Iterative Learning Control of Power Flow Calculation

Yinjun Zhang[1,2,*] and Jianhuan Su[2]

[1]*Aeronautics and Astronautics Engineering Institute, Air Force Engineering University, Xi'an 710038, Shanxi Province, China*

[2]*School of Physics and Electrical Engineering Hechi University Yizhou GuangXi 546300, China*

Abstract: The paper analyzes the flow calculation of power system, using iterative learning algorithm to calculation the power flow, compare with traditional improving Newton etc. algorithm, Iterative learning algorithm has fast convergence can also be to achieve a high precision tracking. In this paper convergence of the algorithm is global, and gives control of the convergence conditions and rigorous theoretical proof.

Keywords: Flow calculation, iterative learning control, power system.

1. INTRODUCTION

The power system flow calculation is a basic power calculation for researching power system steady state. Its task is ensure the operational status of the whole system according to the operation conditions and network structure, such as the voltage on each bus(amplitude and phase),power distribution and power loss etc. in the network. The results of power flow calculation are the basis for computation and fault analysis of power system stability [1].

Nearly 20 years, the study power flow algorithm is still very active, but most research revolves around improving the Newton method and the P-Q decomposition method performed. In addition, with the development of artificial intelligence theory, genetic algorithms, artificial neural networks, fuzzy algorithm is introduced gradually flow calculation [2-4]. However, so far these new models and algorithms cannot replace Newton's method and P-Q decomposition method [5, 6] position. Due to the size of the system [7] continues to expand the power of computing speed increasing demands, parallel computing technology computer will also be widely used in the flow calculation, and has become an important area of research.

2. PROBLEM DESCRIPTION

According to the expression of the power equation (1):

$$P_i - jQ_i = V_i \sum_{j=1}^{n} Y_{ij}^* V_j^* \tag{1}$$

Where Pi, Qi respectively are active and reactive power, Vi is the voltage the node i, $V_j^*$is the j-th node voltage conjugate, Y_{ij}^* is the admittance matrix conjugate, change the express form, we get equation (2).

$$\frac{P_i - jQ_i}{V_i^*} = \sum_{j=1}^{n} Y_{ij}^* V_j^* \tag{2}$$

Expand equation (2), then get equation (3)(4). (3)

$$\frac{P_i - jQ_i}{V_i^*} = \sum_{j=1}^{n} Y_{ij}\ V_j \ \ + Y_{ii}\ V_i \tag{3}$$

$$V_i = \frac{\frac{P_i - jQ_i}{V_i^*} - \sum_{\substack{j=1 \\ j\neq i}}^{n} Y_{ij}\, V_j}{Y_{ii}} \tag{4}$$

Then we get iterating learning algorithms basis equation(5)

$$V_i^{(k+1)} = \frac{1}{Y_{ii}}\left[\frac{P_i - jQ_i}{V_i^{*(k)}} - \sum_{j=1}^{n} Y_{ij}\ V_j^{k+1} - \sum_{j=1}^{n} Y_{ij}\ V_j^{k}\right] \tag{5}$$

According to equation (5), next, we will discuss the PV, PQ and balance node respectively. The PQ node is injected into the power of P and Q is given, applicable to a given active and reactive power of the generators and no substation power conditioning equipment node. For such a node, will take directly the last voltage values into equation (5) can be calculated by iteration.

PV node that is active power P and voltage U is given for the generator nodes equipped with phase modulation substation node, due to its reactive power Qi is unknown, it must be calculated first.

According to the reactive power and current expression:

$$Q_i^{(k)} = x_0 \mathrm{Im}(U_i^{(k)} I_i^{(k)}), \tag{5-1}$$

$$I_i^{*(k)} = \sum_{j-1}^{i-1} Y_{ij}^* U_j^{*(k+1)} + \sum_{j=1}^{n} Y_{ij}^* U_j^{*(k)} \tag{5-2}$$

take equation (5-1) into equation (5-2), then get equation (5-3).

$$Q_i^{(k)} = \mathrm{Im}[U_i^{(k)}(\sum_{j=1}^{i-1} Y_{ij}^* U_j^{*(k+1)} + \sum_{j=i}^{n} Y_{ij}^* U_j^{*(k)})] \tag{5-3}$$

1874-1290/15 2015 Bentham Open

Then the equation (5-3) substituted equation (5) can be obtained $V_i^{(k+1)}$, due to the voltage of PV node is given, so on the basis of $V_i^{(k+1)} = U_i e^{j0}$ we can get node phase angle.

Next, we just according to equation (5-4), then using the third part iterative learn algorithm to prove its uniform convergence.

$$\left|U^{(k+1)} - U_i^{(k)}\right| \le \varepsilon. \tag{5-4}$$

3. ITERATING LEARNING ALGORITHMS

Given power flow is non-liner system (6)

$$\begin{cases} \dot{x}(t) = f(x,u,t) \\ y(t) = C(x,t) + B(t)u(t) \end{cases} \tag{6}$$

and its initial condition, where $x \in R^n, u \in R^m, y \in R^l, B \in R^{m.L}$,f and C are the vector function of corresponding dimension and Lipschiz continuous, shown the equations (6)(7).

$$\|f(x_1,u_1,t) - f(x_2,u_2,t)\| \le L_f(\|x_1 - x_2\| + \|u_1 - u_2\|) \tag{7}$$

$$\|C(x_1,t) - f(x_2,t)\| \le L_c\|x_1 - x_2\| \tag{8}$$

L_f and L_c does not know.

For a given target output $y_d(t)$, to find the control input $u_k(t)$, such that it corresponds to the output $y_k(t)$ of the system meet B with $t \in [0,T]$.

$B = e_k(t) = y_k(t) - y_d(t) \to 0, k \to \infty$ and input control $u_k(t)$ can ensure according to learning. In the discussion that follows, we will use the following norm and lemma:

Functions $g_k(t)$:(λ,ξ) norm of range [0,T] $\to R^n$ is

$$\|g_k\|_{\lambda(\xi)} = \sup_{0\le t\le T}\{(\|g_k(t)\|e^{-\lambda t})\xi^k\}$$

Lemma 1 If there is a constant $M > 0, \xi > 1$, let $\|u_{k+1}(t) - u_k(t)\| \le M\xi^{-k}, \forall t \in [0,T]$ then sequence $\{u_k(t)\}$ convergence within range [0,T].

Proof. We just prove $\{u_k(t)\}$ is Cauchy sequence within [0,T].

For any natural number p and k, we can get

$$\|u_{k+p}(t) - u_k(t)\| = \left\|\sum_{j=k}^{k+p}\Delta u_j(t)\right\| \le \sum_{j=k}^{k+p}\|\Delta u_j(t)\| \le \sum_{j=k}^{k+p} M\xi^{-k}$$

$$= M\xi^{-k}(1 + \xi^{-1} + \cdots \xi^{-p}) \le \frac{M\xi^{-k}}{1-\xi^{-1}} = M^*\xi^{-k} \tag{9}$$

Where $M^* = \frac{M}{1-\xi^{-1}}$, as $\xi > 1$

so according to equation (9), we can get $\{u_k(t)\}$ is uniform convergence within [0,T].

Lemma 2 If $(\upsilon,\rho) \in (0,1)^2$, and $\frac{a\rho^N}{1-\rho} \in (0,1)$, then for any number of normal b, c, there is $(\xi,\eta) \in \Omega$ that makes the $(\upsilon\xi, f(\xi,\eta)) \in (0,1)^2$, where $\Omega = \{(x,y)|x > 1, y > 0\}$,

$$f(x,y) = \frac{(\rho x)^N}{1-\rho x}[x^{N+1}(a + cy) + bxy]$$

Proof Because $\nu \in (0,1)$, then exists $\xi_1 > 1$,so that the $\nu\xi \in (0,1)$ for $\xi \in (0,1)$, we notice the equation(10) and $f(x,y)$ are continuous within $(x_0,y_0) = (0,1)$, there is $(\xi_2,\eta_1) \in \Omega$ that makes the equation(11) comes into existence and $f(\xi,\eta) \in (0,1)$. Then we take $\xi_3 = \min\{\xi_1,\xi_2\}$, so when equation (12) was founded, we should have $(\nu\xi, f(\xi,\eta)) \in (0,1)^2$. That lemma2 was established by $\Omega_1 \subset \Omega$.

$$f(1,0) = \frac{(a\rho)^N}{1-\rho} \in (0,1) \tag{10}$$

$$(\xi,\eta) \in \Omega_0 = \{(x,y)|1 < x < \xi_2, 0 < y < \eta_1\} \tag{11}$$

$$(\xi,\eta) \in \Omega_1 = \{(x,y)|1 < x < \xi_3, 0 < y < \eta_1\} \tag{12}$$

4. TRANSFORMATIONS OF ALGORITHM AND PROBLEM

For the system (6), our learning control algorithm is equation (13-a, 13-b, 13-c).

$$u_{k+1}(t) = u_{ka}(t) + u_{kb}(t), k = 1,2 \ldots \tag{13-a}$$

Where

$$u_{ka}(t) = \Gamma(t)u_k(t) + D(t)e_k(t), k = 1,2 \ldots \tag{13-b}$$

$$u_{kb}(t)\begin{cases}\sum_{i=1}^{N}\Gamma_i(t)u_{k-i}(t), k = N+1, N+2, \ldots \\ 0, k = 1,2,\ldots,N\end{cases} \tag{13-c}$$

$\sum_{i=1}^{N}\Gamma_i(t) = I - \Gamma(t), e_k(t) = y_k(t) - y_d(t), y_k(t) = C(x_k(t),t) + B(t)u_k(t)$.

We still suppose the system through the system initial point given corresponding to each state $x_k(t)$, namely $x_k(0) = x_0(k = 1.2 \ldots.)$, or $\|x_{k+1}(t) - x_k(t)\| \le lv^k, v \in (0,1)$.

The purpose is to determine the learning coefficient of the algorithm $\Gamma(t), \Gamma_i(t), D(t)$ that makes

Algorithm converges, namely

$u_k(t) \to u_d(t), k \to \infty, t \in [0,T]$;

2) Goal tracking, namely

$y_k(t) \to y_d(t), k \to \infty, t \in [0,T]$.

For the problem 1) and 2), we will be transformed into an asymptotic stability of the system, as we consider the convergence of $\{u_k(t)\}, \{e_k(t)\}$ on k, so only consider the case of k>N.

Lemma 3 marked $\Delta u_k(t) = u_k(t) - u_{k-1}(t)$ and

$$A(t) = \begin{pmatrix} I + B(t)D(t) & B(t)\Gamma(t) - I \\ D(t) & \Gamma(t) - I \end{pmatrix},$$

$Q(t) = \begin{pmatrix} e_k(t) \\ \Delta u_k(t) \end{pmatrix} F_k(t) =$
$\begin{pmatrix} C(x_{k+1}(t),t) - C(x_k(t),t) - B(t)\sum_{i=1}^{N}\Gamma_i(t)\sum_{j=1}^{i-1}\Delta u_{k-j}(t) \\ -\sum_{i=1}^{N}\Gamma_i(t)\sum_{j=1}^{i-1}\Delta u_{k-j}(t) \end{pmatrix}$

Then the equation (14) is valid.

$$Q_{k+1}(t) = A(t)Q_k(t) + F_k(t), k \geq N. \tag{14}$$

Proof From algorithm (13), we know

$$u_{k+1}(t) = \Gamma(t)u_k(t) + D(t)e_k(t) + \sum_{i=1}^{N}\Gamma_i(t)u_{k-i}(t) \text{ for } k \geq N. \tag{15}$$

From $\sum_{i=1}^{N}\Gamma_i(t) = I - \Gamma(t)$, we can get $u_{k+1}(t) - u_k(t) = (\Gamma(t) - I)u_k(t) + \sum_{i=1}^{N}\Gamma_i(t)u_{k-i}(t)\, D(t)e_k(t)$

$\sum_{i=1}^{N}\Gamma_i(t)u_{k-i}(t)\, D(t)e_k(t) = (\Gamma(t) - I)\big(u_k(t) - u_{k-1}(t)\big) + \sum_{i=1}^{N}\Gamma_i(t)(u_{k-i}(t) - u_{k-i-1}(t))$

$+(\Gamma(t) - I)u_{k-1}(t) + \sum_{i=1}^{N}\Gamma_i(t)u_{k-i}(t) + D(t)e_k(t)$

$= (\Gamma(t) - I)\big(u_k(t) - u_{k-1}(t)\big) + \sum_{i=1}^{N}\Gamma_i(t)(u_{k-i}(t) - u_{k-i-1}(t)) + \sum_{i=1}^{N}\Gamma_i(t)(u_{k-i}(t) - u_{k-i-1}(t))$

$-\sum_{i=1}^{N}\Gamma_i(t)(u_{k-1}(t) - u_{k-i-1}(t)) + D(t)e_k(t) = (\Gamma(t) - I)\big(u_k(t) - u_{k-1}(t)\big)$

$$-\sum_{i=1}^{N}\Gamma_i(t)(u_{k-1}(t) - u_{k-i}(t)) + D(t)e_k(t) = (\Gamma(t) - I)\Delta u_k(t) - \sum_{i=1}^{N}\Gamma_i(t)\Delta u_{k-j}(t) + D(t)e_k(t). \tag{16}$$

Otherwise

$e_{k+1}(t) = e_k(t) + y_{k+1}(t) - y_k(t) = e_k(t) + C(x_{k+1}(t),t) - C(x_k(t),t) + B(t)(u_{k+1}(t) - u_k(t))$

$= (I + B(t)D(t))e_k(t) + C(x_{k+1}(t),t) - C(x_k(t),t)$

$+B(t)(\Gamma(t) - I)\Delta u_k(t) - B(t)\sum_{i=1}^{N}\Gamma_i(t) + \sum_{j=1}^{i-1}\Delta u_{k-j}(t)$

From equation (16) and (17), we know

$$\begin{pmatrix} e_{k+1}(t) \\ \Delta u_{k+1}(t) \end{pmatrix} = \begin{pmatrix} I + B(t)D(t) & B(t)\Gamma(t) - I \\ D(t) & \Gamma(t) - I \end{pmatrix}\begin{pmatrix} e_k(t) \\ \Delta u_k(t) \end{pmatrix} + \begin{pmatrix} C(x_{k+1}(t),t) - C(x_k(t),t) - B(t)\sum_{i=1}^{N}\Gamma_i(t)\sum_{j=1}^{i-1}\Delta u_{k-j}(t) \\ -\sum_{i=1}^{N}\Gamma_i(t)\sum_{j=1}^{i-1}\Delta u_{k-j}(t) \end{pmatrix} \tag{17}$$

Namely $Q_{k+1}(t) = A(t)Q_k(t) + F_k(t), t \geq N$.

5. STABILITY ANALYSIS

Seen from section 4, to make sure the algorithm convergence and target tracking, we just proof the following condition

1) $\lim_{k\to\infty}\|Q_k(t)\| = 0, t \in [0,T]$;

2) $\{u_k(t)\}$ is uniform convergence in the range [0,T].

The problem is further transformed, first given as lemma3:

Lemma 3 If there is M>0 and $\xi > 1$ that makes $\|Q_k(t)\| \leq M\xi^{-k}, \forall t \in [0,T]$,

Then will meet condition 1) and 2).

Proof Under the condition of lemma 3, 1) is obviously satisfied, in addition, there is

$\|\Delta u_k(t)\| \leq \|Q_k(t)\| \leq M\xi^{-k}, \forall t \in [0,T]$, Known by Lemma 3, 2) is satisfied too.

Thus seen by Lemma 3, to make the algorithm convergence and target tracking, we just prove the following facts: there exists a constant M>0 and $\xi > 1$ that makes $\|Q_k(t)\| \leq M\xi^{-k}, \forall t \in [0,T]$ (18)

We note $Q_k(t)$ satisfying the equation (14), then

$Q_k(t) = A^{k-N}(t)Q_N(t) + \sum_{i=N}^{k-1}A^{k-i-1}(t)\,F_i(t), k > N, \sum_{i=N}^{k-1}A^{k-i-1}(t)\,F_i(t), k > N,$

thus

$\|Q_k(t)\| = \|A^{k-N}(t)\|\|Q_N(t)\| + \sum_{i=N}^{k-1}\|A^{k-i-1}(t)\|\,\|F_i(t)\|$

$= \left(\|A^{k-N}(t)\|^{\frac{1}{k-N}}\right)^{k-N}\|Q_N(t)\| +$

$\sum_{i=N}^{k-1}\left(\|A^{k-i-1}(t)\|^{\frac{1}{k-i-1}}\right)^{k-i-1}\|F_i(t)\|$

$$\leq \big(\rho(A(t))\big)^{k-N}\|Q_N(t)\| + \sum_{i=N}^{k-1}\big(\rho(A(t))\big)^{k-i-1}\|F_i(t)\| \tag{19}$$

Next, first of all, we estimate $\|F_i(t)\|$, then defined in the Lemma 1, $\|F_i(t)\|$ satisfies the following estimate:

$$\|F_k\|_{\lambda(\xi)} \leq \sum_{i=1}^{N}[\|B(t)\Gamma_i(t)\| + \|\Gamma_i(t)\| + \frac{LcL_f}{\lambda(\lambda - L_f)}(\|D(t)\| + \|\Gamma(t) - I\|)] + \frac{LcL_f}{\lambda(\lambda - L_f)}(\|D(t)\| + +\|\Gamma(t) - I\|)]\sum_{j=1}^{i-1}\xi^i\,\|Q_{k-j}\|_{\lambda\xi}\frac{LcL_f}{(\lambda - L_f)}(\|D(t)\| + \|\Gamma(t) - I\|)\|Q_k\|_{\lambda(\xi)}. \tag{20}$$

Next we prove the equation (20).

Proof According to F_k definition we know,

$\|F_k(t)\| \leq \sum_{i=1}^{N}\|B(t)\Gamma_i(t)\|\sum_{j=1}^{i-1}\|\Delta u_{k-j}(t)\| + Lc\|x_{k+1}(t) - x_k(t)\| + \sum_{i=1}^{N}\|\Gamma_i(t)\|\sum_{j=1}^{i-1}\|\Delta u_{k-j}(t)\|$

$$= \sum_{i=1}^{N}[\|B(t)\Gamma_i(t)\| + \|\Gamma_i(t)\|\sum_{j=1}^{i-1}\|\Delta u_{k-j}(t)\| + Lc\|x_{k+1}(t) - x_k(t)\| \tag{21}$$

as $x_k(0) = x_0(k = 1,2 \ldots,$ 则

$$\|x_{k+1}(t) - x_k(t)\| = \left\|x_{k+1}(0) \pm x_k(0)\int_0^t(f(x_{k+1}(s) - f(x_k(s))ds\right\| \leq \int_0^t L_f(\|x_{k+1}(s) - x_k(s)\| + \|u_{k+1}(s) - u_k(s)\|)ds \tag{22}$$

Table 1. Iterative data (U2=2, θ_3=0).

k	U2	θ_3	Max error
1	0.9840-0.0130i	0.0525	1.0161
2	0.9671-0.0260i	0.0582	0.0213
3	0.9662-0.0260i	0.0587	9.2414e-004
4	0.9661-0.0260i	0.0587	4.4455e-005
5	0.9661-0.0260i	0.0587	3.7054e-007
6	0.9661-0.0260i	0.0587	3.0879e-007

Table 2. Iterative data (U2=1, $\theta_3 = 5\pi/12$).

k	U2	θ_3	Max error
1	0.9680-0.0260i	0.3411	1.0236
2	0.9662-0.0260i	0.0622	0.3057
3	0.9661-0.0260i	0.0590	0.0036
4	0.9661-0.0260i	0.0587	2.9161e-004
5	0.9661-0.0260i	0.0587	2.4268e-005
6	0.9661-0.0260i	0.0587	2.0221e-006
7	0.9661-0.0260i	0.0587	1.6851e-007

Table 3. Iterative data (U2=2, $\theta_3 = \pi/12$).

k	U2	θ_3	Max error
1	0.9936-0.0052i	0.0639	1.0064
2	0.9676-0.0260i	0.0591	0.0333
3	0.9662-0.0260i	0.0587	0.0014
4	0.9661-0.0260i	0.0587	6.3046e-005
5	0.9661-0.0260i	0.0587	2.9431e-007
6	0.9661-0.0260i	0.0587	2.4525e-006

By the well-known Bellman-Gronwall inequality has

$$\|x_{k+1}(t) - x_k(t)\| \le \int_0^t L_f e^{L_{f(t-s)}} \int_0^s \|u_{k+1}(\tau) - u_k(\tau)\| d\tau ds \quad (23)$$

thus

$$\left(\|x_{k+1}(t) - x_k(t)\|\xi^k\right)e^{-\lambda t} \le \int_0^t L_f e^{-(\lambda - L_f)(t-s)} \int_0^s \left(\|u_{k+1}(\tau) - u_k(\tau)\|\xi^k\right)e^{-\lambda\tau} d\tau ds \le \frac{L_f}{\lambda(\lambda - L_f)}\|\Delta u_{k+1}\|\lambda(\xi) \quad (24)$$

By the equation (24) shows that

$$\|x_{k+1}(t) - x_k(t)\|_{\lambda(\xi)} \le \frac{L_f}{\lambda(\lambda - L_f)}\|\Delta u_{k+1}\|_{\lambda(\xi)} \quad (25)$$

By the equation (19) shows that

$$\|\Delta u_k(t)\| \le \|D(t)\|\|e_k(t)\| + \|\Gamma(t) - I\|\|\Delta u_k(t)\| + \sum_{i=1}^{N}\|\Gamma_i(t)\| \sum_{j=1}^{i-1}\left\|\Delta u_{k-j}(t)\right\|$$

The equations (25)(26) substituting into equation (21). We get

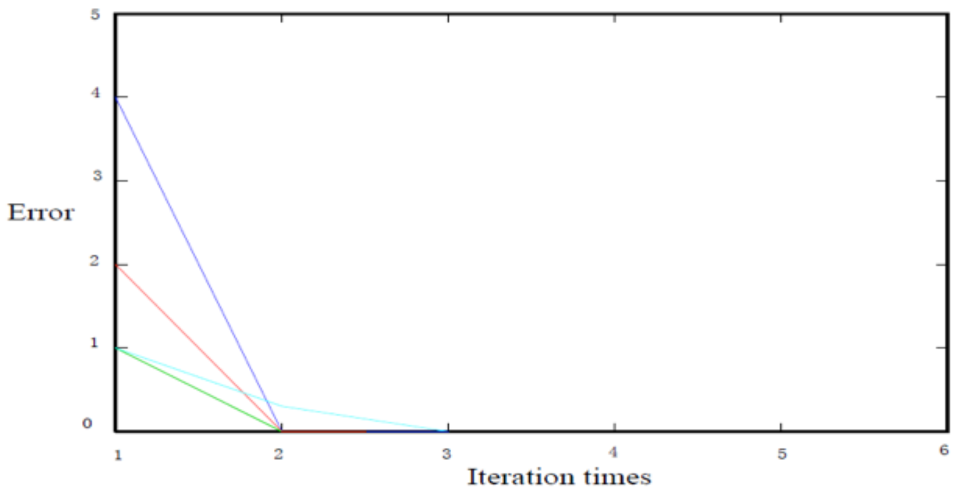

Fig. (1). Relationship of iteration times and error.

Table 4. Iterative data (U2=2, $\theta_3 = \pi/6$).

k	U2	θ_3	Max Error
1	0.9893-0.0087i	0.0618	2.0107
2	0.9674-0.0260i	0.0589	0.0280
3	0.9662-0.0260i	0.0587	0.0012
4	0.9661-0.0260i	0.0587	5.3215e-005
5	0.9661-0.0260i	0.0587	2.3516e-006
6	0.9661-0.0260i	0.0587	1.1645e-007

$$\|F_k(t)\|_{\lambda(\xi)} \le \sum_{i=1}^{N}\left(\|B(t)\Gamma_i(t)\| + \|\Gamma_i(t)\| + \frac{LcL_f}{\lambda(\lambda - L_f)}\|\Gamma_i(t)\|\right)\sum_{j=1}^{i-1}\xi^j \left\|Q_{k-j}\right\|_{\lambda(\xi)}$$
$$+\frac{LcL_f}{\lambda(\lambda - L_f)}(\|D(t)\| + \|\Gamma(t) - I\|)\|Q_k\|_{\lambda(\xi)}.$$

This is the result we want to prove.

Once $F_i(t)$ is estimated, we substituted into the equation (19) is the main result.

6. ANALYSIS OF THE COMPUTATION RESULT

We take the difference initial value of U_2 and θ_3 respectively, then analysis the variation of physical quantity in the iteration process.

According to from Tables **1-4**, we can describe the Fig. (**1**) by Matlab.

CONCLUSION

The paper analysis the power flow computing method by iteration learning algorithm. Result shown iteration learning algorithm is simple and easy programming.

CONFLICT OF INTEREST

The authors confirm that this article content has no conflict of interest.

ACKNOWLEDGEMENTS

Declared none.

REFERENCES

[1] N. G. Hingorani, and L. Gyugyi, *Combined Compensators: Umfied Power Flow Controller (UPFC) and Interline Power Flow Controller (IPFC), in Understanding FACTS: Concepts and Technology of Flexible AC Transmission Systems*, John Wiley & Sons, Inc., Hoboken, NJ, USA, 1999.

[2] M. Zhang, "Power flow calculation method," *Chinese New Technologies and Products*, vol. 12, pp. 15-16, 2013.

[3] X. Zheng, Y. Li, C. Fu, and H. Chen, "Iteration Algorithms in Newton power flow in: the initial value, fuzzy algorithm and some Jacobi," *Science and Technology Communication*, vol. 12, pp. 119-117, 2013.

[4] W. Liu, L. Cai, P. Xu, Y. Qing, J. Wang, and W. Wang, "Power system critical line identification based on Flow the number of referrals," *Chinese CSEE*, vol. 31, pp. 90-98, 2013.

[5] Z. Wu, *Consider UPFC Power System Optimal Power Flow Study*, Shenyang University of Technology, 2013.

[6] F. Liu, *Research on Cooperation Problem of Online Steady-state Power Flow Calculation with Real-time Measurements*, Huabei Power University, 2013.

[7] T. Z. Zhang, *Improved Genetic Algorithm Based on Ship Power System Reactive Power Optimization Study*, Harbin Engineering University, 2013.

Received: October 16, 2014 Revised: December 23, 2014 Accepted: December 31, 2014