

粒子群优化算法在 FIR 数字滤波器设计中的应用

李辉,张安,赵敏,徐琦

(西北工业大学电子信息学院,陕西西安 710072)

摘要: 本文针对有限脉冲响应(FIR)数字滤波器的设计实质上是一个多参数优化问题,提出了一种用粒子群优化算法(PSO)设计 FIR 数字滤波器的方法.首先将滤波器的设计问题转化为滤波器参数的优化问题,然后利用粒子群优化算法对整个参数空间进行高效并行搜索以获得参数的最优化. FIR 数字低通、带通滤波器设计实例证明了该方法的有效性和优越性.

关键词: 粒子群优化算法; FIR 数字滤波器; 滤波器设计

中图分类号: TN713⁺.7 **文献标识码:** A **文章编号:** 0372-2112 (2005) 07-1338-04

Particle Swarm Optimization Algorithm for FIR Digital Filters Design

LI Hui, ZHANG An, ZHAO Min, XU Qi

(Department of Electronic Engineering, Northwestern Polytechnical University, Xi'an, Shanxi 710072, China)

Abstract: The particle swarm is an algorithm for finding optimal region of complex search spaces through the interaction of individuals in a population of particles. A new method based on particle swarm optimization(PSO) is proposed to design FIR digital filters. The design of FIR digital filters is converted into the optimization of the parameters of FIR digital filters. PSO is used to search the whole parameters space effectively and globally in order to optimize parameters. The effectiveness and superiority of the introduced method are demonstrated by experimental results on the low pass and band pass FIR digital filters. And compared with other optimization algorithms PSO has advances in computational power.

Key words: particle swarm optimization; FIR digital filters; filter design

1 引言

FIR 数字滤波器以其系统稳定性、易于实现线性相位、允许设计多通带(或多阻带)滤波器以及硬件容易实现等特点,在通信、雷达、声纳、生物医学、地震勘探等方面有广泛的应用.传统的 FIR 数字滤波器设计方法有频率抽取法、窗函数法和一致逼近法^[1].频率抽取法和窗函数法简单易行,但不易精确地确定其通带和阻带的边界频率.一致逼近法以 Parks-McClellan 算法为代表,该方法能获得较好的通带和阻带性能,并能准确地指定通带和阻带的边缘,是一种有效的设计方法.近年来一些学者对 FIR 数字滤波器的设计做了大量的研究工作,提出了用优化算法来设计 FIR 数字滤波器,它是在一定的优化准则下,使设计的滤波器性能最优.基于混合整数规划技术的设计方法就是一种全局优化算法^[2],这是目前在最大误差最小化意义上能保证全局最优解的唯一的算法,但其运算量随滤波器阶数增加呈指数增大,从而限制了该算法的应用.基于局部邻域搜索技术的设计方法是另一类重要方法^[3],该方法的主要特点是运算量小,但只能找到局部最优解,优化效果并不明显.文献[4]将模拟退火的设计方法应用到 FIR 数字

滤波器的设计中,取得了较好的效果,但由于该方法采用了随机策略,导致了运算量比较大.

粒子群优化(Particle Swarm Optimization, PSO)算法是由 Eberhart 与 Kennedy 在 1995 年发明的一种新的全局优化进化算法^[5].该算法源于对鸟类捕食行为的模拟.算法的突出特点是结构简单,运行速度快,计算量小,程序实现非常简洁,需要调整的参数少.因此,引起了众多研究者的关注,近十年来它已在很多工程计算领域得到了应用,并成为一种重要的优化工具.本文将粒子群优化算法应用于 FIR 数字滤波器的设计,给出了设计 FIR 数字滤波器的粒子群优化算法,并在计算机上进行了仿真.仿真结果证明了本方法的有效性.

2 粒子群优化算法

2.1 算法简介

与其他优化算法相似,粒子群优化算法同样基于群体(这里称作粒子群)与适应度,通过适应度将群体中的个体(这里称作粒子)移动到好的区域.粒子代表问题的一个可能解,它具有位置和速度两个特征.每个粒子都被看作是 d 维空间中的一个没有体积和质量的微粒,在搜索空间中以一定的速度

飞行,并根据对个体和集体飞行经验的综合分析来动态调整这个速度,然后更新个体当前的位置.算法首先初始化一群随机粒子,然后通过迭代找到最优解.在每一次迭代中,粒子通过跟踪两个极值来更新自己:一个是粒子本身所找到的最优解,即个体极值记为;另一个是整个粒子群目前找到的最优解,即全局极值记为.

设 S 为 d 维欧式空间的一个非空集合.粒子群中第 i 个粒子在 S 中的位置记为 $X_i = (x_{i1}, x_{i2}, \dots, x_{id})$, 粒子 i 的速度记为 $V_i = (v_{i1}, v_{i2}, \dots, v_{id})$. 粒子在找到上述两个极值后,就根据下面两个公式来更新自己的速度与位置^[6]:

$$v_{id}^{k+1} = v_{id}^k + c_1 \cdot \text{Rand}() \cdot (pbest_i^k - x_{id}^k) + c_2 \cdot \text{Rand}() \cdot (gbest^k - x_{id}^k) \quad (1)$$

$$x_{id}^{k+1} = x_{id}^k + v_{id}^{k+1} \quad (2)$$

其中 k 为迭代次数, w 为惯性权重(inertia weight),它使粒子保持运动的惯性,使其有能力探索新的区域. c_1 和 c_2 为加速常数,它们使每个粒子向 pbest 和 gbest 位置加速运动. Rand 为 $[-1, 1]$ 范围内变化的随机数.此外粒子的速度 V_i 被一个最大速度 V_{max} 限制.如果当前粒子在某维的速度 v_{id} 超过该维的最大速度 v_{maxd} ,则该维的速度被限制为最大速度 v_{maxd} . V_{max} 决定了粒子在解空间的搜索精度,如果 V_{max} 太大,粒子会飞过最优解,如果 V_{max} 太小,粒子则陷入局部最优解而无法进行全局搜索.

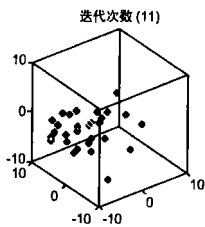


图 1 粒子搜索示意图 (迭代初期)

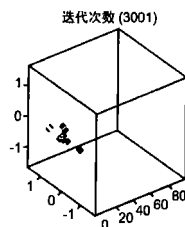


图 2 粒子搜索示意图 (迭代结束)

在算法迭代初期,粒子在较大的空间内进行搜索,如图 1 所示.随着迭代次数的增加,粒子在个体极值和全局极值的引导下,逐渐收敛到小的范围,在迭代结束时粒子最终收敛的位置代表算法找到的最优解,如图 2 所示.粒子群优化算法的流程图如图 3 所示.

2.2 算法解析

公式(1)和(2)表示了粒子在求解空间中,由于相互影响导致了运动速度和位置的调整.惯性权重、加速常数 c_1 和 c_2 和最大速度 V_{max} 共同维护粒子对全局和局部搜索能力的平衡,因此

此算法的参数设置与算法系统性能之间有着密切的联系.在实际应用中算法参数的设置与具体问题密切相关,目前比较常用的参数设置方法是针对具体求解问题,利用充分的实验

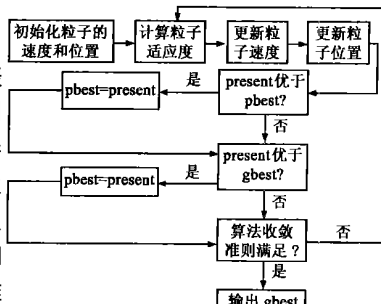


图 3 粒子群优化算法流程图

来确定.

早期的 PSO 算法一般令 $c_1 = c_2 = 2.0$, 以保证公式(1)中的随机乘积均值等于 1^[5]. V_{max} 一般设置为粒子的最大位移 X_{max} , 即 $V_{max} = X_{max}$ ^[7]. 而一般在 0.9 到 0.4 之间取值.文献[6]通过大量的实验证明:如果 w 随算法迭代的进行而线性减小,将显著改善算法的收敛性能.设 w_{max} 为最大惯性权重, w_{min} 为最小惯性权重, run 为当前迭代次数, run_{max} 为算法迭代总次数,则有:

$$w = w_{max} - \text{run} \cdot \frac{w_{max} - w_{min}}{\text{run}_{max}} \quad (3)$$

通过这样的改进,算法初期 w 较大,算法搜索的空间也就较大;算法晚期, w 很小,可以提高算法的精细搜索能力.文献[8]指出,在 0.3 ~ 0.5 范围内,算法速度有所提高,且设置水平不会导致优化效果的显著变化.参数 c_1 对算法整体性能的影响较弱,但参数的变化对算法整体的特性有着显著的影响.因此,在优化效果要求较高的应用中,应优先考虑调整 c_2 的设置水平,以改变粒子间相互影响的程度,从而提高每个粒子察觉其他粒子的能力,实现更好的适应性调整.

3 设计 FIR 数字滤波器的粒子群优化算法

设 N 阶 FIR 数字滤波器的单位取样响应为 $h(0), h(1), \dots, h(N-1)$, 则其传递函数可表示为^[11]:

$$H(z) = \sum_{n=0}^{N-1} h(n) z^{-n} \quad (4)$$

取 $z = e^{j\omega}$, 则滤波器的频率响应为:

$$H(e^{j\omega}) = \sum_{n=0}^{N-1} h(n) e^{-j\omega n} \quad (5)$$

若设 FIR 数字滤波器的理想频率响应为 $|H_d(e^{j\omega})|$, 则在离散点 $\{\omega_i | i = 1, 2, \dots, M\}$ 上,所设计滤波器的幅度 $|H(e^{j\omega_i})|$ 与理想滤波器的幅度 $|H_d(e^{j\omega_i})|$ 的误差平方和为:

$$E = \sum_{i=1}^M (|H(e^{j\omega_i})| - |H_d(e^{j\omega_i})|)^2 \quad (6)$$

把式(5)代入式(6),可得

$$E = \sum_{i=1}^M \left(\left| \sum_{n=0}^{N-1} h(n) e^{-j\omega_i n} \right| - |H_d(e^{j\omega_i})| \right)^2 \quad (7)$$

显然 E 是滤波器系数 $h(n)$ 的非线性函数,因此 E 是一个有 N 个未知数的函数.根据频域最小均方误差准则, FIR 滤波器的设计,就是要选取滤波器系数 $h(0), h(1), \dots, h(N-1)$ 使目标函数 E 最小,显然这是一个组合优化问题,因此我们可用 PSO 算法来求解上述组合优化问题.

3.1 适应度函数

PSO 算法通过适应度来确定粒子当前位置的优劣,所以必须根据实际问题的需要选择适应度函数 F . 这里选择函数式(7)作为 FIR 滤波器设计的适应度函数.即:

$$F = E = \sum_{i=1}^M \left(\left| \sum_{n=0}^{N-1} h(n) e^{-j\omega_i n} \right| - |H_d(e^{j\omega_i})| \right)^2 \quad (8)$$

显然, F 函数值越小,则该粒子对应的滤波器系数的均方误差就越小,该粒子就对应更佳的滤波器系数.算法结束后,整个运行期间适应度最小的粒子所代表的参数值就是算法得

到的最优解,也就是算法最终得到的滤波器系数。

3.2 参数编码

为了用 PSO 算法求解 $h(n)$, 我们应对优化变量 $h(n)$ 进行适当的编码, 以形成 PSO 算法中的粒子。根据 PSO 算法本身的特点, 可以直接用实数来表示各参数。另外对每一个粒子而言, 如果用 $h(n)$ 表示当前位置, 那么还应该有一个对应的 V 表示粒子的速度。 $h(n)$ 是一个 N 维的变量, 所以 V 也应该是一个 N 维变量, 即:

$$V = \{V_0, V_2, \dots, V_{N-1}\}.$$

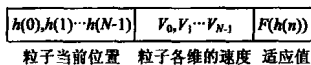


图 4 粒子编码结构

这样, 粒子就可以采用如图 4 所示的编码结构:

3.3 用 PSO 算法设计 FIR 滤波器的步骤

- (1) 给定滤波器的技术指标;
- (2) 设定 PSO 算法的参数, 包括群体大小, 参数维数, 惯性权重等;
- (3) 根据计算出的滤波器系数初始化各粒子的位置与速度;
- (4) 根据式 (8) 计算粒子的适应度;
- (5) 根据式 (1) 和式 (2) 更新粒子的速度与位置;
- (6) 重新计算粒子适应度;
- (7) 判断是否更新粒子的个体极值 p_{best} 和粒子群的全局极值 g_{best} ;
- (8) 重复 (5) ~ (7) 步, 直到满足精度要求或达到预先设定的迭代次数;
- (9) 输出 g_{best} , 得到优化后的 FIR 数字滤波器的系数 $h(0), h(1), \dots, h(N-1)$;

4 实验仿真结果

为了验证本文算法的有效性, 在计算机上采用 Matlab 语言进行了 FIR 数字滤波器设计的仿真实验。同时, 为了比较算法性能, 还采用 Parks-McClellan 算法进行了滤波器优化设计。仿真实验中, 粒子群优化算法的参数设置为: 群体大小 $popsize = 50$, 参数维数 $dim\ size = 12$, 最大惯性权重 $max = 0.9$, 最小惯性权重 $min = 0.4$, 粒子群中所有粒子位置的取值范围都限制在 $[-1, 1]$, 最大速度 $V_{max} = 1$, 最大迭代次数 $iterations = 2000$ 。

例 1 设计一个阶数为 30 的 FIR 低通数字滤波器, 其技术指标为:

$$H_d(e^j) = \begin{cases} 1, & 0 & 0.4 \\ 0, & 0.5 & \end{cases}$$

例 2 设计一个阶数为 30 的 FIR 带通数字滤波器, 其技术指标为:

$$H_d(e^j) = \begin{cases} 1, & 0 & 0.28 & 0.72 \\ 0, & 0.32 & 0.68 & \end{cases}$$

图 5 和图 7 分别是两种算法得到的 FIR 低通和带通滤波器的幅频响应, 其阻带和通带的衰减程度如图 6 和图 8 所示。从图中可以看到, PSO 算法得到的滤波器幅频响应的通带和阻带优化结果都比较理想, 阻带衰减性能明显优于 Parks-McClellan 算法, 通带的最大波动更小, 这恰好是设计所需要的。但从图中也可以看到用 PSO 算法设计的滤波器其阻带和通

带的边界频率处误差较大, 而用 Parks-McClellan 算法设计的滤波器在通带内具有等波纹特性, 在边界频率处更接近于理想响应, 这表明 Parks-McClellan 算法设计的滤波器在通带和阻带上的最大误差更小。

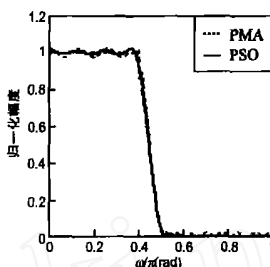


图 5 FIR 低通数字滤波器的幅频响应

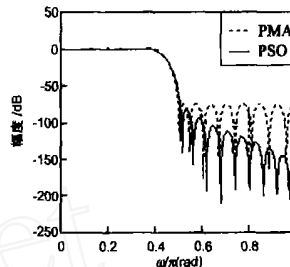


图 6 FIR 低通数字滤波器的对数幅频响应

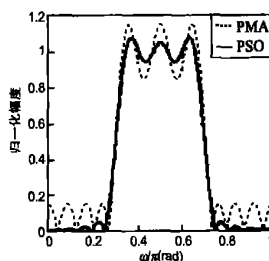


图 7 FIR 带通数字滤波器的幅频响应

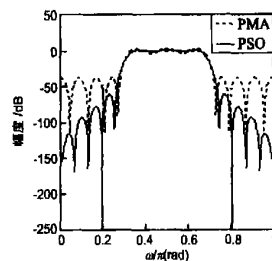


图 8 FIR 带通数字滤波器的对数幅频响应

表 1 PSO 与 GA 算法性能比较

算法	CPU 时间(s)	最大迭代次数	均方误差
PSO	38.4	2653	1.726
GA	57.6	2861	3.572

分析 PSO 算法在计算量上的性能, 仿真中将遗传

算法 (GA) 也用于低通滤波器的设计, 两种算法的迭代次数均为 3000 次, 结果如表 1。由表 1 可知 PSO 算法的计算量明显小于 GA 算法, 而且其优化性能也好于后者。这是因为 PSO 保留了基于种群的全局搜索策略, 但是其采用的速度-位移模型操作简单, 避免了复杂的遗传操作。

此外通过仿真分析还发现粒子群优化算法的性能受其初始值的影响不大。初始值设置的范围只会影响算法的收敛速度, 而不会影响收敛结果。只要迭代次数设置的合理, 在迭代次数范围内粒子总会找到全局最优值。

总之, 用 PSO 算法设计的滤波器在总体上能够得到令人满意的性能, 因此该方法是一种有效的设计方法。

5 结论

作为新的优化算法, 粒子群优化算法给大量非线性、不可微和多峰值复杂问题的优化提供了一种新的思路。本文将 PSO 算法用于 FIR 数字滤波器设计, 仿真结果显示了本方法的可行性。该方法不仅可以用于设计低通、带通、带阻滤波器, 而且还可以设计多通带滤波器。其主要特点是需要选择的参数少, 程序实现简洁, 并在种群数量、寻优速度方面具有一定的优势, 运算量相对较小。

与其他的优化算法一样, PSO 算法同样存在陷入局部最

优解的问题。目前解决这一问题的主要方法是增加粒子群的规模,调整算法中各参数的设置。那么如何有效避免 PSO 算法陷入局部最优解,并进一步提高滤波器的设计质量是需要进一步研究的内容。

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] D Kodak. Design of optimal finite wordlength FIR digital filters with linear phase[J]. IEEE Trans on Assp ,1980 ,28(3) :304 - 308.
- [3] Y C Lin ,S P Parker. FIR filter design over a discrete power-of-two coefficient space[J]. IEEE Trans on Assp ,1983 ,31(3) :583 - 591.
- [4] N Benvenuto ,M Marchesi. Digital filters design by simulated annealing [J]. IEEE Trans on Circuit and System ,1989 ,36(3) :459 - 465.
- [5] Kennedy J ,Eberhart R C. Particle swarm optimization[A]. Proc IEEE Int Conf on Neural Network[C]. Perth ,Australia : IEEE ,1995. 1942 - 1948.
- [6] Shi Y ,Eberhart R C. A modified swarm optimizer[A]. IEEE International Conference on Evolutionary Computation [C]. Anchorage ,AK USA : IEEE ,1998. 69 - 73.
- [7] R C Eberhart ,Y Shi . Comparing inertia weights and constriction factors in particle swarm optimization [A]. 2000 Congress on Evolutionary Computation[C]. CA USA :La Jolla ,2000. 84 - 88.
- [8] 彭宇,彭喜元,刘兆庆. 微粒群算法参数效能的统计分析[J]. 电子学报 ,2004 ,32(3) :209 - 213.

作者简介:

李 辉 男,1968年6月出生于陕西省,现为西北工业大学电子信息学院副教授,在读博士研究生,研究方向为系统工程及信号处理。E-mail :lihui2601@hotmail.com.

张 安 男,1962年生于陕西省,现为西北工业大学电子信息学院教授,博士生导师,研究方向为航空武器电子综合化系统及综合电子战仿真与效能评估分析等。