继电保护装置递推算法的容错机制

王 亮 吕 航 顾乔根 张晓宇 李 力
（南京南瑞继保电气有限公司，南京 211102）

摘要 递推傅里叶算法是常用的继电保护数字信号处理算法，其特点是利用上次计算结果推算本次结果，优点是计算效率非常高。但是，递推算法具有记忆效应，偶然因素导致的错误会一直保持且不易察觉，给继电保护装置的安全稳定运行带来隐患。因此，本文提出一种适用于递推算法的容错机制，可消除递推算法的记忆效应，防止因偶然出错导致不可恢复的计算偏差，从而有效防范此类问题引发继电保护误动或拒动。

关键词：继电保护；递推算法；容错；记忆效应

Fault-tolerant algorithm for recursive algorithm in protection relays

WANG Liang LÜ Hang GU Qiaogen ZHANG Xiaoyu LI Li
(NR Electric Co., Ltd, Nanjing 211102)

Abstract Recursive Fourier algorithm is a commonly used digital signal processing algorithm for protection relays. Its characteristic is to use the last calculation result to calculate the current result. Its calculation efficiency is very high. However, the recursive algorithm has a memory effect. Errors caused by accidental factors will be kept and be difficult to detect, bringing hidden dangers to the operation of the protection device. In this paper, a fault-tolerant mechanism suitable for recursive algorithm is proposed to eliminate the memory effect and prevent irrecoverable calculation deviation caused by occasional or accumulated errors, thus effectively preventing such problems from causing misoperation or failure of protection devices.

Keywords：protection relays; recursive algorithm; fault-tolerant; memory effect

0 引言

算法是微机保护研究的重点之一，目前已提出的算法有多种。分析和评价不同算法优劣的指标是精度和速度，计算精度越高，意味着所需采样点越多、计算工作量越大，所以研究微机保护算法的实质是如何更好地权衡速度和精度。

傅里叶算法是微机保护中最常用的算法，其特性优良，目前仍是信号处理领域应用最广泛的算法之一。在实际的微机保护算法实现过程中，全周傅里叶算法需要利用信号一个周波的采样点进行 $2n$（n 为一个周波的采样点数）次乘加操作，采样率越高，计算耗时越多。递推傅里叶算法是用于离散傅里叶计算的一种算法，它充分利用已有计算结果来减少计算量，无论 n 为何值，每次计算只进行 4 次乘加操作，极大地提高了计算效率，因而被广泛应用于继电保护及相关的专业领域[1-9]。

然而，递推算法具有“记忆效应”，即当已有的计算结果被意外修改时，后续的计算偏差将一直保持[10-12]。例如：由于电磁干扰，微机保护装置中某个保存递推计算结果的内存单元被误修改，极有可能在特定条件下导致保护误动或拒动，进而酿成事故。实际工程中，此类事件并不鲜见，但由于错误非常隐蔽，一直以来未引起足够重视。虽然采用非递推的全周傅里叶算法不会出现此类问题，但是在采样率高、输入通道多的情况下，计算量将增长几十倍，极大地占用处理器的计算资源，甚至导致 CPU 超负荷。因此，本文提出一种递推算法的容错机制，以解决上述问题。

1 递推算法特性分析

为便于分析问题，本文先从简单的平均值算法

入手，分析递推算法的特性。平均值计算的一般表达式为

$$A=\frac{1}{n}\sum_{i=1}^{n}x_i \tag{1}$$

式中：A 为平均值；n 为数据窗长度；x_i 为第 i 个采样值。

设采样数据窗 1 为 $S_1=\{x_1, x_2,\cdots, x_n\}$，数据窗 2 为 $S_2=\{x_2, x_3,\cdots, x_{n+1}\}$，则 S_1 和 S_2 的平均值 A_1 和 A_2 分别为

$$A_1=\frac{1}{n}\left(x_1+x_2+x_3+\cdots+x_{n-1}+x_n\right) \tag{2}$$

$$A_2=\frac{1}{n}\left(x_2+x_3+x_4+\cdots+x_n+x_{n+1}\right) \tag{3}$$

每进行一次平均值计算，都需要进行 n 次加法操作，对比式（2）和式（3）不难发现，二者只有两项差别，即式（2）中第一项和式（3）中最后一项，其余均相同，因此可由式（2）递推出式（3），而无需每次都进行 n 次加法操作，公式为

$$nA_2=nA_1-x_1+x_{n+1} \tag{4}$$

由式（4）得到平均值递推算法的一般表达式为

$$nA_{i+1}=nA_i-x_i+x_{n+i} \tag{5}$$

可见无论 n 为何值，递推公式只有 2 次加减操作即可完成平均值计算，效率极高。

值得注意的是，在实际算法实现过程中，保护装置上电复位后需要对数据窗 S_1 进行初始化，若没有初始化、初始化异常或偶然因素引起内存修改，以及出现浮点数计算的字长截断效应，都会导致递推算法中 A_i 异常，由于递推算法的记忆效应，该异常将被一直保持且不易察觉，从而造成重大隐患。因此，必须找到一种机制来避免记忆效应可能带来的不良影响。

2 递推算法容错机制

平均值递推算法采用连续数据窗，连续数据窗是指两个数据窗只有一个采样值的差别。每个采样中断刷新一个采样值并计算一次平均值。观察式（5）可知，记忆效应存在于 nA_i 中，即第 i 个数据窗中所有采样值的和，其值对应非递推算法表达式为

$$nA_i=x_i+x_{i+1}+\cdots+x_{i+n-1} \tag{6}$$

经过 n 次递推计算后，对应的非递推表达式为

$$nA_{i+n}=x_{i+n}+x_{i+n+1}+\cdots+x_{i+2n-1} \tag{7}$$

对比式（6）和式（7）可见，二者是相邻的数据窗，即两个数据窗采样序列首尾连续，但刚好没有相同项。如果针对每个相邻的数据窗建立一种监测和刷新机制，定时消除递推算法的记忆效应，则可实现算法容错，具体实现机制如下。

递推算法计算平均值，对应的非递推公式从式（6）到式（7）共进行 n 次平均值递推计算，如果在每个相邻的数据窗进行一次非递推完整求和操作，所获得的值与递推值应该相等，若不相等则表示出现异常，保护程序应进行纠错处理。由于进行一次完整的求和操作会显著增加某个中断的计算量，不是最优算法，所以可将求和过程分布在 n 次递推过程中，即每个中断只累加式（7）中的一项，整个求和过程在从式（6）到式（7）的过程中分 n 次完成，这样每个中断只增加一次加法操作，对整个递推算法的计算效率几乎没有影响。设用于容错求和计算的变量为 R，则有

$$R_n=\sum_{k=i+n}^{i+2n-1}x_k \tag{8}$$

式（8）的求和运算不是一次全部完成，而是每个采样中断只累加一个最新的采样数据，每经过 n 次中断得到一个结果 R_n，正常情况下有 $nA_{i+n}=R_n$，若 $nA_{i+n}\neq R_n$ 则说明出现异常，应采取措施，这样即形成容错机制。每经过一个完整数据窗长度，就对递推计算结果进行一次校验，一般应用中可在每 n 次递推后强制令 $nA_{i+n}=R_n$， 以消除记忆效应，起到纠错的作用。需要注意的是，每 n 次累加校验完成后 R_n 要清零，重新开始下一个数据窗的累加。

3 容错机制在递推傅里叶算法中的应用

3.1 递推傅里叶容错原理

全周傅里叶算法是保护装置常用算法之一，其基本思想源于傅里叶级数。假设输入信号为一个周期性时间函数 $x(t)$，包含基频分量、恒定直流分量及整数次谐波分量，可表示为

$$x(t)=\frac{a_0}{2}+\sum_{k=1}^{\infty}\left[a_k\cos(k\omega t)+b_k\sin(k\omega t)\right] \tag{9}$$

式中：k 为自然数，代表谐波次数；$a_0/2$ 为直流分量；a_k 和 b_k 分别为各次谐波的余弦项和正弦项的振幅；ω 为基波角频率。

使用计算机处理时，对基波分量的离散化表达式为

$$a_1 = \frac{2}{n}\sum_{i=1}^{n}\left(x_i \cos\frac{2i\pi}{n}\right) \tag{10}$$

$$b_1 = \frac{2}{n}\sum_{i=1}^{n}\left(x_i \sin\frac{2i\pi}{n}\right) \tag{11}$$

式中：a_1 和 b_1 为基波余弦项和正弦项的振幅；x_i 为采样点；n 为工频基波每周波采样点数。

与前述的平均值递推公式类似，傅里叶算法的递推表达式为

$$a_{1\_i+1} = a_{1\_i} - \frac{2}{n}x_i\cos\frac{2i\pi}{n} + \frac{2}{n}x_{i+n}\cos\frac{2i\pi}{n} \tag{12}$$

$$b_{1\_i+1} = b_{1\_i} - \frac{2}{n}x_i\sin\frac{2i\pi}{n} + \frac{2}{n}x_{i+n}\sin\frac{2i\pi}{n} \tag{13}$$

在保护装置实现中，为使表达式简洁并降低计算量，$2/n$ 一项未包含在递推计算中，而是最后统一处理，因此下文表达式不出现 $2/n$ 项。设工频基波一周波采样数据窗 $S_i=\{x_i, x_{i+1},\cdots, x_{i+n-1}\}$，傅里叶因子为$\{c_1, c_2,\cdots, c_n\}$和$\{s_1, s_2,\cdots, s_n\}$，将第 i 个数据窗的非递推傅里叶计算以相量形式表示，则实部和虚部的递推表达式分别为

$$R_{\mathrm{e}(i+1)} = R_{\mathrm{e}i} + \left(x_{i+n} - x_i\right) * \cos(i \bmod n) \tag{14}$$

$$I_{\mathrm{m}(i+1)} = I_{\mathrm{m}i} + \left(x_{i+n} - x_i\right) * \sin(i \bmod n) \tag{15}$$

式中：R_e 表示实部；I_m 表示虚部；i MOD n 表示傅里叶因子是以 n 为模的求余运算。

非递推表达式的矩阵形式为

$$R_{\mathrm{e}(i+1)} = \left[x_{i+1}\ x_{i+2}\ \cdots\ x_{i+n}\right]\begin{bmatrix}\cos 1\\ \cos 2\\ \vdots\\ \cos n\end{bmatrix} \tag{16}$$

$$I_{\mathrm{m}(i+1)} = \left[x_{i+1}\ x_{i+2}\ \cdots\ x_{i+n}\right]\begin{bmatrix}\sin 1\\ \sin 2\\ \vdots\\ \sin n\end{bmatrix} \tag{17}$$

经过 n 次递推计算后，对应的非递推表达式为

$$R_{\mathrm{e}(i+n+1)} = \left[x_{i+n+1}\ x_{i+n+2}\ \cdots\ x_{i+2n}\right]\begin{bmatrix}\cos 1\\ \cos 2\\ \vdots\\ \cos n\end{bmatrix} \tag{18}$$

$$I_{\mathrm{m}(i+n+1)} = \left[x_{i+n+1}\ x_{i+n+2}\ \cdots\ x_{i+2n}\right]\begin{bmatrix}\sin 1\\ \sin 2\\ \vdots\\ \sin n\end{bmatrix} \tag{19}$$

矩阵表达式清晰地展示出两个相邻数据窗的计算细节。仿照第 2 节的处理思路，可得到递推傅里叶算法的容错公式。设用于容错计算的实部和虚部变量分别为 R_{Re} 和 R_{Im}，有

$$R_{\mathrm{Re}(i+n+1)} = \sum_{k=i+n+1}^{i+2n}\left[x_k \cos(k \bmod n)\right] \tag{20}$$

$$R_{\mathrm{Im}(i+n+1)} = \sum_{k=i+n+1}^{i+2n}\left[x_k \sin(k \bmod n)\right] \tag{21}$$

式（20）和式（21）分 n 次中断完成，每经过 n 次递推计算校验一次 $R_{\mathrm{e}(i+n+1)}=R_{\mathrm{Re}(i+n+1)}$ 与 $I_{\mathrm{m}(i+n+1)}=R_{\mathrm{Im}(i+n+1)}$是否成立，每个采样中断只增加 2 次乘加操作，对计算效率的影响非常小，整个容错机制实现步骤与前述平均值递推算法容错机制完全相同，此处不再赘述。

3.2　算法验证

设有一组针对 50Hz 信号，采样率为 1 200Hz 的 48 点连续采样数据，为方便验证，将其分为各 24 点相邻的数据窗 S_0 和 S_1，S_0={6 553, 6 330, 5 675, 4 633, 3 276, 1 696, 0, –1 696, –3 276, –4 633, –5 675, –6 330, –6 553, –6 330, –5 675, –4 633, –3 276, –1 696, 0, 1 696, 3 276, 4 633, 5 675, 6 330}，S_1={4 095, 3 956, 3 547, 2 896, 2 047, 1 060, 0, –1 060, –2 047, –2 896, –3 547, –3 956, –4 095, –3 956, –3 547, –2 896, –2 047, –1 060, 0, 1 060, 2 047, 2 896, 3 547, 3 956}。傅里叶因子 cos(i MOD n)={32 767, 31 650, 28 377, 23 169, 16 383, 8 480, 0, –8 480, –16 383, –23 169, –28 377, –31 650, –32 767, –31 650, –28 377, –23 169, –16 383, –8 480, 0, 8 480, 16 383, 23 169, 28 377, 31 650}，sin(i MOD n)={0, 8 480, 16 383, 23 169, 28 377, 31 650, 32 767, 31 650, 28 377, 23 169, 16 383, 8 480, 0, –8 480, –16 383, –23 169, –28 377, –31 650, –32 767, –31 650, –28 377, –23 169, –16 383, –8 480}。

根据式（14）和式（15）按一周波 24 点计算采样序列 S_0 的递推傅里叶积分，递推一周波后得到实部 R_e=2 576 559 262，虚部 I_m=0。此结果与根据式（20）和式（21）非递推计算结果一致，即 $R_\mathrm{e}=R_{\mathrm{Re}}$，

$I_m=R_{Im}$。假设此时保存递推运算结果的 I_m 变量其内存的最低位受干扰改变了 1 个 bit，即 I_m 的值由 0 变 1，在此基础上继续根据式（14）和式（15）进行 S_1 采样序列的递推傅里叶积分，再经过 24 点计算后得到实部 R_e=1 610 293 106，虚部 I_m=1。此时根据式（20）和式（21）计算采样序列 S_1 非递推傅里叶积分的结果为实部 R_{Re}=1 610 293 106，虚部 R_{Im}=0。可以看出，递推傅里叶积分中实部计算保持正确，但虚部的错误被延续，当检测到 $R_e=R_{Re}$，$I_m \neq R_{Im}$ 时，可令 $I_m=R_{Im}$ 来纠正这个错误。

4 结论

本文提出了一种适用于递推算法的容错机制，理论推导显示，对于采样序列的傅里叶计算，每经过一个完整计算窗长度，递推和非递推计算值会完全相等一次，利用这一特性实现了傅里叶容错算法。时间维度上，在原有递推算法的基础上仅增加了 $1/n$ 时间开销；空间维度上，每次递推算法仅增加一个变量，即可检测和消除递推算法的记忆效应。算例验证了该算法的有效性。此外，在多种型号保护装置中的实测结果表明，该算法带来的计算负荷增量可忽略不计，采样率越高、采样通道越多，算法的收效越明显，具有等同递推傅里叶算法的高效特性。目前，此方法已在各类保护装置递推算法中得到实际运用，在几乎不增加计算负荷的同时，有效避免了记忆效应引起的计算错误，提高了保护装置的可靠性。

参考文献

[1] 刘一江. 微机距离保护的新方法[J]. 电工技术学报, 1999, 14(6): 65-68.

[2] 陈明凯, 郑翔骥, 汪晓强. 减少频谱泄漏的一种新的等角度间隔采样递推算法[J]. 电工技术学报, 2005, 20(8): 94-98.

[3] 罗蛟, 江亚群, 黄纯, 等. 基于 DRSC 窗递推 DFT 算法的电力谐波检测[J]. 电工技术学报, 2013, 28(9): 47-53.

[4] 刘校销, 郑涛, 黄婷. 基于等效漏电感参数辨识的磁控式并联电抗器匝间故障保护方案[J]. 电工技术学报, 2020, 35(1): 134-145.

[5] 韦先灿, 高伟, 杨耿杰. 基于改进动态线损估计的智能电表误差估计方法[J]. 电气技术, 2022, 23(2): 7-12.

[6] 和嘉宇, 宗鸣. 微型断路器中电热式电流检测参数辨识研究[J]. 电气技术, 2020, 21(6): 50-55.

[7] 刘红星. 基于真空断路器选相控制过零点检测的研究[J]. 电气技术, 2016, 17(9): 51-54.

[8] 贺家李, 宋从矩. 电力系统继电保护原理[M]. 4 版. 北京: 中国电力出版社, 2004.

[9] 张立华, 徐文立, 常成, 等. 一种适用于微机保护的新的递推 DFT 算法[J]. 电力系统自动化, 2000, 24(5): 28-31.

[10] 叶其孝, 沈永欢. 实用数学手册[M]. 2 版. 北京: 科学出版社, 2006.

[11] STEIN E M, WEISS G. Introduction to Fourier analysis on Euclidean spaces[M]. NJ, USA: Princeton University Press, 1971.

[12] BRACEWELL R N. The Fourier transform and its applications[M]. 3rd ed. Boston, USA: McGraw Hill, 2000.

收稿日期：2024-09-18

修回日期：2024-10-31

作者简介

王 亮（1973—），男，江苏省南京市人，硕士，高级工程师，主要从事继电保护方向研究工作。