IMPACT: International Journal of Research in Engineering & Technology (IMPACT: IJRET)
ISSN 2321-8843
Vol. 1, Issue 5, Oct 2013, 47-56

MODIFIED ELECTRIC POWER SYSTEM STATE ESTIMATION – *MULTI- PROCESSING TECHNIQUE*

H. NAGARAJA UDUPA[1] & H. RAVISHANKAR KAMATH[2]

[1]Research Scholar, Mewar University, Mewar, Rajastan, India

[2]Department of E & E, Malwa Institute of Technology, Indore, Madhya Pradesh, India

ABSTRACT

Estimating the state of the power system in Real time is a challenging task, mainly because, solving the non-linear large power system equations by iterative solution comsumes huge computational time and memory. Online Energy Management System (EMS) these state estimators need to Estimate the state of the system at very short intervals. Conventional methods like Newton-Raphson is not suitable for such real time application.

To improve the performance of SE, certain techniques like AI, Real time, Hirarchial and Dynamic methods are listed in reference [1][4][5][11][12][13]. The most common "Two-Level" SE technique is applied by spliting the large power system in to sub-networks and each sub-networks are managed by the local control station in coordination with the Centre station – [2][3][7][8][9][14]. Even though it reduces the computational time, still it is not suitable for real time applications. In the "Two-Level SE" mainly same NR method is used, however, before applying the NR technique, the electric network is physically divided, which results to mathematical approximation.

This paper presents an approach to solve the problem in hand without physically dividing the network and/or any approximation on the existing NR method. Single large mathematical problem can be divided into smaller independent parallel task by changing the multiplication method of large sparce jacobian matrix. This leads to a whole new approach resulting in reducing in computing time as well as dynamic memory requirement.

KEYWORDS: **SE**-State Estimation, **WLS**-Weight Least Square, **RTU**-Remote Terminal Unit, **NR**-Newton Rapson, **AI**-Artificial Intelligence, **Node Area**-A Node along with its Connected Node is Referred as Node Area

INTRODUCTION

The power system is highly distributed in nature, Instead of dividing the network in to sub networks, it is better to divide the large single problem into small independent sub tasks so that multiple processor can be used for computation. Major challenge here to obtaing the independent sub-task. The NR method mathematical solution is procedure oriented iterative technique which involvs large number of non-linear simultaneous equations. The ideal solution would have been the vertical division of the probem into many sub-independent parallel task. By modifying the existing Newton-Raphson state estimation computational procedure requires the clear understanding of the procedural steps involed in the existing NR method. The brief insight of NR method is given below.

NEWTON RAPSON (WLS) METHOD FOR STATE ESTIMATION –BRIEF REVIEW

The Newton Rapson final equation is $A\ \Delta x = b$ (taking weighting matrix W which is a diagonal matrix W_{ii} is the standard deviation of meter i.) By applying the tylor series to the nonlinear equations of power syatem following equations

are derived. Here, the error vector includes nonlinear vector function f(x). In order to estimate x, an initial value x_0 is assumed and after every iteration x_0 will be up-dated till the convergence.

$(J_0^T W J)\ \Delta x = J_0^T W\ \Delta z$

Let $A= (J_0^T W J)$ & $b = J_0^T W\ \Delta z$

$A\ \Delta x = b$ (1)

This is a set of linear equations, if higher order terms of the taylor expansion of f(x) were really negligible, the solution yield the correct 'x'. The jacobian J is itself a function of x. The state variable vector x can be obtained by solving the equation $A*\Delta x = b$ iteratively. The vector x should therefore be changed accordingly after every iteration till the convergence is obtained.

$x^{n+1} = x^n + \Delta x^1$ (1^{st} iteration count) until convergence is reached. Weighted least squares (WLS) minimization technique is used.

No of state variables = (2*n-1)

Total no of measurements = m

These measurements may include one or all quantities such as [P_i ,Q_i ,p_{ij} ,q_{ij}, V_i ,δ_i]

- P_i ,Q_i = Real & Imaginary part of injected power respectively
- p_{ij} ,q_{ij} = Real & imaginary part of line follows respectively
- V_i & δ_i = Voltage magnitude & phase angles.
- Dimension of Jacobian matrix = m * (2n - 1)

$\Delta z = z^{measured} - z^{calculated}$

'W' is the diagonal weigh matrix of the order of (m*m)

$x = \lfloor \delta_1, \delta_2, ... \delta_{n-1}, v_1, v_2, ... v_n, \rfloor$; state variables.

n= number of network nodes, 1,2,…n.

Elements of jacobian are derived from injected power and line flow equations.

$$\begin{bmatrix} \Delta P_i \\ \Delta Q_i \\ \Delta p_{ij} \\ \Delta q_{ij} \\ \Delta v_i \\ \Delta \partial_i \end{bmatrix} = \begin{bmatrix} |H_1| & |H_2| \\ |H_3| & |H_4| \\ |H_5| & |H_6| \\ |H_7| & |H_8| \\ |0| & |1| \\ |1| & |0| \end{bmatrix} \begin{bmatrix} \Delta \delta_i \\ \Delta v_i \end{bmatrix} \quad (2)$$

PROPOSED ALGORITHUM

Proposed New Algorithum

The following modification of forming the jacobian results in independent parallel task and thereby multiple processors can be used to carry out the task parallelly. Jacobian being the large sparse matrix, the suggested new method can fully avoid storing and retrieving of large saprse matrix. the let ‘m’ be the total number of measurements taken for the given network. Each row of jacobian is formed based on the type of measurement underconsideration, if there are ‘m’ measurements, the jacobian of the network will contain ‘m’ rows and (2n-1) coloms.

Instead of forming the complete jacobian of the netork and then multiplying to find ‘A’ matrix, (see eq1), It can be computed by each row of jacobean [$J_1 * W_1 * J_1 = A_1$]and [$J_1 * W_1 * z_1 = b_1$], this results in first terms of each elements of the resultant matrix “A” and “b” respectively, which can be visualized as shown below.

$$\{A_1 + A_2 \ldots + A_i \ldots + A_m\}\ \Delta x = \{b_1 \ldots + b_r \ldots + b_m\} \tag{3}$$

$$\sum_{j=1}^{m}(A_j)\Delta x = \sum_{j=1}^{m}(b_j) \tag{4}$$

$$A = \{A_1 + A_2 + A_r . + A_m\} \tag{5}$$

$$b = \{b_1 . + b_r + b_m\} \tag{6}$$

Hence, it is not necessary to form complete two dimentonal Jacobian and then carry-out the multiplication task. New jacobian is a one dimention arry of the order of [1*(2n-1)] is replaced by [m*(2n-1)], thereby no special sparse matrics storage is required.

Node Wise Grouping of Measurement: Jacobian Relation

The measurements can be grouped node wise, and it can be represented as shown below.

$$Node-1\ Measurements\begin{bmatrix}\begin{Bmatrix}\Delta P_1\\ \Delta Q_1\\ \Delta p_{ij}^1\\ \Delta q_{ij}^1\\ \Delta v_1\\ \Delta \partial_1\end{Bmatrix}\end{bmatrix} = \begin{bmatrix}\underbrace{\begin{Bmatrix}|H_1^1| & |H_2^1|\\ |H_3^1| & |H_4^1|\\ |H_5^1| & |H_6^1|\\ |H_7^1| & |H_8^1|\\ |H_9^1| & |H_{10}^1|\\ |H_{11}^1| & |H_{12}^1|\end{Bmatrix}}_{J_{NA1}}\end{bmatrix}\begin{bmatrix}\Delta\delta_i\\ \Delta v_i\end{bmatrix} \tag{7}$$

$$Node-r\ measurements\begin{bmatrix}\begin{Bmatrix}\Delta P_r\\ \Delta Q_r\\ \Delta p_{ij}^r\\ \Delta q_{ij}^r\\ \Delta v_r\\ \underbrace{\Delta \partial_r}_{z_{NAr}}\end{Bmatrix}\end{bmatrix} = \begin{bmatrix}\underbrace{\begin{Bmatrix}|H_1^r| & |H_2^r|\\ |H_3^r| & |H_4^r|\\ |H_5^r| & |H_6^r|\\ |H_7^r| & |H_8^r|\\ |H_9^r| & |H_{10}^r|\\ |H_{11}^r| & |H_{12}^r|\end{Bmatrix}}_{J_{NAr}}\end{bmatrix}\begin{bmatrix}\Delta\delta_i\\ \Delta v_i\end{bmatrix} \tag{8}$$

$$Node-n^{th}\ measurements\begin{bmatrix}\begin{Bmatrix}\Delta P_n\\ \Delta Q_n\\ \Delta p_{ij}^n\\ \Delta q_{ij}^n\\ \Delta v_n\\ \Delta \partial_n\end{Bmatrix}\end{bmatrix}=\begin{bmatrix}\underbrace{\begin{Bmatrix}|H_1^n| & |H_2^n|\\ |H_3^n| & |H_4^n|\\ |H_5^n| & |H_6^n|\\ |H_7^n| & |H_8^n|\\ |H_9^n| & |H_{10}^n|\\ |H_{11}^n| & |H_{12}^n|\end{Bmatrix}}_{J_{NAn}}\end{bmatrix}\begin{bmatrix}\Delta\delta_i\\ \Delta v_i\end{bmatrix} \tag{9}$$

$$A_{NAr}=\left\{\left(A_r^P+A_r^Q+\sum A_r^{pij}+\sum A_r^{qij}+A_r^v+A_r^\delta\right)\right\}$$

$$=J_{NAr}^T*W_r*J_{NAr} \tag{10}$$

$$b_{NAr}=\left\{\left(b_r^P+b_r^Q+\sum b_r^{pij}+\sum b_r^{qij}+b_r^v+b_r^\delta\right)\right\}$$

$$=J_{NAr}^T*W_r*z_{NAr} \tag{11}$$

$$A=\left(A_{NA1}+A_{NA2}+..A_{NAr}+..+A_{NAn}\right)=\sum_{j=1}^{n}(A_{NAj})$$

$$b=\left(b_{NA1}+b_{NA2}+...+b_{NAr}+...+b_{NAn}\right)=\sum_{j=1}^{n}(b_{NAj})$$

$$\sum_{j=1}^{n}(A_{NAj})\Delta x=\sum_{j=1}^{n}(b_{NAj}) \tag{12}$$

$$\sum_{j=1}^{m}(A_j)=\sum_{j=1}^{n}(A_{NAj})=A \tag{13}$$

$$\sum_{j=1}^{m}(b_j)=\sum_{j=1}^{n}(b_{NAj})=b \tag{14}$$

Hence AΔx = b

$(\ )_{NAi}$ – subscript 'NAi' refers to ith node area

A_r^P - It is the sub set of matrix 'A' for the injected Real power measurement taken at 'r'th node/bus. Similarly, A_r^Q (here Q represents reactive power) and like wise for other measurements.

Hence, from the above derivation, it is evedient that up to the formation of 'A_{NAi}' (node level), computations can be carried-out independently without any approximation, which gives the scope for parallel processing.

System Design

Assumptions

- All measurements are taken at the same instant
- At each node total of measurement can be: - injected power measurements of node + (node voltage measurement + node angle measurement +the line flow measurements) of that node to other connected nodes.

Flow Chart

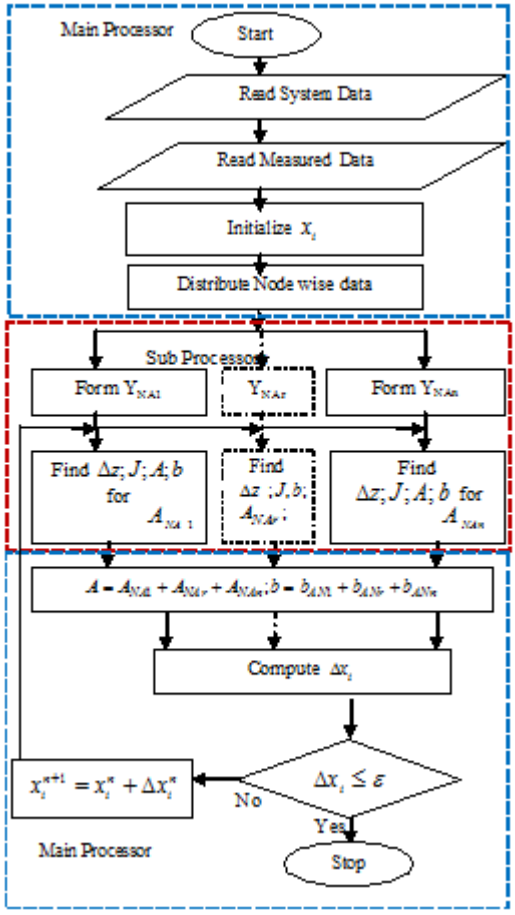

Figure 1

COMPUTING TIME ANALYSIS OF SE

General Technique

The computational time of state estimation using NR method is directly linked with the number of nodes in the network. Generally, to solve A*Δx =b, if the computational time required is (say) "t_1" for the network having n_1 number of nodes, then the computational time required for the network having n_2 number of nodes is approximately equals to $(n_2/n_1)^2*t_1$. Or in other words the SE computational time increases "square time" the ratio of increase in the number of nodes of the network. It's because as the number of node increases the size of the Jacobian and "A" matrix. For example

The size of matrix "A" for the network having 10 nodes is nearly equal to (20 x20) =400.

The size of matrix "A" for the network having 30 nodes is nearly equal to (60 x60) =3600.

$(3600/400) = 9 = (30/10)^2$.

Optimum Computing Technique

The fact is, these matrices are highly sparse in nature. These none zero elements of jacobian/"A" has a direct relation with the network incident matrix. Using this information one can design the computation focusing only on none zero elements to avoid operations which results in zero. This can be done by using the network incident matrix as index to compute arithmetic operation of none-zero elements. This makes the computational time required is directly proportional to the number of measurements and does not depend on the size of the matrices. SE computational time reduces drastically by using optimum computing technique.

For example, a 10 bus system having number of lines connected (say) = 14, then the size of Ybus = 10x10 = 100, whereas none zero elements in the Ybus = 10+14x2 = 38;

Hence, none zero elements in the matrix "A" = 38 x4=152; whereas size of "A" approximately equal to (20x20) =400. Say "t_1" is SE computational time for the network having 10 nodes using general method, then by using optimum technique, SE computational time for the same network is approximately equal to (152/400)x t_1, let this be equal to t_2. It's obvious that $t_2 << t_1$. Now if the network size is increased to n_2 (say for example n_2 =30, number of connected lines=41, none zero Ybus elements=30+41x2=112 hence number of none zero elements of matrices=112x4=448), then the computing time is approximately proportional to (448/152) = which is nearly equal '3', hence the computing time is nearly equal to= (n_2/n_1) x t_2 . Because, whatever the size of matrices, computational time depends upon the number of none zero elements which increases almost linearly with node.

Total SE computational time can be related with the following equation

$$(SE)_t = \left(\Delta xi = A^{-1} * b\right)_t + \left(J^T * J\right)_t + (\varepsilon)_t$$

$(SE)_t$ = total SE computational time

$\left(\Delta xi = A^{-1} * b\right)_t$ = Time taken to solve Δxi

$\left(J^T * J\right)_t$ = Time taken to obtain A & b

$(\varepsilon)_t$ = updating time

EXAMPLE & RESULTS

Note: All quantities are in pu.

Table 1: Input Line Data for 13 Bus System (ISE)

Bus No	R(Pu)	X(Pu)
1-2	.00147967	.00286760
2-3	.00043843	.00124174
3-4	.00027711	.00078428
4-5	.00059760	.00166769
4-8	.00159967	.00310017
5-6	.00034314	.00097190
5-9	.00034314	.00097190
6-7	.00032364	.00091669
7-10	.00032364	.00091669
8-9	.00029438	.00083380
9-10	.00053157	.00150562
9-12	.00037793	.00107050
10-11	.00058777	.00166488
11-13	.00032364	.00091669
12-13	.00036843	.00104355

Table 2: Injected Power & Voltage Measurements

Ems No.	Real	1mg.	Bus No.	Volt
1	28.5335	13.0837	1.	1.05300
2	-4.73	-1.55	3.	0.95792
3	-1.27	-0.41	10.	0.92304
4	-0.35	-0.11	11.	0.91939
5	-4.38	-1.44	12.	0.92178
6	-2.11	-0.69	13.	0.92036
7	-0,42	-0.13	0	0
8	-4.73	-1.55	0	0
9	-1.27	-0.41	0	0
10	-0.35	-0.11	0	0
11	-4.38	-1.44	0	0
12	-2.11	-0.69	0	0
13	-0.42	-0.13	0	0

Table 3: Line Flow Measurements

Bus No.	Real	1mg.
1-2	28.5327	13.0837
2-1	-27.2179	-10.5355
2-3	22.4879	8.9855
3-2	-22.2197	-8.2258
3-4	20.9000	7.7700
4-5	12.9700	5.1000
4-8	7.6200	2.1282
5-9	4.3979	1.9040
8-4	-7.4182	-1.9161
8-9	2.7900	0.3800
9-10	1.,8043	0.5194
11-13	-1.4600	-0.4582

Table 4: SE Result of L3 Bus Test System
***Slack Bus**

No	Volt	Ang Rad
1	1.053269	0.101912
2	0.979368	0.041320
3	0.958231	0.015757
4*	0.945912	0.000000
5	0.928889	—0.021096
6	0,925904	—0.025075
7	0.924513	-0.026839
8	0.926468	—0.022792
9	0.925278	—0.025320
10	0.923398	—0.028199
11	0.919737	—0.033261
12	0.922133	—0.029788
13	0.920707	—0.031856

***** Convergence =.001 for voltage,
= .00001 for angle.

Table 5: Computational Time for 13 Bus Test System

	Total SE Time	Time-up to Jacobian Product
Conventional method	1.08 ms	0.88 ms
New method (single processore)	0.52 ms	0.32 ms
New method (paralle processores)*****	0.225 ms	0.025 ms

*****Number of parallel processors are assumed to be =13

Table 6: Relative Time

	Total SE Time Ratio	Time Ratio-Jacobian Product
Conventional /new single ≈	2	2.75
Conventional /new parallel ≈	5	35
New single/New parallel ≈	2.36	13

The above table shows actual results obtained which is same as that of the Integrated SE results obtained by the conventional method. It shows that both approach yields the same results. Thereby this new concept has been proved with this example.

CONCLUSIONS

Node wise grouping of data leads to new approach for applying the NR solution. The number of state variables to be computed at each bus-level is very small as compared to the whole network, and till the last step computations can be carriedout independently & separately thereby the fast computation is possible. If sufficient numbers of measurements are ensured at each node, then state estimantion can be carried out at each bus. At Central computing station only the final estimated state variables are to be considered as sudeo-measurements. The other possibilities of this approach is under progress.

ACKNOWLEDGEMENTS

The authors would like to thank Dr. Vinay Pant and Dr. G. K. Singh, Prof., Dept. E&E, IIT Roorkee for their motivation and support.

REFERENCES

1. Anurag K., Srivastava, “Distributed state Estimation with PMU Using grid Computing”, 978-1-4244-4241-6/ 09/$25.00@2009IEEE
2. M.Y.Patel, A.A.Girgis,, “Two-Level State Estimation for Multi-Area Power system”, 1-4244-1298/$25.00 @2007IEEE
3. Patel M. Y., Girgis A. A., "*Two-Level State Estimation for Multi-Area Power System*",1-4244-1298-6/07/$25.00 ©2007 IEEE.
4. Jeu-Min Lin, Shyh-Jier Huang, and Kuang-Rong Shih, *“Application of Sliding Surface Enhanced Fuzzy Control for Dynamic State Estimation of a Power System”,* IEEE Transactions On Power Systems, VOL. 18, NO. 2, MAY 2003.
5. S. J. Huang and K. R. Shih, *“Dynamic-state-estimation scheme including nonlinear measurement-function considerations*”, IEE PROC. Generation,Transmission and Distribution, Vol 149, No. 6, November 2002.
6. S. K. Sinha and J. K. Mandal, *“Dynamic State Estimator Using ANN based Bus load Prediction”,* IEEE Transactions on Power Systems, Vol. 14, No. 4, November 1999.
7. Habiballah, PhD, *“Modified two-level state estimation approach”* IEE Proc-Gener. Transm. & Distrib., Vol. 143, No. 2, March 1996.

8. Bahgat A. , Sakr M. M. F., El-Shafei A. R., "*Two level dynamic state estimator for electric power systems based on nonlinear transformation*", *IEE Proceedings*, Vol. 136, Pt. C, No. I, JANUARY 1989.

9. W. W. Kotiuga, "*Development of a Least Absolute Value Power System Tracking State Estimator*", IEEE transactions on Power Apparatus andSystems, Vol. PAS-104, No. 5, May 1985.

10. K.L. Mofreh, M. Salem, McColl, R.D MOFFATT,: "*Two-level power system state estimation*", 20th Universities Power engineering conference (UPEC 85), Huddersfield, UK, April 1985, pp. 3740

11. K.L. Mofreh, M. Salem, McColl, R.D.Moffatt, and SULLEY J.L.: "*Multi-level state estimation for electric power systems*", 19th Universities Power Eng. Conf. Paper 14.2, Dundee, April 1984

12. KURZYN, M.S.: "*Real-time state estimation for large-scale power systems"*, ihid, 1983, PAS-102, pp. 2055-2063

13. Van Cutsem, T.H. Horward J.L., Ribbens-Pavella, M., and EL-Fattah,Y.M. : '*Hierarchial state estimation*', Int. J. Electrical Power & Energy Syst., 1980, 2, pp. 7&80

14. Van Cutsem, T.H., Horward, J.L., and Ribben-Pavella M.: "*Application of a two-level state estimation for anomalous data analysis in electric power system"*, Proc. IFAC Symp. on automatic control in power generation, distribution and protection, Pretoria, September 1980, pp. 423433