Fuzzy-PID 控制 [J]. 测试技术学报, 2003, (1): 79-82.

作者简介 马鹏飞 男, 1968 年出生, 黑龙江人, 1992 年毕业于吉林工业大学, 获学士学位。长期从事机电产品的设计、研究工作。