最小二乘算法的研究与改进

王毅非

(西南交通大学电气工程学院 , 四川 成都　610031)

摘要: 最小二乘法具有良好的滤波和数据窗可变的特性 ,但运算量很大 ,以至于无法满足微机保护实时性的要求。文中针对最小二乘算法的这一缺陷 ,通过深入的分析和研究提出了最小二乘改进算法 ,改进后的算法不仅具有良好的滤波特性和数据窗可变的特点 ,而且运算量大大减少 ,从而使高精度的最小二乘算法能够很好地应用于微机保护中。理论分析和计算机仿真表明 :改进算法能准确估计出待求量。

关键词: 最小二乘;　算法;　保护

中图分类号 : TM771　　**文献标识码** : A　　**文章编号** : 1003-4897(2000)03-0005-04

就微机保护而言 ,保护的算法决定了保护的性能 ,提高保护算法的精度和速度可以使保护准确、高速、灵敏地检出故障。尽管当今计算机芯片运算速度和计算精度得到了大幅度的提高 ,但计算精度和速度的问题仍然是保护算法要解决的关键。在保护中得到广泛应用的傅氏算法 ,滤波特性好、精度高 ,但数据窗长 ,保护的响应时间长。与傅氏算法相比 ,最小二乘算法除具有滤波特性好、精度高的特点之外 ,其数据窗可变 ,但由于其运算量过大 ,难以发挥其优势 ,从而在保护中的应用受到了限制。为此 ,对最小二乘算法进行改进 ,并与当今计算机芯片和数据采集系统相结合 ,获得除具有高精度和良好滤波特性外 ,又具有较小的运算量和短数据窗的算法具有重要的意义。

1　最小二乘法基本原理与存在问题

在电力系统中 ,电流的波形中存在有衰减的直流分量及非整倍数高频分量 ,可以写成下式

$$i(t) = I_0 e^{-\frac{t}{\tau}} + \sum_{n=1}^{N}(I_{nc}\cos n\omega_0 t + I_{ns}\sin n\omega_0 t) + e(t) \tag{1}$$

式中　$e(t)$ 为非整倍数频率分量及噪声或称为干扰或误差 ,假定其为白噪 ; $I_0 e^{-\frac{t}{\tau}}$ 为衰减的直流分量 ,可展开成级数形式 ,取前两项足以满足实际工程精度的要求 :

$I_0 e^{-\frac{t}{\tau}} = I_0 - K_1 t$　令 $K_1 = I_d$　得 $I_0 e^{-\frac{t}{\tau}} = I_0 - I_d t$

对 (1) 式电流信号进行采样 ,为分析方便假设信号中只含 5 次以下谐波和白噪 ,则第 k 次采样值为

$$i_k = I_0 - I_d k\Delta T_s + I_{1c}\cos\omega_0 k\Delta T_s + I_{1s}\sin\omega_0 k\Delta T_s + \cdots + I_{5c}\sin\omega_0 5k\Delta T_s + e_k \tag{2}$$

采样 n 次并令

$$Y = [\, i_1 \quad i_2 \quad \cdots \quad i_n \,]^T \tag{3}$$

$$I = [\, I_0 \quad I_d \quad I_{1c} \quad I_{1s} \quad I_{2c} \quad I_{2s} \quad \cdots \,] \tag{4}$$

$$W = [\, e_1 \quad e_2 \quad \cdots \quad e_n \,]^T \tag{5}$$

$$A = \begin{bmatrix} 1 & -\Delta T_s & \cos\omega_0\Delta T_s & \sin\omega_0\Delta T_s & \cdots & \sin5\omega_0\Delta T_s \\ 1 & -2\Delta T_s & \cos\omega_0 2\Delta T_s & \sin2\omega_0\Delta T_s & \cdots & \sin5\omega_0 2\Delta T_s \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 1 & -n\Delta T_s & \cos n\omega_0\Delta T_s & \sin n\omega_0\Delta T_s & \cdots & \sin5\omega_0 n\Delta T_s \end{bmatrix} \tag{6}$$

则 :

$$Y = AI + W \tag{7}$$

待求量 I 的最小二乘估计值 $\hat{I}$ 为

$$\hat{I} = [\, A^T A \,]^{-1} A^T Y = BY \tag{8}$$

式中 B 可离线确定 ,若待求的未知量 I 为 $m \times 1$ 维 ,观测量 Y 为 $n \times 1$ 维 ,则

$$B = \begin{bmatrix} b_{11} & b_{12} & \cdots & b_{1n} \\ b_{21} & b_{22} & \cdots & b_{2n} \\ & \cdots & \cdots & \cdots \\ b_{m1} & b_{m2} & \cdots & b_{mn} \end{bmatrix}$$

为 m × n 维。

由此可以看出 ,该算法能滤除高频成分、直流分量和白噪 ,根据精度和速度的要求数据窗可增可减 ,但存在以下两个问题 :

(1) 为了计算出某个待求量需要进行 n 次乘法和 n - 1 次加法 ,要计算出 m 个待求量则需要 m × n 次乘法和 m × (n - 1) 次加法 ,若待求量个数 m = 12 ,为了保证精度 $n \geq 2 \times m + 2 = 26$,则计算 m 个待求量需要 312 次乘法 ,实时计算量过大 ,再好的芯片

收稿日期 :　1999-09-27 ;　　**改回日期** :　1999-11-22

作者简介 :　王毅非 (1961 -) ,男 ,硕士 ,讲师 ,从事铁道电气化及自动化的研究工作。

也难以满足要求。

(2)为了提高响应速度,需缩小数据窗长度,但数据窗长度若缩短,在保证精度不变的前提下,采样频率必须加大,为此运算量过大这一问题就显得更加突出。

由此可知降低最小二乘算法的运算量是解决问题的关键。

2 最小二乘算法的改进

2.1 一步递归算法

不失一般性,将(2)式改写为

$$i(k) = a_1(k) I_0 + a_2(k) I_d + a_3(k) I_{1s} + a_4(k) I_{1c} + \cdots + a_m(k) I_{5s} + e(k) \quad (9)$$

令

$$a(k) = [\, a_1(k) \quad a_2(k) \quad \cdots \quad a_m(k) \,]^T \quad (10)$$

$$A(k,n) = [\, a(k) \quad a(k+1) \quad \cdots \quad a(k+n-1) \,]^T \quad (11)$$

$$Y(k,n) = [\, i(k) \quad i(k+1) \quad \cdots \quad i(k+n-1) \,]^T \quad (12)$$

$$P(k,n) = A^T[k,n] A(k,n) = \sum_{i=k}^{k+n-1} a(i) a^T(i) \quad (13)$$

以上各式中 $Y(k,n)$ 及其对应的 $A(k,n)$ 和 $P(k,n)$ 含 k 至 $(k+n-1)$ 时刻共 n 个数据所提供的信息。n 表示所含信息的数据个数也就是数据窗的长度,k 表示所含信息的起始时刻。

将(11)、(12)、(13)式代入(8)式可以得到

$$\hat{A}(k,n) = P^{-1}(k,n) A^T(k,n) Y(k,n) \quad (14)$$

令 $k = k+1$

则 $$\hat{A}(k+1,n) = P^{-1}(k+1,n) A^T(k+1,n) Y(k+1,n) \quad (15)$$

$$\begin{aligned}\hat{A}(k+1,n) &= P^{-1}(k+1,n)[\, P(k,n) A^T(k,n) Y(k,n) - a(k) i(k) + a(k+n) i(k+n) \,] \\ &= P^{-1}(k+1,n) P(k,n) \hat{A}(k,n) - P^{-1}(k+1,n) a(k) i(k) + P^{-1}(k+1,n) a(k+n) i(k+n) \\ &= P^{-1}(k+1,n)[\, P(k+1,n) + a(k) a^T(k) - a(k+n) a^T(k+n) \,] \hat{A}(k,n) - P^{-1}(k+1,n) a(k) i(k) + P^{-1}(k+1,n) a(k+n) i(k+n) \end{aligned} \quad (16)$$

令

$$K(k+1,n,k) = P^{-1}(k+1,n) a(k) \quad (17)$$

$$K(k+1,n,k+n) = P^{-1}(k+1,n) a(k+n) \quad (18)$$

代入(16)式得

$$\hat{A}(k+1,n) = \hat{A}(k,n) - K(k+1,n,k)[\, i(k) - a^T(k) \hat{A}(k,n) \,] + K(k+1,n,k+n)[\, i(k+n) - a^T(k+n) \hat{A}(k,n) \,] \quad (19)$$

由于上式中本次待求量的估计值是根据上次的估计值加上修正项一步计算得到,故称为一步递归算法,其中系数 $K(k+1,n,k)$、$K(k+1,n,k+n)$ 为列向量。

2.2 两步递归算法

由(14)式令 $n = n+1$,可得

$$\begin{aligned}\hat{A}(k,n+1) &= P^{-1}(k,n+1) A^T(k,n+1) Y(k,n+1) \\ &= \hat{A}(k,n) + K(k,n+1,k+n)[\, i(k+n) - a^T(k+n) \hat{A}(k,n) \,] \end{aligned} \quad (20)$$

其中

$$K(k,n+1,k+n) = P^{-1}(k,n+1) a(k+n) \quad (21)$$

其中 $\hat{A}(k,n)$ 是由 k 到 $k+n-1$ 时刻 n 个数据得到的估计值,而 $\hat{A}(k,n+1)$ 则是由 k 到 $k+n$ 时刻 $n+1$ 个数据得到的估计值。为了保持数据窗长度 n 不变,增加了 $k+n$ 时刻的新数据,就要去掉 k 时刻的老数据。相应的估计值可表示为 $\hat{A}(k+1,n)$。

$$\begin{aligned}\hat{A}(k+1,n) &= P^{-1}(k+1,n) A^T(k+1,n) Y(k+1,n) \\ &= \hat{A}(k,n+1) - K(k+1,n,k)[\, i(k) - a^T(k) \hat{A}(k,n+1) \,] \end{aligned} \quad (22)$$

式中

$$K(k+1,n,k) = P^{-1}(k+1,n) a(k) \quad (23)$$

由(20)～(23)式可写出如下两表达式

$$\begin{cases} \hat{A}(k,n+1) = \hat{A}(k,n) + K(k,n+1,k+n)[\, i(k+n) - a^T(k+n) \hat{A}(k,n) \,] \\ k(k,n+1,k+n) = P^{-1}(k,n+1) a(k+n) \end{cases} \quad (24)$$

$$\begin{cases} \hat{A}(k+1,n) = \hat{A}(k,n+1) - K(k+1,n,k)[\, i(k) - a^T(k) \hat{A}(k,n+1) \,] \\ k(k+1,n,k) = P^{-1}(k+1,n) a(k) \end{cases} \quad (25)$$

其计算步骤是:先由数据窗长度为 n 的 k 时刻的估计值 $\hat{A}(k,n)$,按(24)式计算得到数据窗长度为 $n+1$ 的 k 时刻的估计值 $\hat{A}(k,n+1)$,再由(25)式计算得到数据窗长度为 n 的 $k+1$ 时刻的估计值 $\hat{A}(k$

$+1, n)$。由于计算分两步进行故称为两步递归算法。

2.3　递归算法在保护中应用时进一步减小运算量的措施

注意,若将两步递归算法的(25)式代入(24)式,经整理同样能够得到与(19)式相似的一步递归算法,所以以下仅讨论一步递归算法。

由递增归算法(19)式可知,在计算估计值时,首先需要计算出 $K(k+1, n, k)$ 与 $K(k+1, n, k+n)$ 系数。若假定待求量为 m 个,数据窗长度为 n,则 $Y(k+1, n)$ 及其对应的 $a(k+1)$、$a(k+n+1)$、$A(k+1, n)$ 和 $P(k+1, n)$ 分别为 $n \times 1$, $m \times 1$、$m \times 1$、$n \times m$ 和 $m \times m$ 维。因此计算 $K(k+1, n, k)$、$K(k+1, n, k+n)$ 需要求 $m \times m$ 维矩阵的逆运算和 $m \times m$ 次乘法运算,计算量很大,尽管可以导出相应的递归算式,但其计算量仍然很大。为满足保护的要求有必要对系数 $K(k+1, n, k)$、$K(k+1, n, k+n)$ 做特殊处理。为讨论方便令

$Q = diag(Q_0, Q_1, \cdots, Q_m)$ 为对角分块矩阵,其中

$$Q_i = \begin{bmatrix} \cos i\varphi & +\sin i\varphi \\ -\sin i\varphi & \cos i\varphi \end{bmatrix} \tag{26}$$

式中 $i = 1, 2, \cdots, m$, $\varphi = \omega_0 \Delta T_s = \frac{2\pi}{N}$ (N 为一个周期采样次数,ΔT 为采样时间间隔)。另外当直流分量为恒定值时 $Q_0 = 1$, 而当直流分量为指数衰减时

$$Q_0 = \begin{bmatrix} 1 & -\Delta T \\ 0 & 1 \end{bmatrix}$$

注意到:$a(k+N+1) = [Q^N]^T a(k+1)$, $A(k+N+1) = A(k+1)\, Q^N$

由(17)式可得

$$\begin{aligned} K(k+1, n, k+N) &= P^{-1}(k+N+1, n)\, a(k+N) \\ &= [A^T(k+N+1, n)\, A(k+N+1, n)]^{-1} a(k+N) \\ &= [Q^N]^{-1} [A^T(k+1, n)\, A(k+1, n)]^{-1} a(k) \\ &= [Q^N]^{-1} K(k+1, n, k) \end{aligned} \tag{27}$$

对于保护算法而言,电力系统中电流波形用(1)式表达已经足够准确。除衰减的直流分量外都可看为是周期性的。为此分两种情况进行讨论。

(1) 当波形中直流分量为恒定值时

此时 $Q_0 = 1$, $Q^T = Q^{-1}$ 且 $Q^N = E$, (6)式中 A 阵不含第二列,且也具有周期性。此时(27)式变为

$$K(k+1, n, k+n) = K(k+1, n, k) \tag{28}$$

所以当波形具有周期性时,系数 $K(k+1, n, k)$、$K(k+1, n, k+n)$ 也具有周期性,为此可离线计算一个周期的 $K(k+1, n, k)$、$K(k+1, n, k+n)$ 值并存放好,实时计算时可直接取用。

(2) 当直流分量为指数衰减量时

一种办法是在估算前先进行一次差分滤波,由(2)式可知经差分滤波后可认为波形中仅含有恒定直流分量和各次谐波,因此可按情况(1)处理,但这样将使响应时间增加一个采样间隔。

另一种办法是首先离线计算 $K(k+1, n, k)$、$K(k+1, n, k+n)$ 并存放一个周期的数值以备周期重复顺序取用,其次在实时计算时每隔一个周期 $A_0 = A_0 - N\Delta T I_d$ 修正一次估计值中的恒定直流分量。

经以上处理系数 $K(k+1, n, k)$、$K(k+1, n, k+n)$ 仍然可以离线计算,并仅存放一个周期的值,从而使递归算法的运算量得到大幅度减小。

从上述理论可以看出,改进后算法具有如下特点:

(1) 改进算法实时计算量小

若假定待求量为 m 个,数据窗长度为 n,则计算 $a^T(k) A(k, n)$ 和 $a^T(k+n) A(k, n)$ 共需要 $2m$ 次乘法,由(19)式递归计算每个估计值时,还需要两次乘法。所以计算所有参量共需 $4m$ 次乘法。平均计算每个参数需 4 次乘法。而原最小二乘一次完成算法需要 n 次乘法, $n \geq 2 \times m + 2$, 当 $m = 12$ 时可知需计算 26 次乘法。所以改进后实时计算量大大缩小。

(2) 由于运算量大大减小,为缩短数据窗提供了条件。

在 CPU 和数据采集系统允许的条件下,可适当提高采样频率,并保持数据窗内采样点数不变,这样可以缩短数据窗,提高保护的响应速度。

(3) 改进算法仍然保持了原算法的滤波特性与精度

由于改进算法是在原算法基础上导出的,所以滤波特性与精度保持不变。

3　计算机仿真

仿真时有关假设

(1) 仿真模型为

$$\text{模型 1}: i(t) = I_0 + \sum_{n=1}^{5} (I_{nc} \cos n\omega_0 t + I_{ns} \sin n\omega_0 t) + e(t) \tag{29}$$

$$\text{模型 2}: i(t) = I_0 e^{-t/\tau} + \sum_{n=1}^{5} (I_{nc} \cos n\omega_0 t + I_{ns} \sin n\omega_0 t) + e(t) \tag{30}$$

(2)假定故障前直流分量和各谐波的实虚部分别为

故障前 I = [2 12 24 6 12 4 8 3 6 2.5 4.8]

(3)数据窗长度为半个工频周期,采样点数为24个。

按照上述条件进行计算分别得表1和表2。由表可知经一个数据窗长度能够正确计算出各项谐波系数。由此也说明了递归算法仍具有很好的滤波特性。

表1 对于模型1递归算法的仿真计算结果

迭代次数	算法种类	I_0	I_{1c}	I_{1s}	I_{2c}	I_{2s}	I_{3c}	I_{3s}	I_{4c}	I_{4s}	I_{5c}	I_{5s}
	真 值	9	96	192	48	96	32	64	24	48	19	38.4
10	一次完成算法	-4.6629	-7.2824	-3.6337	-3.2290	-4.5406	-0.4078	-2.8602	0.3529	-0.9440	0.1634	-0.1267
10	递归算法	-4.3454	-6.5155	-3.8471	-2.3726	-4.6041	-0.1785	-2.6770	0.5558	-0.7577	0.1851	-0.0688
25	一次完成算法	9.0000	96.0000	192.0000	48.0000	96.0000	32.0000	64.0000	24.0000	48.0000	19.2000	38.4000
25	递归算法	8.9817	95.9170	192.0142	47.9865	96.0174	31.9978	64.0110	24.0010	48.0037	19.2005	38.4005

表2 对于模型2递归算法的仿真计算结果(按照固定存贮系数项周期重复使用)

迭代次数	算法种类	I_0	I_d	I_{1c}	I_{1s}	I_{2c}	I_{2s}	I_{3c}	I_{3s}	I_{4c}	I_{4s}	I_{5c}	I_{5s}
	真 值	100	0	96	192	48	96	32	64	24	48	19	38.4
25	一次完成算法	98.4	1.6682	96.6	191.8	48.1	96	32	64	24	48	19.2	38.4
25	递归算法	98.5	1.6863	96.6	191.7	48.1	95.9	32	64	24	48	19.2	38.4
50	一次完成算法	96	1.4722	95.9	192.5	47.9	96	32	64	24	48	19.2	38.4
50	递归算法	96.2	1.4858	96	192.5	47.9	96	32	64	24	48	19.2	38.4

注:表1、表2中一次完成算法的结果为按公式(8)所得到的结果。

4 结论

通过理论分析和计算仿真,可以得出以下结论。

(1)改进后的最小二乘算法除仍然具有滤波特性好和数据窗可变的特点外,计算工作量大为减少,完全能满足微机保护对实时性的要求。

(2)改进后的最小二乘算法的计算精度较高,也能满足微机保护的要求。

参考文献:

[1] 陈德树. 计算机继电保护原理与运行. 北京:水利电力出版社,1990.

[2] [瑞典]K J 奥斯特隆姆、B 威顿马克著,李清泉等译. 自适应控制. 北京:科学出版社,1992.

[3] 李清泉. 自适应控制系统理论设计与应用. 北京:科学出版社,1990.

[4] 贺威俊,张淑琴等. 晶体管与计算机继电保护原理. 成都:西南交通大学出版社,1990.

[5] [美]IEEE 电力工程委员会、IEEE 电力系统继电保护委员会编,孙军、陶惠良等译. 微处理机式继电器和保护系统. 重庆:重庆大学出版社,1990。

Study and improvement of least squares algorithm

WANG Yi - fei

(College of Electrical Engineering SWJU, Chengdu 610031, China)

Abstract: The least squares algorithm (LSA) has nicely filter characteristic and variable data window. But the amount of counting is so large that it can not meet the real - time requirement of microprocessor - based protection. Based on thorough analyses and researches, an improved LSA is presented in this paper, which can overcome this shortcoming. Besides the original characteristics of LSA, the improved LSA greatly reduces the amount of counting so that it can be applied to the microprocessor - based protection. The result of analyses and simulations indicates that the improved LSA can be used to accurately calculate electrical parameters.

Keywords: least squares; algorithm; protection