Implementation of Polar Form Algorithms with the Rectangular Form Complex Number

Gong Chengming, *Member, IEEE*

***Abstract*—Power system analysis algorithms under polar coordinates are traditionally implemented with the polar form complex number, which is abstracted as phase and magnitude. This paper addresses a method to implement the polar form algorithms with the common rectangular form complex number, which is abstracted as real and imaginary parts. The key points of the addressed method are to express the derivatives to polar components under rectangular coordinates, and to rotate and stretch a complex number under rectangular coordinates. With the addressed method, the polar form algorithms can be implemented with direct complex operations and will share more components with rectangular form algorithms.**

***Index Terms*—hybrid coordinates, iteration algorithm, power system analysis**

I. Introduction

As the phase and magnitude of a voltage in the power system have clear physical meanings, implementing the power system algorithms under polar coordinates has many advantages; e.g., the PV buses in power flow and measurements of phase and magnitude in state estimation can be processed directly under polar coordinates. The polar form algorithms have been widely introduced and used.

The traditional implementation of polar form algorithms differs greatly with the rectangular form algorithms [1] [2] [3] [4]. Firstly, the definition and abstraction of complex numbers is different. In the rectangular form algorithms, the complex number is defined by real and imaginary parts, while in the polar form algorithms, it is defined by phase and magnitude. Secondly, the expressions of power flows and their derivatives are different. In the rectangular form algorithms, the power flows and derivatives can be expressed with direct complex operations, and the expressions are very concise. In the polar form algorithms, they have to be expressed with triangular operations in the real domain and thus become much more complicated.

It will be advantageous to implement the polar form algorithms with the rectangular form complex number. Though many high-level programming languages provide complex data types (e.g., in FORTRAN) or standard complex classes (e.g., in C++) for direct complex operations, the traditional implementation of polar form algorithms cannot use these complex data types or classes. As for the implementation with the rectangle form complex number, the power flows and derivatives will be expressed with these types or classes in the complex domain, and the complex operation will be performed as a whole. Furthermore, the implementation with rectangular form complex numbers will share more components with the rectangular form algorithms. From the software engineering point of view, this kind of sharing is helpful for improving the quality of software design and consequently improving the productivity and reliability of the software.

This paper addresses a solution for the problem of implementing the polar form algorithms with the rectangular form complex number. The paper firstly formulates the polar form algorithms more generally, and analyzes what operations are needed for an abstracted complex type to support the polar form algorithms. The paper then addresses how to implement these operations for a rectangular form complex number. Lastly, reformed implementation of polar form algorithms with the rectangular form complex number is addressed.

II. Formulation

Iteration algorithms based on complex variables are widely used in power system analysis. For certain problems such as power flow, state estimation, etc., corresponding non-linear equations are built and attempted to be solved by iteratively updating the state variables with local linear solutions in limited steps. The complex bus voltages are normally selected as state variables, and the algorithms may be implemented under rectangular coordinates or polar coordinates.

The traditional implementation under polar coordinates updates the phase vector θ and magnitude vector $|\mathbf{V}|$ iteratively and calculate the power flows and derivatives based on θ and $|\mathbf{V}|$. The implementations can be generally formulated as:

1) Initialize state variables $\theta^{(0)}$ and $|\mathbf{V}|^{(0)}$;
2) Try to find converged solutions in limited cycles of iteration:
   a) solve $\boldsymbol{\Delta}\theta^{(k)}$ and $\boldsymbol{\Delta}|\mathbf{V}|^{(k)}$, based on power flows and derivatives according to certain problems;
   b) Update the state variables, let $\theta^{(k+1)} = \theta^{(k)} + \boldsymbol{\Delta}\theta^{(k+1)}$ and $|\mathbf{V}|^{(k+1)} = |\mathbf{V}|^{(k)} + \boldsymbol{\Delta}|\mathbf{V}|^{(k)}$.

In this kind of implementation, the complex number, which is abstracted as phase and magnitude, cannot be handled as a whole to calculate the power flows and derivatives, and the expressions of power flows and derivatives have to be expanded as explicit triangular operations.

For example, given a general branch which links bus i and bus j and whose two-terminal model is

$$\begin{pmatrix} y_{ii} & y_{ij} \\ y_{ji} & y_{jj} \end{pmatrix} = \begin{pmatrix} g_{ii} + jb_{ii} & g_{ij} + jb_{ij} \\ g_{ji} + jb_{ji} & g_{jj} + jb_{jj} \end{pmatrix}, \qquad (1)$$

This paper was supported in part by the National Science and Technology Infrastructure Program of the Ministry of Science and Technology of China under Grant 2008BAA13B06.

Gong Chengming is with NARI Technology Development Co. Ltd., Nanjing, Jiangsu, 210061, China (e-mail: gongchm@naritech.cn).

the active power flow at the i-side from bus i to bus j will be

$$p_{ij} = |V_i|^2 g_{ii} + |V_i| |V_j| (g_{ij} \cos\theta_{ij} + b_{ij} \sin\theta_{ij}), \quad (2)$$

where θ_i, $|V_i|$, θ_j, and $|V_j|$ are the voltage phases and magnitudes at bus i and bus j respectively, and $\theta_{ij} = \theta_i - \theta_j$.

It is not difficult to deduce from (2) the derivatives of p_{ij} to the voltage phase or the magnitude at both ends. The derivative of p_{ij} to the phase at the i-side is given below as an example [4].

$$\frac{\partial p_{ij}}{\partial \theta_i} = |V_i||V_j|(b_{ij} \cos\theta_{ij} - g_{ij} \sin\theta_{ij}). \quad (3)$$

When observing this implementation, a question may arise as to whether it is possible to simplify such expressions as (2) and (3) and unify them with the similar expressions in rectangular form implementations. This is the problem this paper has studied and solved.

Actually, the essence of the polar form algorithms is to update the bus voltages with phase changes and magnitude changes. The features of the polar form algorithms have nothing to do with how the complex voltages are expressed (as rectangular form or polar form) and how the changes of phase and magnitude are calculated. Therefore, the polar algorithms can be formulated more generally as:

1) Initialize state variables $\mathbf{V}^{(0)}$;
2) Try to find converged solutions in limited cycles of iteration:
   a) solve $\boldsymbol{\Delta}\theta^{(k)}$ and $\boldsymbol{\Delta}|\mathbf{V}|^{(k)}$, based on power flows and derivatives according to certain problems;
   b) Update the state vector $\mathbf{V}^{(k)}$ to $\mathbf{V}^{(k+1)}$ by solved $\boldsymbol{\Delta}\theta^{(k)}$ and $\boldsymbol{\Delta}|\mathbf{V}|^{(k)}$.

It can be seen from the new formulation that, theoretically, any complex abstraction can be used to implement the polar algorithm, as long as it can be utilized to calculate the power flows and derivatives and can be updated with changes of phase and magnitude. The polar form complex number is not the only option for implementing the polar form algorithms. As the rectangular form complex can be operated directly in modern programming languages, the implementations based on it may have some advantages that the implementations based on the polar form complex number do not have. The next two sections will further discuss how the rectangular form complex number can easily meet the demand in the latter formulation of polar form algorithms.

III. Power Flows and Their Derivatives to Polar Components under Rectangular Coordinates

When the complex voltages are expressed in rectangular coordinates, it is convenient to calculate the power flows in the same way as in the rectangular form algorithms; e.g., compared with (2), the complex current and complex power at the i-side from bus i to bus j will be calculated as

$$I_{ij} = V_i y_{ii} + V_j y_{ij}, \quad (4)$$

and

$$S_{ij} = V_i I_{ij}^*. \quad (5)$$

To further calculate the derivatives, a general form will first be discussed. For any continuous and smooth function with the complex voltage vector $\mathbf{V}$ as the free variables, $f(\mathbf{V})$, the partial derivatives of $f(\mathbf{V})$ to any component of a voltage is able to be expressed as

$$\frac{\partial f(\mathbf{V})}{\partial x_i} = \frac{\partial f(\mathbf{V})}{\partial V_i} \frac{\partial V_i}{\partial x_i}, \quad (6)$$

where x_i, one component of voltage V_i, may be the real part e_i or the imaginary part f_i under rectangular coordinates, the phase θ_i or the magnitude $|V_i|$ under polar coordinates.

In (6), the problem of $\frac{\partial f(\mathbf{V})}{\partial x_i}$ is decomposed into two parts. One part solves $\frac{\partial f(\mathbf{V})}{\partial V_i}$, which will handle the complex number as a whole and will be shared by any form of x_i. The other part calculates $\frac{\partial V_i}{\partial x_i}$, which only relates to the complex voltage V_i itself. Calculation of the derivatives and unification of the calculation under different coordinates will be facilitated by the decomposition.

For calculation under polar coordinates, as $V_i = |V_i| e^{j\theta_i}$, the second part $\frac{\partial V_i}{\partial x_i}$ will be

$$\frac{\partial V_i}{\partial \theta_i} = |V_i| j e^{j\theta_i} = jV_i, \quad (7)$$

and

$$\frac{\partial V_i}{\partial |V_i|} = e^{j\theta_i} = \frac{V_i}{|V_i|}. \quad (8)$$

With (6), (7), and (8), derivatives of all power flows to a polar component of a complex voltage can be deduced. Compared with (3), the derivatives of I_{ij} and S_{ij} to phase angle at bus i are deduced from (4), (5), and given below as examples.

$$\frac{\partial I_{ij}}{\partial \theta_i} = (\frac{\partial V_i}{\partial V_i} y_{ii} + \frac{\partial V_j}{\partial V_i} y_{ij}) \frac{\partial V_i}{\partial \theta_i} = jV_i y_{ii} \quad (9)$$

and

$$\frac{\partial S_{ij}}{\partial \theta_i} = \frac{\partial V_i}{\partial \theta_i} I_{ij}^* + V_i \frac{\partial I_{ij}^*}{\partial \theta_i} = jV_i V_j^* y_{ij}^*. \quad (10)$$

The complex numbers are handled as a whole in (9) and (10) so the calculations can be implemented conveniently with the rectangular form complex numbers. Higher-order derivatives can be further deduced in the same way. It can be proved that the real part in (10) is equivalent to (3). The proof is given in the appendix.

IV. Updating Polar Form Complex Number with Phase and Magnitude Changes

Updating a complex number by phase change and magnitude change can be regarded as rotating and stretching the complex number. Regardless of the form of the complex number, the rotation and stretching can be expressed as corresponding operators. The operator of rotating a complex $V_i^{(k)}$ with $\Delta\theta_i^{(k)}$ is

$$r_i^{(k)} = e^{j\Delta\theta_i^{(k)}} = \cos\Delta\theta_i^{(k)} + j\sin\Delta\theta_i^{(k)}, \quad (11)$$

and the operator of stretching a complex $V_i^{(k)}$ in length $\Delta |V_i|^{(k)}$ is

$$s_i^{(k)} = 1 + \frac{\Delta \left|V_i\right|^{(k)}}{\left|V_i\right|^{(k)}}, \left|V_i\right|^{(k)} \neq 0. \tag{12}$$

With (11) and (12), updating the complex voltage by $\Delta\theta_i^k$ and $\Delta \left|V_i\right|^{(k)}$ will be

$$V_i^{(k+1)} = s_i^{(k)} r_i^{(k)} V_i^{(k)}. \tag{13}$$

It is the same case as in (9) and (10): the complex voltage in (13) is handled as a whole and can be easily operated under rectangular coordinates.

V. Reformed Implementation

By applying the results in the previous two sections to the reformed formulation in section II, the polar form algorithms can be implemented with rectangular form complex numbers. The final formulation is expressed as

1) Initialize state variables $\mathbf{V}^{(0)}$;
2) Try to find converged solutions in limited cycles of iteration:
   a) solve $\boldsymbol{\Delta}\theta^{(k)}$ and $\boldsymbol{\Delta}|\mathbf{V}|^{(k)}$, based on power flows and derivatives according to certain problems. The power flows and derivatives will be calculated by expressions like (4), (5), (9), and (10);
   b) Update the state vector $\mathbf{V}^{(k)}$ to $\mathbf{V}^{(k+1)}$ according to (11) and (12).

This implementation will benefit from using direct complex operations provided by a modern programming language. At the same time, this implementation will share the same expressions of power flows with rectangular form algorithms, and the expressions of derivatives differ in the rectangular form and polar form algorithms only at the second part in the right side of (6).

VI. Conclusion

This paper proposes a method to implement the polar form power system analysis algorithms with the rectangular form complex number. The proposed method can both benefit from the direct complex operation and share more common components with rectangular form algorithms. It is expected that the productivity and reliability of the power system analysis software can be notably improved by using the proposed method.

A power flow program has been implemented using the proposed method, and the results have been verified as correct. Other aspects such as efficiency need to be further compared with the traditional implementations.

Appendix

Proof of the equivalence of (3) and the real part in (10).

Proof:

$$\begin{aligned}
V_i V_j^* &= \left|V_i\right| e^{j\theta_i} \left|V_j\right| e^{-j\theta_j} \\
&= \left|V_i\right| \left|V_j\right| (\cos\theta_{ij} + j\sin\theta_{ij}) \\
&\Downarrow \\
\frac{\partial S_{ij}}{\partial \theta_i} &= j V_i V_j^* y_{ij}^* \\
&= j \left|V_i\right| \left|V_j\right| (\cos\theta_{ij} + j\sin\theta_{ij})(g_{ij} - jb_{ij}) \\
&= \left|V_i\right| \left|V_j\right| (\cos\theta_{ij} + j\sin\theta_{ij})(b_{ij} + jg_{ij}) \\
&\Downarrow \\
\frac{\partial p_{ij}}{\partial \theta_i} &= Re(\frac{\partial S_{ij}}{\partial \theta_i}) \\
&= \left|V_i\right| \left|V_j\right| (b_{ij}\cos\theta_{ij} - g_{ij}\sin\theta_{ij})
\end{aligned}$$

■

Acknowledgment

The author would like to thank Professor Gao Zonghe at NARI for the initial discussion on a derivative formulation under polar coordinates, and thank Dr. Liu Xuewen at Imperial College London for the suggestion of enhancing the logic of the paper. The author also thanks the NAS group at NARI for allowing him to set aside enough time to prepare this paper.

References

[1] Arthur R. Bergen, Vijay Vittal, *Power System Analysis, 2nd Edition*. Prentice Hall, New Jersey, 2000.
[2] Ali Abur, Antonio Gómez Expósito, *Power System State Estimation: Theory and Implementation*. Marcel Dekker, New York, 2004.
[3] Zhang Boming, Chen Shousun, Yan Zheng, *Advanced Power Grid Analysis, 2nd Edition*. Chinese, Tsinghua University Press, Bejing, 2007.
[4] Antonio Gómez Expósito, Antonio J. Conejo, Claudio Cañizares, *Electric Energy Systems - Analysis and Operation*. CRC Press, Boca Raton, 2008.

Gong Chengming (M'2005) was born in Xiangshui, Jiangsu, China in 1977. He graduated from Nanjing University with a B.Sc. degree in 1998 and from Nanjing Automation Research Institute with a Ms.E.E. degree in 2001 respectively.

He is currently a senior software engineer and a deputy department manager in NARI Technology Development Co. Ltd., where he participated in the research, design, and development of the OPEN-3000 Energy Management System. His personal research interests include power system modeling, state estimation, and the application of modern software technologies to analysis and design of the network analysis software. He has published several papers in these areas.