

基于 ANN 的电力系统测频补偿算法

金明 青岛大学电气及自动化工程学院 (266071)

杨善利 青岛电业局用电处 (266001)

【摘要】 利用人工神经网络技术,提出了一种新的测频补偿算法。该方法适用于微型低频减载装置。与传统的补偿算法相比,新算法具有计算量小、精度高、对采样速度适应性强等特点,可用于在线测频计算。文中给出的算例证明了算法的有效性。

【关键词】 人工神经网络 低频减载 频率测量

引言

频率是衡量电能质量的重要指标之一,当电网发生短路故障,大型发电机组突然切除,或用电负荷突然大幅度增加时,电网频率将显著降低,致使大量用电设备不能正常运行,甚至造成电力系统崩溃,危害相当严重。目前普遍采用的措施是在某些变、配电所中装设低频减载装置,在频率下降过程中,适当切除部分用电设备,使系统功率达到新的平衡,则电网频率会逐渐恢复到接近正常值。

随着电力系统的发展,电网运行方式日益复杂和多样化,对低频减载装置性能指标的要求也日益提高。用传统频率继电器构成的低频减载装置由于级差大、级数少而在实际应用中暴露出种种不足,主要体现在不能很好地适应系统中出现的不同功率缺额,不能有效地防止系统频率的持续下降并恢复频率,常常造成频率的悬停和超调现象。随着计算机技术的飞速发展,微型计算机在变电所控制装置中的应用日益广泛,目前越来越多的变电所中采用了微型机的低频减载装置,取得了较好的控制效果。

任何一种微型机低频减载装置的核心问题都是如何根据对输入电压信号的采样信息,快速准确地计算出系统的当前频率。这就要求装

置所采用的测频算法既要有较小的工作量,同时又要有足够的计算精度。在目前提出的多种算法中, cross 改进算法由于计算量较小、计算精度较高而得到了较为广泛的应用。

1 cross 改进算法及其缺点

传统的 cross 算法是建立在线性插值基础上的,计算量较小,其计算过程如下所述:

设电压的数学表达式为:

$$u(t) = u_m \sin(\omega t - \varphi) \quad (1)$$

令电压信号的标准基波周期为 T , 则任一时刻 t 以前一个周期的电压瞬时值为:

$$u(t - T) = u_m \sin[\omega(t - T) - \varphi] \quad (2)$$

t 以前任意时刻的电压瞬时值为:

$$u(t - \tau) = u_m \sin[\omega(t - \tau) - \varphi] \quad (3)$$

若令 τ 为固定的采样周期, f_N 为标准基波频率, N 为一个标准基波内的采样次数, 则根据线性插值原理可以写出下列关系式:

$$N - T_1 = \frac{u(t) - u(t - N\tau)}{u(t - (N-1)\tau) - u(t - N\tau)} \quad (4)$$

式中 T_1 是待求的电压信号的周期。

上式表明: 当任一时刻 t 的电压值 $u(t)$ 与标准的一周期前的电压瞬时值 $u(t - N\tau)$ 相等时, 则信号的实际周期将保持标准周期不变, 否则说明信号的实际周期与标准周期有偏差, 利用线性插值的原理来估算这个偏差即得到 (4)

window length can be flexible selected as a full cycle, half a cycle or a quarter cycle. 3. The results got from the new method are more accurate than those of calculated from the Fourier method.

Keywords Wavelet analysis Impedance calculating The Fourier method

式。但线性插值方法本质上只有对线性波形才是准确的,而实际上来自电网的电压信号实质上是非线性关系,为使计算结果趋于合理,实际计算中一般采用加权平均的方法进行。

cross 算法计算信号频率的完整公式为:

$$x(m, t) = \frac{1}{2} \frac{[u(t) - u(t - N)] [D(t) - D(t - N +)]}{[D(t) D(t - N -)]} \quad (5)$$

其中 $D(t) = u(t) - u(t -)$

$x(m, t) = N - T_1$, T_1 是经过加权平均计算后得到的输入信号的周期

x 为计算所得周期偏差的系数值

最终计算信号频率的公式为:

$$f = \frac{N}{N - x} f_N \quad (6)$$

实际计算表明传统的 cross 算法当实际频率与标准频率接近时,能取得较好的精度,但当实际频率与标准频率偏差较大时,计算精度明显降低。这是因为, cross 算法是根据线性插值原理来计算频率偏差的,当实际频率与标准频率偏差较大时,这种假定的线性关系就不能成立了。解决的方法有两种,一是提高采样速度,缩小等值区间;二是改进算法,对计算误差进行补偿。前者受计算机硬件的限制,一般较少采用,文献^[1]对传统的 cross 算法进行了改进;

首先根据实际计算,得到采用 cross 算法所产生计算误差与频率偏差之间存在的关系,这种关系可用一系列的二元组表示,每个二元组的形式为 (x_i, Err_i) , 其中 x_i 为频率偏差, Err_i 为相应的计算误差,然后利用一个适当的多项式来拟合这个关系,最后再利用这个多项式对原计算公式进行补偿。这样 cross 改进算法中的频率计算公式为:

$$f = \frac{N}{N - x} f_N + (x) \quad (7)$$

其中 (x) 为修正部分。 (x) 的表达式理论上应利用曲线拟合技术获得。

实算结果表明, (x) 表达式的形式与每周

期采样次数有关。对于某些采样速度,相应的多项式次数较高,这就给计算增加了较多的负担。同时 (x) 表达式无统一的形式,当装置采样速度改变时,需重新对 (x) 的形式进行拟合,对拟合技术不熟悉的人员难以完成。

本文利用人工神经网络技术,提出了一种新的 cross 补偿算法。该方法利用 BP 网络来模拟计算误差与频率偏差之间存在的关系,网络的训练是离线完成的,训练好的网络只需将相应权值存入微机即可用于在线补偿计算,计算量小,计算精度高。当装置采样速度改变时,只需利用新的 (x_i, Err_i) 关系对原 BP 网络进行重新训练,而不需改变网络的结构,同时训练工作简单方便,只需启动相应的训练程序即可。

2 BP 网络及其学习算法

本文中所采用的 BP 网络模型结构如图 1 所示。该网络由输入层、输出层和隐含层所组成。输入层节点与隐含层节点、隐含层节点与输出层节点之间充分联接,同一层节点之间互不关联。当信号输入时,首先传到隐含层节点,经过作用函数后,再把隐含层节点的输出信号传到输出层节点,经过处理后给出输出结果。

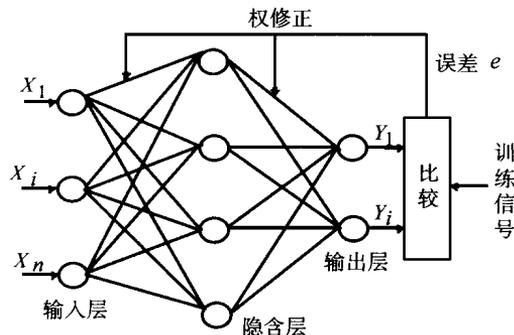


图 1 BP 网络示意图

可以证明,图 1 所示的三层 BP 网络可以映射任何复杂的非线性关系。

图 1 中同时给出了 BP 网络训练过程的原理图。训练过程由正向传播和反向传播两个过程组成。在正向传播过程中,输入信号从输入层经隐含层单元处理,并传向输出层,每一层神经元的状态只影响下一层神经元的状态。如果在

输出层不能得到期望的输出,则转入反向传播,将输出信号的误差沿原来的连接通路返回,并修改各层神经元的权值,这两个过程反复多次,直到误差达到精度要求。BP网络一旦训练结束,对于新的输入只需经正向传播即可得到相应的输出。

本文中所采用的BP网络,输入层节点数为1,相应于利用传统cross算法所得的频率偏差 f ;输出层节点数为1,相应于需补偿的误差修正量。隐含层单元数的选择是一个十分复杂的问题,目前没有理想的解析式来表示。隐含层单元数的多少与多种因素有关,一般来说,隐含层单元数太少可能导致网络训练的效果不好,使计算精度变差;隐含层单元数太多又使学习的时间过长,误差也不一定最小。本文中采用近来提出的BP网络结构自整定算法^[2]动态选取隐含层节点个数,大大改善了学习效率和网络的适应能力。

设第 n 层第 i 个节点的输出为 O_{ni} , W_{nij} 表示该节点与前一层第 j 个节点的联接权值, M_n 为第 n 层的节点数, $f(x)$ 为作用函数,则正向传播过程的计算公式为:

$$net_{ni} = \sum_{j=1}^{M_{n-1}} W_{nij} O_{(n-1)j} \quad (8)$$

$$O_{ni} = f(net_{ni}) \quad (9)$$

$f(x)$ 采用sigmoid型,即

$$f(x) = \frac{1 - e^{-x}}{1 + e^{-x}} \quad (10)$$

BP网络的学习算法为:

对输出层各节点

$$n_i = f(net_{ni})(O_i^* - O_{ni}) \quad (11)$$

$$W_{nij} = n_{ni} O_{(n-1)j} \quad (12)$$

$$W_{nij, NEW} = W_{nij, OLD} + W_{nij} \quad (13)$$

其中 O_i^* 为第 i 个节点的期望输出, n 为学习率。

对隐含层各节点, (11)式可写为

$$n_i = f(net_{ni}) \sum_{k=1}^{M_{n+1}} (n+1)_k W_{(n+1)ki} \quad (14)$$

训练误差指标为平方型误差:

$$E = \sum_{i=1}^{M_n} (O_i^* - O_{ni})^2 \quad (15)$$

其中 M_n 为输出层节点数。

3 补偿算法的实现

本文提出的基于BP网络的测频补偿算法的实现,由以下两步构成:

如前所述,我们可以将采用传统cross算法所产生计算误差与频率偏差之间的关系用一系列的二元组 (x_i, Err_i) 表示,每个二元组组成一个样本对,将所有的样本对作为训练样本集。要获得较好的训练效果,样本集的组织是至关重要的。样本集的组织一般应遵循以下两条原则:(1)应使训练样本集中样本对的数量足够多;(2)应使样本点均匀分布在整个可能的频率偏差范围内。样本集中各样本对可利用(5)(6)式简单地通过离线计算而获得。

然后将样本集中的各个样本对依次送入BP网络的学习机制中进行训练。训练过程结束后,只需将相应权值存入微机即可用于在线补偿计算。

当装置采样速度改变时,而要求改变补偿特性时,只需启动网络训练程序,送入新的 (x_i, Err_i) 所组成的样本集,待训练过程完成后,即可获得新的补偿特性。

4 数字试验与结果

为便于比较,我们采用了和文献^[1]中同样的采样速度对上文中提出的方法进行了试算,结果令人满意,现将计算结果列表如下,计算过程取 $N = 24$, $m = N$, $f_N = 50\text{Hz}$ 。

需说明的是,上表中所有频率为整数的试验点都是作为样本处理的,因此这些点对应的误差极小。

5 结论

1. 本文提出的利用人工神经网络技术构成的测频补偿算法,经实算证明具有计算量小,计算精度高等特点,适用于微机型的低频减载装置。

表 1 计算结果

试验频率	计算结果	计算误差
50.000	50.000000	0.000000
49.500	49.500981	0.000981
49.000	48.999988	- 0.000012
48.500	48.500745	0.000745
48.000	48.000013	0.000013
47.500	47.499479	- 0.000521
47.000	47.000039	0.000039
46.500	46.500669	0.000669
46.000	45.999969	- 0.000031
45.500	45.500876	0.000876
45.000	44.999981	0.000019

2. 当装置采样速度改变时, 只需利用新的 (x_i , Err_i) 关系对原 BP 网络进行重新训练, 而不需改变网络的结构, 同时训练工作简单方便,

只需启动相应的训练程序即可。

参考文献

- 1 胡艳婷等. 一种供安全自动装置用的新的频率测量方法. 电力系统自动化, 1987, (6).
- 2 贺江峰等. 一种 BP 网络结构和学习参数自整定的学习方法. 第二届中国人工智能联合学术会议论文集. 清华大学出版社, 1994. 10.
- 3 史忠植. 神经计算. 北京: 电子工业出版社, 1993. 11.

1998—04—06 收稿

金明, 男, 1965 年生, 硕士, 从事电力系统研究工作。

A NEW COMPENSATION AL GRITHM FOR FREQUENCY MEASUREMENT BASED ON ANN

Jin Ming (The College of Electrical and Automation of Qingdao University, Qingdao 266071)

Yang Shanli (Qingdao Electrical Power Bureau, Qingdao 266001)

Abstract Based on Ann, a new kind of compensation algorithm used for frequency measurement is developed in this paper. The new method is available for its less calculation quantity, faster computation speed and the adaptation to the sampling speed of the device. It can be used in microcomputer UFLS (Under Frequency Load shedding) devices as frequency measurement component. Some examples are used to test the new algorithm and the results show the algorithm performs well.

Key words ANN UFLS Frequency measurement