A Combined TSA-SPA Algorithm for Computing Most Sensitive Eigenvalues in Large-scale Power Systems

C. Y. Chung, *Senior Member, IEEE,* and Bo Dai, *Member, IEEE*

Abstract— **A novel algorithm (TSA-SPA) that combines the Two-Sided Arnoldi method (TSA) and the Sensitive Pole Algorithm (SPA), is proposed in this paper for calculation of the most sensitive eigenvalues to control parameters in large power systems. In the proposed method, first, with the shift-invert transformation precondition, TSA builds two Krylov subspaces and obtains a reduced matrix of a much smaller scale, which contains eigenvalues close to the chosen shift point. Second, SPA is adopted to realize the most sensitive eigenvalue computation. TSA-SPA can find the most sensitive eigenvalues of interest, with satisfactory reliability and convergence, in a specified frequency domain. With proper selection of sizes of Krylov subspace and the reduced matrix, the convergence to good eigenvalue approximations is practically guaranteed. Moreover, with the deflation technique, the algorithm is also capable of finding several other dominant eigentriplets which may relate to inter-area and/or local control modes. The efficiency of the proposed algorithm has been validated on small and large-scale power systems. It has been found that compared to other available sensitive pole algorithms, the proposed algorithm has more robust and reliable performance. The proposed algorithm is suitable for practical applications in large-scale power systems.**

***Index Terms*— Eigenvalues, sensitivity, large-scale eigenvalue problems, small-signal stability, system oscillations.**

I. Introduction

With the ever-increasing scale of power systems, low frequency oscillations occur more frequently and have become one of the most important problems in system planning and operation [1-5]. Controllers, such as power system stabilizers (PSS), have to be properly coordinated to ensure sufficient damping of oscillatory modes, i.e. damping ratios of the corresponding eigenvalues [6-17]. Therefore, computing critical eigenvalues of a poorly-damped power system sensitive to control parameters is essential for studying the effect of control parameters on these oscillatory modes and to facilitate control design.

The traditional full-eigenvalue computation methods such as QR/QZ methods are inefficient when applied to modern large-scale power systems because they require high computational memory for solving high dimensional problems [18]. Moreover, normally, only a few dominant eigenvalues are of interest to system operators; it is unnecessary and impractical to compute all eigenvalues of the system.

Selective eigenvalue algorithms of subspace methods, e.g., subspace iteration, Lanczos method and Arnoldi method, are commonly adopted for computation of eigenvalues of large matrices, through matrix reduction. Among them, the Arnoldi method is one of the most popular techniques and is believed to be the most efficient approach [19-21] as it improves the slow convergence in subspace iteration with large dominance ratios [22] and also overcomes instabilities that Lanczos may encounter [23]. It computes only eigenvalues of interest and offers low computational cost with the sparse descriptor matrices. However, it is a single-sided method which uses one Krylov subspace, i.e. it can only calculate the right eigenvectors. Although the Nonsymmetric Lanczos method, a two-sided method, can calculate eigentriplets, i.e. obtain eigenvalues and the associated right and left eigenvectors simultaneously, incurable breakdowns caused by biorthogonalization between left and right Lanczos vectors may occur during building of Krylov subspaces [24].

Dominant Pole Algorithm (DPA) is proposed in [25] to compute eigenvalues with the largest residue of a high order transfer function based on input and output vectors. Indeed, eigenvalues with the highest participation of state variables [25] can also be obtained through replacing input and output vectors with suitable vectors in the process of computation of DPA.

However, the above algorithms are still unable to calculate the most-sensitive eigenvalues with respect to specified control parameters. Based on DPA, Sensitive Pole Algorithm (SPA) is, therefore, proposed [26] for computing poles (i.e. eigenvalues) most sensitive to control parameters changes, and then it is extended to Subspace Accelerated SPA (SASPA) by applying subspace acceleration [27, 28] to improve its global convergence. Because SASPA operates on sparse descriptor matrices, it is also useful for analyzing large-scale systems. SASPA can obtain several most sensitive eigenvalues and the corresponding eigenvectors through suitably selecting pole estimates by shift strategy and deflation method.

Based on the idea of subspace methods, a novel algorithm (TSA-SPA) which combines Two-Sided Arnoldi method (TSA) and SPA is developed to compute the most sensitive eigenvalues and to improve robustness, flexibility and

This work was supported by Research Grants Council of Hong Kong (PolyU 5154/08E) and the Department of Electrical Engineering of The Hong Kong Polytechnic University.

C. Y. Chung and Bo Dai are with the Computational Intelligence Applications Research Laboratory (CIARLab), Department of Electrical Engineering, The Hong Kong Polytechnic University, Hung Hom, Hong Kong. (email: eecychun@polyu.edu.hk; eedebra@polyu.edu.hk).

scalability of the SPA method. TSA method [29] not only inherits the advantages of the Arnoldi method but also does not have the breakdown problem of nonsymmetric Lanczos method. Shift-invert transformation [22, 30] is adopted as the precondition in TSA to reduce the region for eigenvalue search and then SPA is adopted to make sensitive eigenvalues dominant in the region of interest. Compared to SASPA, which scans the whole complex plane, TSA-SPA focuses on a specific area, as dictated by the shift. Further, the shift is not required to be updated with subspace expansion and thus the LU factorization needs to be executed only once and can be stored for use in successive iterations, making computation cheaper than SASPA. Furthermore, Implicitly Restarted Arnoldi Method (IRAM) [31] is adopted in TSA. IRAM has high efficiency in altering the starting vector as it greatly improves the convergence while keeping the size of subspace constant with small scale; and enables the convergence speed to remain almost unaffected by shift point selection. Finally, deflation technique [28, 31, 32] is adopted in the proposed method to find several other dominant eigenvalues without modification of either the reduced matrix or spanned Krylov subspaces.

The rest of this paper is organized as follows. Section II covers background of the proposed algorithm. Section III introduces the proposed TSA-SPA algorithm. Section IV provides a comprehensive study of the application of the proposed algorithm on a small power system. Section V then applies the proposed algorithm to large-scale power systems. Finally, conclusions are provided in Section VI.

II. BACKGROUND

The mathematical model of a linearized single-input single-output (SISO) dynamic system around an operating point x_0 can be expressed as:

$$\begin{cases} \dot{\boldsymbol{z}}(t) = \boldsymbol{A}\boldsymbol{z}(t) + \boldsymbol{b}u(t) \\ y(t) = \boldsymbol{c}^* \boldsymbol{z}(t) + Du(t) \end{cases} \tag{1}$$

where $\boldsymbol{A} \in \Re^{n\times n}$ is the state matrix, $\boldsymbol{z}(t) \in \Re^n$ is the state vector, $u(t)$ is the input variable, $y(t)$ is the output variable, $\boldsymbol{b} \in \Re^n$ is the input column vector, $\boldsymbol{c} \in \Re^n$ is the output column vector and $D \in \Re$ is the direct transmission matrix of unit rank. Without loss of generality, $D = 0$. Superscript * denotes the conjugated transpose function while T denotes the transpose function in this paper.

A. Brief Overview of SPA

Transfer function $\boldsymbol{H}(s)$ of the dynamical system in (1) can be expressed as a sum of residues $R_i \in C$ over finite first-order k poles. Commonly, poles with small residues are neglected due to the little influence they have on the transfer function [25].

$$\boldsymbol{H}(s) \cong \sum_{i=1}^{k} \frac{R_i}{s - \lambda_i} \tag{2}$$

where the residues are

$$R_i = (\boldsymbol{c}^* \boldsymbol{x}_i)(\boldsymbol{y}_i^* \boldsymbol{b}) \tag{3}$$

and vectors $\boldsymbol{x}_i$, $\boldsymbol{y}_i$ denote the corresponding right and left eigenvectors, respectively, of pole λ_i.

Besides, the eigenvalue sensitivity of a parameter p can be determined by:

$$\frac{\partial \lambda_i}{\partial p} = \boldsymbol{y}_i^* \frac{\partial \boldsymbol{A}}{\partial p} \boldsymbol{x}_i \tag{4}$$

DPA [25] determines the pole with the largest residue and the corresponding eigenvectors based on the given input and output vectors $\boldsymbol{b}$ and $\boldsymbol{c}$. The eigenvalue with the largest sensitivity can be calculated by computing the pole with the largest residue of a modified dynamic system where $\boldsymbol{b} = (\partial \boldsymbol{A}/\partial p)\boldsymbol{x}_i$ and $\boldsymbol{c} = (\partial \boldsymbol{A}^*/\partial p)\boldsymbol{y}_i$, making the largest residue $R_i = (\boldsymbol{c}^* \boldsymbol{x}_i)(\boldsymbol{y}_i^* \boldsymbol{b}) = (\boldsymbol{y}_i^* (\partial \boldsymbol{A}/\partial p)\boldsymbol{x}_i)^2$ accord with eigenvalue with the largest sensitivity. SPA [26], therefore, extends DPA to compute the most sensitive eigentriplet by renewing vectors $\boldsymbol{b}$ and $\boldsymbol{c}$, no longer the input and output constant vectors in DPA, until the right and left eigenvectors $\boldsymbol{x}_i$ and $\boldsymbol{y}_i$ are converged and the above specific conditions are satisfied. The procedure of SPA is shown in Alg. 1 as follows, and the detailed proof can be referred to in [26].

Algorithm 1: SPA Algorithm

INPUT: Matrix $\boldsymbol{A}$, $\boldsymbol{A}_p \equiv (\partial \boldsymbol{A}/\partial p)$, pole estimate s_0 and tolerance ε

OUTPUT: Sensitive pole λ and corresponding right and left eigenvectors $\boldsymbol{x}$ and $\boldsymbol{y}$

1> Set $k = 0$, $\boldsymbol{v}_0, \boldsymbol{w}_0 \in \Re^n$ s.t. $\boldsymbol{A}_p \boldsymbol{v}_0 \neq 0$ and $\boldsymbol{A}_p^* \boldsymbol{w}_0 \neq 0$

2> **while** not converged **do**

3> $\boldsymbol{b} = \boldsymbol{A}_p \boldsymbol{v}_k \big/ \left\| \boldsymbol{A}_p \boldsymbol{v}_k \right\|_2$

4> $\boldsymbol{c} = \boldsymbol{A}_p^* \boldsymbol{w}_k \big/ \left\| \boldsymbol{A}_p^* \boldsymbol{w}_k \right\|_2$

5> Solve $\boldsymbol{v}_{k+1} \in \boldsymbol{C}^n$ from $(s_k \boldsymbol{I} - \boldsymbol{A})\boldsymbol{v}_{k+1} = \boldsymbol{b}$

6> Solve $\boldsymbol{w}_{k+1} \in \boldsymbol{C}^n$ from $(s_k \boldsymbol{I} - \boldsymbol{A})^* \boldsymbol{w}_{k+1} = \boldsymbol{c}$

7> Compute the new pole estimate

$$s_{k+1} = s_k - \frac{\boldsymbol{c}^* \boldsymbol{v}_{k+1}}{\boldsymbol{w}_{k+1}^* \boldsymbol{v}_{k+1}}$$

8> $\boldsymbol{v}_{k+1} = \boldsymbol{v}_{k+1} / \left\| \boldsymbol{v}_{k+1} \right\|_2$ and $\boldsymbol{w}_{k+1} = \boldsymbol{w}_{k+1} / \left\| \boldsymbol{w}_{k+1} \right\|_2$ The pole $\lambda = s_{k+1}$ with $\boldsymbol{x} = \boldsymbol{v}_{k+1}$ and $\boldsymbol{y} = \boldsymbol{w}_{k+1}$ has converged if $\left\| \boldsymbol{A}\boldsymbol{x} - s_{k+1}\boldsymbol{I}\boldsymbol{x} \right\|_2 < \varepsilon$

9> Set $k = k + 1$

10> **end while**

However, for large-scale systems, when solving the linear problem, Steps 5 and 6 in SPA become time-consuming due to the higher dimension n of Matrix $\boldsymbol{A}$ and s_k has to be updated via iterations in the n-dimension. To accelerate global convergence, SASPA [26] application is proposed, by integrating subspace acceleration in SPA, to search the most

sensitive poles in a matrix of a much smaller scale. The main idea behind subspace acceleration is that given a pole estimate, candidate vectors $\boldsymbol{v}_k$ and $\boldsymbol{w}_k$ can be obtained and kept in Steps 5 and 6 of Alg. 1 and two subspaces are built by candidate vectors $\boldsymbol{V}_m = [\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_m]$ and $\boldsymbol{W}_m = [\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_m]$, where m denotes the two subspaces' dimensions. In each subspace, vectors are orthogonalized to each other by using modified Gram-Schmidt [32]. Matrix $\boldsymbol{A}$ is, therefore, substituted by $\boldsymbol{W}_m^* \boldsymbol{A} \boldsymbol{V}_m$ and the most sensitive eigenvalues and the corresponding eigenvectors $\boldsymbol{v}_{k+1}$ and $\boldsymbol{w}_{k+1}$ can be calculated from the projected eigenproblem ($\boldsymbol{W}_m^* \boldsymbol{A} \boldsymbol{V}_m$, $\boldsymbol{W}_m^* \boldsymbol{V}_m$). With a carefully selected new pole estimate, SASPA can also find out other sensitive eigenvalues successfully through deflation procedure. For large scale power systems, a dynamic criterion is applied in shift strategy for a good new pole estimate. The new eigenvalue is obtained when the difference between sensitivity of the new eigenvalue and the already found eigenvalues is larger than a specified value.

However, for SASPA, subspaces are built by solving linear problems in Steps 5 and 6 of Alg. 1 with updated pole estimate s_k; selection of s_k based on the selection strategy may influence convergence speed. Furthermore, LU factorization in n - dimension should then be executed in every iteration, which can increase computation cost. Besides, the found eigenvalues may scatter throughout the complex plane and sensitive eigenvalues in areas of interest might not be included due to the limited dimension of the subspace. To address these limitations, a novel TSA-SPA method is proposed and explained in detail in the next section.

B. Two-Sided Arnoldi Method

As shown in Alg. 2, Krylov subspace $\boldsymbol{V}_k$ of k dimensions is first built by Arnoldi method [33], and eigenpairs are computed from the $k \times k$ upper Hessenberg matrix $\boldsymbol{H}_k$. Since the upper Hessenberg matrix is much smaller than Matrix $\boldsymbol{A}$, Arnoldi method is very effective in solving large-scale eigenvalue problems. Unlike Nonsymmetric Lanczos method, the Krylov subspace in Arnoldi method is built by only Matrix $\boldsymbol{A}$ projection and no bi-orthogonal procedure between Krylov subspaces is executed. Arnoldi method can calculate eigenvalues with the largest modulus or the largest real part, etc. [33]. However, Arnoldi method suffers several disadvantages. First, the size of Krylov subspace needs to be large enough to achieve a good eigenspace approximation of Matrix *A*. As can be seen from Step 9 of Alg. 2, if factor $h_{k+1,k}$ of residual matrix $h_{k+1,k} \boldsymbol{v}_{k+1} \boldsymbol{e}_k^T$ is zero, then eigenvalues of $\boldsymbol{H}_k$ (termed as Ritz value) are eigenvalues of *A*. To reduce $h_{k+1,k}$ to a very small value, Krylov subspace $\boldsymbol{V}_k$ has to be expanded adequately, which may require high memory storage [31]. Thanks to the restarting technique [34], the size of Krylov subspace can be held constant. In this paper, the IRAM, instead of the standard Arnoldi method, is adopted in TSA. Secondly, as shown in Alg. 2, columns of $\boldsymbol{V}_k$ and upper Hessenberg matrix $\boldsymbol{H}_k$ are obtained by means of single-side Arnoldi recurrence. The main deficiency of the procedure is that only right eigenvectors can be obtained and that does not yield a good approximation of the left eigenvector unless Matrix $\boldsymbol{A}$ is symmetric [29].

TSA method, a powerful alternative to subspace methods (Alg. 3), is proposed to solve this problem [29]. The orthogonal projections of Matrices $\boldsymbol{A}$ and $\boldsymbol{A}^T$ are executed independently. Bases for the two Krylov subspaces $\boldsymbol{V}_k = [\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_i, \ldots, \boldsymbol{v}_k]$ and $\boldsymbol{W}_k = [\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_i, \ldots, \boldsymbol{w}_k]$ are built independently in TSA, which means the bi-orthogonal procedure executed between Schur vectors $\boldsymbol{v}_i$ and $\boldsymbol{w}_i$ is not required and, therefore, breakdown will not occur. As long as $\boldsymbol{U}_k$ (Step 3 of Alg. 3) has full rank, reduced matrices $\widehat{\boldsymbol{H}}_k$ ($\widehat{\boldsymbol{H}}_k \equiv \boldsymbol{U}_k^{-1} \boldsymbol{W}_k^* \boldsymbol{A} \boldsymbol{V}_k$) and $\widehat{\boldsymbol{M}}_k$ ($\widehat{\boldsymbol{M}}_k \equiv \boldsymbol{W}_k^* \boldsymbol{A} \boldsymbol{V}_k \boldsymbol{U}_k^{-1}$) can always be obtained satisfactorily, as the residual matrix biorthogonal to the left (right) Krylov subspaces $\boldsymbol{W}_k^* h_{k+1,k} \boldsymbol{v}_{k+1} \boldsymbol{e}_k^T = 0$ and $m_{k+1,k}^* \boldsymbol{e}_k \boldsymbol{w}_{k+1}^* \boldsymbol{V}_k = 0$. Compared with nonsymmetric Lanczos method, TSA is numerically stable and easier to implement [29].

Algorithm 2: Arnoldi Method

INPUT: Matrix $\boldsymbol{A}$, initial vector $\boldsymbol{v}_1$ and reduced matrix dimension k

OUTPUT: Eigenvalues λ_i and corresponding right eigenvectors $\boldsymbol{x}_i$, $i = 1, \ldots, k$

1> Normalize $\boldsymbol{v}_1$

2> **for** $i = 1$ to k **do**

3> $\boldsymbol{w}_i = \boldsymbol{A} \boldsymbol{v}_i$

4> $h_{ji} = \boldsymbol{v}_j^* \boldsymbol{w}_i$, $j = 1, \ldots, i$

5> $\boldsymbol{w}_i \leftarrow \boldsymbol{w}_i - \sum_{j=1}^{i} h_{ji} \boldsymbol{v}_j$

6> $h_{i+1,i} = \|\boldsymbol{w}_i\|_2$

7> $\boldsymbol{v}_{i+1} = \boldsymbol{w}_i / h_{i+1,i}$

8> **end do**

9> Let $\boldsymbol{H}_k = [h_{ji}]_{i,j=1}^k$, which satisfying $\boldsymbol{A} \boldsymbol{V}_k = \boldsymbol{V}_k \boldsymbol{H}_k + h_{k+1,k} \boldsymbol{v}_{k+1} \boldsymbol{e}_k^T$ and $\boldsymbol{V}_k = [\boldsymbol{v}_1, \ldots, \boldsymbol{v}_k]$

10> Compute the eigenpairs $(\lambda_i, \boldsymbol{z}_i)$, $i = 1, \ldots, k$ of $\boldsymbol{H}_k$ by the QR method

11> Compute the right eigenvectors $\boldsymbol{x}_i = \boldsymbol{V}_k \boldsymbol{z}_i, i = 1, \ldots, k$, λ_i is eigenvalue of matrix $\boldsymbol{A}$ once satisfies $\|\boldsymbol{A} \boldsymbol{x}_i - \lambda_i \boldsymbol{x}_i\|_2 < \varepsilon$

C. Precondition Transformation

Arnoldi method is translation invariant, i.e. the corresponding eigenvectors for Matrix $\boldsymbol{A}$ and its transformation are the same. Shift-invert transformation enables Arnoldi method to compute eigenvalues located near certain regions of

the complex plane. The transformation, i.e. $\boldsymbol{T}_{SI}=(\boldsymbol{A}-\sigma\boldsymbol{I})^{-1}$, maps eigenvalues λ close to the shift point σ, away from the origin, and eigenvalues far from σ, close to zero. In other words, the eigenvalue is changed to $(\lambda-\sigma\boldsymbol{I})^{-1}$, which makes eigenvalues adjacent to the shift point have the largest modulus. The transformation is very useful, especially when λ lies in a cluster of complex eigenvalues. This can be easily implemented in Arnoldi method by replacing it with Step 3 of Alg. 2 by solving $\boldsymbol{w}_i$ from $(\boldsymbol{A}-\sigma\boldsymbol{I})\boldsymbol{w}_i=\boldsymbol{v}_i$. In this paper, complex shift σ is determined by the specified damping ratio and frequency. Illustration of shift-invert transformation is shown in Fig. 1.

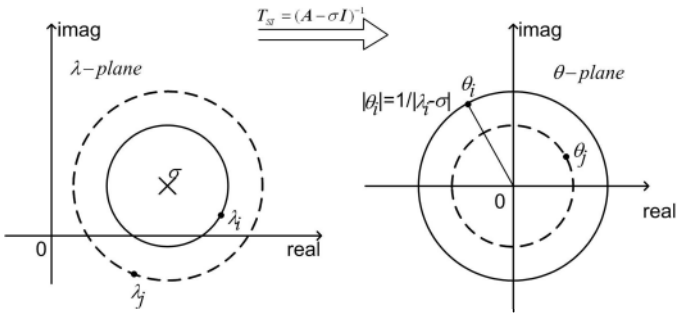

Fig. 1. Illustration of shift-invert transformation with shift point σ

Algorithm 3: Two-sided Arnoldi Algorithm

INPUT: Matrix $\boldsymbol{A}$, initial vectors $\boldsymbol{v}_1$, $\boldsymbol{w}_1$ and reduced matrix dimension k

OUTPUT: Eigenvalues λ_i and corresponding right and left eigenvectors $\boldsymbol{x}_i$ and $\boldsymbol{y}_i$, $i=1,\ldots,k$

1> Apply Algorithm 2 to $\{\boldsymbol{A},\boldsymbol{v}_1\}$ to generate $(\boldsymbol{V}_k,\boldsymbol{H}_k)$, satisfying $\boldsymbol{A}\boldsymbol{V}_k=\boldsymbol{V}_k\boldsymbol{H}_k+h_{k+1,k}\boldsymbol{v}_{k+1}\boldsymbol{e}_k^T$

2> Apply Algorithm 2 to $\{\boldsymbol{A}^T,\boldsymbol{w}_1\}$ to generate $(\boldsymbol{W}_k,\boldsymbol{M}_k)$, satisfying $\boldsymbol{A}^T\boldsymbol{W}_k=\boldsymbol{W}_k\boldsymbol{M}_k+m_{k+1,k}\boldsymbol{w}_{k+1}\boldsymbol{e}_k^T$

3> $\boldsymbol{U}_k=\boldsymbol{W}_k^*\boldsymbol{V}_k$

4> Correct matrix $\boldsymbol{H}_k$ to $\widehat{\boldsymbol{H}}_k=\boldsymbol{H}_k+\boldsymbol{U}_k^{-1}\boldsymbol{W}_k^*h_{k+1,k}\boldsymbol{v}_{k+1}\boldsymbol{e}_k^T$

5> Correct matrix $\boldsymbol{M}_k$ to $\widehat{\boldsymbol{M}}_k^*=\boldsymbol{M}_k^*+m_{k+1,k}^*\boldsymbol{e}_k\boldsymbol{w}_{k+1}^*\boldsymbol{V}_k\boldsymbol{U}_k^{-1}$

6> Compute the eigenvectors $\boldsymbol{x}_{ri}$ and $\boldsymbol{y}_{ri}$ from $\widehat{\boldsymbol{H}}_k$ and $\widehat{\boldsymbol{M}}_k^*$; compute $\boldsymbol{x}_i=\boldsymbol{V}_k\boldsymbol{x}_{ri}$ and $\boldsymbol{y}_i=\boldsymbol{W}_k\boldsymbol{y}_{ri}$; $\widehat{\boldsymbol{H}}_k$ and $\widehat{\boldsymbol{M}}_k^*$ have the conjugated eigenvalues λ_i and λ_i^* once satisfies $\left\|\boldsymbol{A}\boldsymbol{x}_i-\lambda_i\boldsymbol{x}_i\right\|_2<\varepsilon$ or $\left\|\boldsymbol{A}^*\boldsymbol{y}_i-\lambda_i^*\boldsymbol{y}_i\right\|_2<\varepsilon$, $i=1,\ldots,k$

III. Combined TSA-SPA Algorithm

In TSA (see Step 6 in Alg. 3), eigenvectors corresponding to the calculated eigenvalue λ_i can be obtained from vectors $\boldsymbol{x}_{ri}$ and $\boldsymbol{y}_{ri}$ in Krylov subspace, i.e. $\boldsymbol{x}_i=\boldsymbol{V}_k\boldsymbol{x}_{ri}$ and $\boldsymbol{y}_i=\boldsymbol{W}_k\boldsymbol{y}_{ri}$, where $\boldsymbol{x}_{ri}$ and $\boldsymbol{y}_{ri}$ are eigenvectors of matrices $\widehat{\boldsymbol{H}}_k$ and $\widehat{\boldsymbol{M}}_k$, respectively, with dimension $k\times k$. Thus, the sensitivity of eigenvalues to the control parameter in (4) can be rewritten in the following form:

$$\frac{\partial\lambda_i}{\partial p}=\boldsymbol{y}_i^*\frac{\partial\boldsymbol{A}}{\partial p}\boldsymbol{x}_i=\boldsymbol{y}_{ri}^*\boldsymbol{W}_k^*\frac{\partial\boldsymbol{A}}{\partial p}\boldsymbol{V}_k\boldsymbol{x}_{ri} \tag{5}$$

Besides, since TSA reduces Matrix $\boldsymbol{A}$ of *n*-dimension to *k*-dimension matrices $\widehat{\boldsymbol{H}}_k=\boldsymbol{U}_k^{-1}\boldsymbol{W}_k^*\boldsymbol{A}\boldsymbol{V}_k$ and $\widehat{\boldsymbol{M}}_k=\boldsymbol{U}_k\boldsymbol{H}_k\boldsymbol{U}_k^{-1}$ independently, as shown in Steps 4 and 5, it can be readily deduced that

$$\frac{\partial\widehat{\boldsymbol{H}}_k}{\partial p}=\boldsymbol{U}_k^{-1}\boldsymbol{W}_k^*\frac{\partial\boldsymbol{A}}{\partial p}\boldsymbol{V}_k \tag{6}$$

$$\frac{\partial\widehat{\boldsymbol{M}}_k}{\partial p}=\boldsymbol{W}_k^*\frac{\partial\boldsymbol{A}}{\partial p}\boldsymbol{V}_k\boldsymbol{U}_k^{-1} \tag{7}$$

In Step 6, eigenvectors $\boldsymbol{x}_{ri}$ and $\boldsymbol{y}_{ri}$ satisfy that $\widehat{\boldsymbol{H}}_k\boldsymbol{x}_{ri}=\lambda_i\boldsymbol{x}_{ri}$ and $\boldsymbol{y}_{ri}^*\widehat{\boldsymbol{M}}_k=\lambda_i\boldsymbol{y}_{ri}^*$; therefore, from Steps 4 and 5, it can be deduced that eigenvectors $\boldsymbol{x}_{ri}$ and $\boldsymbol{y}_{ri}$ satisfy the following equations, respectively:

$$\boldsymbol{W}_k^*\boldsymbol{A}\boldsymbol{V}_k\boldsymbol{x}_{ri}=\lambda_i\boldsymbol{W}_k^*\boldsymbol{V}_k\boldsymbol{x}_{ri}=\lambda_i\boldsymbol{U}_k\boldsymbol{x}_{ri} \tag{8}$$

$$\boldsymbol{y}_{ri}^*\boldsymbol{W}_k^*\boldsymbol{A}\boldsymbol{V}_k=\lambda_i\boldsymbol{y}_{ri}^*\boldsymbol{W}_k^*\boldsymbol{V}_k=\lambda_i\boldsymbol{y}_{ri}^*\boldsymbol{U}_k \tag{9}$$

Therefore, from (5)-(9), if $\boldsymbol{b}=(\partial\widehat{\boldsymbol{H}}_k/\partial p)\boldsymbol{x}_{ri}$ and $\boldsymbol{c}=(\partial\widehat{\boldsymbol{M}}_k^*/\partial p)\boldsymbol{y}_{ri}$, instead of vectors $\boldsymbol{b}$ and $\boldsymbol{c}$ in Alg. 1, the eigenvalue with the largest residues of the reduced matrix $\boldsymbol{W}_k^*\boldsymbol{A}\boldsymbol{V}_k$ is:

$$\begin{aligned}R_i&=(\boldsymbol{c}^*\boldsymbol{U}_k\boldsymbol{x}_{ri})(\boldsymbol{y}_{ri}^*\boldsymbol{U}_k\boldsymbol{b})\\&=(\boldsymbol{y}_{ri}^*\boldsymbol{W}_k^*\frac{\partial\boldsymbol{A}}{\partial p}\boldsymbol{V}_k\boldsymbol{U}_k^{-1}\boldsymbol{U}_k\boldsymbol{x}_{ri})(\boldsymbol{y}_{ri}^*\boldsymbol{U}_k\boldsymbol{U}_k^{-1}\boldsymbol{W}_k^*\frac{\partial\boldsymbol{A}}{\partial p}\boldsymbol{V}_k\boldsymbol{x}_{ri})\\&=(\boldsymbol{y}_i^*\frac{\partial\boldsymbol{A}}{\partial p}\boldsymbol{x}_i)^2=(\frac{\partial\lambda_i}{\partial p})^2\end{aligned}$$

The largest residue R_i accords with the one obtained by SPA. Therefore, similar to SPA, the proposed TSA-SPA algorithm can also extend DPA to compute the most sensitive eigentriplet. During iterations, vectors $\boldsymbol{b}$ and $\boldsymbol{c}$ are renewed by the right and left eigenvectors $\boldsymbol{x}_{ri}$ and $\boldsymbol{y}_{ri}$, with multiplying matrix $\boldsymbol{W}_k^*\boldsymbol{A}_p\boldsymbol{V}_k$, until eigenvectors $\boldsymbol{x}_{ri}$ and $\boldsymbol{y}_{ri}$ are converged.

The procedure of the proposed algorithm is shown in Alg. 4: the initial eigen-problem is reduced to $(\boldsymbol{N}_k,\boldsymbol{U}_k)$ by using TSA at first, and then the most sensitive eigenvalues are calculated from $(\boldsymbol{N}_k,\boldsymbol{U}_k)$ by SPA. From the algorithm, it can be seen that vectors renewed at each iteration, Steps 5a and 5c, have dimension *k* only, where *k* << *n*, and the linear problems solved in Steps 5c and 5d are also of relatively small dimensions. This boosts computation speed of the proposed algorithm. Moreover, Shift-invert transformation is realized in Step 1 for building the two Krylov subspaces $\boldsymbol{V}_k$ and $\boldsymbol{W}_k$; thus $(\boldsymbol{N}_k,\boldsymbol{U}_k)$ usually contains several eigenvalues around the given shift point. It is

noted that as a rule of thumb, Krylov subspace of size twice that of the reduced matrix $\boldsymbol{N}_k$ is practically sufficient to guarantee convergence to good eigenvalue approximations [30]. Furthermore, sparsity technique can be adopted since both matrix $\boldsymbol{A}$ and matrix $\boldsymbol{A}_p$ are sparse, so the algorithm can be implemented at a fast speed. Finally, with the deflation procedure, the proposed algorithm can readily calculate several other sensitive eigenvalues. It is noted that although QR/QZ methods may be applied directly to $(\boldsymbol{N}_k, \boldsymbol{U}_k)$ to compute all eigenvalues, it still cannot tell which eigenvalue is the most sensitive one, unless sensitivities of all eigenvalues are calculated one by one, which is time consuming when dozens of eigenvalues are obtained. It also needs to be noted that matrix $\boldsymbol{N}_k = \boldsymbol{W}_k^* \boldsymbol{A} \boldsymbol{V}_k$ need not be formed explicitly like SASPA, instead, $\boldsymbol{W}_k^* \boldsymbol{A} \boldsymbol{V}_k$ equals to $\boldsymbol{U}_k \widehat{\boldsymbol{H}}_k$, and thus the matrix multiplication problem can be solved in $k \times k$ -dimension.

Typically, deflation procedures are often employed to make a rank one modification of the original matrix so as to displace the eigenvalue calculated, while keeping all other eigenvalues unchanged. In this paper, the idea of Wielandt deflation method [31] is adopted in the residue index renewal. Instead of modifying the eigenvalue obtained, sensitivity of the associated eigenvalue is changed to zero while keeping sensitivities of other eigenvalues unchanged. Its main advantage is that once an eigenvalue is obtained, it is no longer dominant in the deflation procedure, while the reduced matrix and the spanned subspaces are preserved without any modification. The deflation procedure for residue index renewal is shown in Steps 7 to 10 of Alg. 4. Compared to SASPA, other sensitive eigenvalues are calculated in the same reduced matrix $\boldsymbol{N}_k$, whereas SASPA renews the subspaces in the *n*-dimension complex plane. Therefore, the computational burden of the proposed method can be reduced reasonably. Besides, since matrix $\boldsymbol{A}_p$ may keep sparse, the renewed matrix $\boldsymbol{N}_p$ in Step 10 does not deteriorate its sparseness much.

IV. Simulation Results of A Small Power System

This section describes numerical results of application of the proposed algorithm in the IEEE 9-bus system. Static and dynamic data of the system can be found from [35]. Control parameter p is chosen to be K_{PSS} of PSS installed at generator G_2 with the model shown in Fig. 2, where $\Delta\omega$ is rotor speed deviation, K_{PSS} is PSS gain, T_w, T_1 and T_2 are time constants and V_S is PSS output. This simulation is carried out on PC (1.4-GHz Intel Core Duo, 2GB RAM) and the program is written in Matlab 7.1. This system has 21 state variables. Size of Krylov subspace, reduced matrix dimension *k*, number of wanted eigenvalues l and tolerance ε are set to be 20, 8, 4 and 10^{-5}, respectively. Krylov subspace of size twice that of the reduced matrix $\boldsymbol{N}_k$ is used to practically guarantee convergence to good eigenvalue approximations. Initial vectors are chosen as $\boldsymbol{v}_{r0} = \boldsymbol{w}_{r0} = [1, \ldots, 1] \in R^k$ (cf. Step 3 of Alg. 4).

Algorithm 4: TSA-SPA Algorithm

INPUT: Matrix $\boldsymbol{A}$, $\boldsymbol{A}_p \equiv (\partial \boldsymbol{A}/\partial p)$, shift point σ , eigenvalue estimate s_0 , reduced matrix dimension k , number of computed eigenvalues l and tolerance ε

OUTPUT: Most sensitive eigenvalue λ_i and corresponding eigenvectors $\boldsymbol{x}_i$ and $\boldsymbol{y}_i$, $i = 1, \ldots, l$

1> Execute TSA algorithm to generate $\boldsymbol{V}_k$, $\boldsymbol{W}_k$ and set $\boldsymbol{U}_k = \boldsymbol{W}_k^* \boldsymbol{V}_k$, $\boldsymbol{N}_k = \boldsymbol{W}_k^* \boldsymbol{A} \boldsymbol{V}_k$

2> Compute matrix $\boldsymbol{N}_p = \boldsymbol{W}_k^* \boldsymbol{A}_p \boldsymbol{V}_k$

3> set $m = 0$ and $i = 0$, $\boldsymbol{v}_{r0}, \boldsymbol{w}_{r0} \in \Re^k$ s.t. $\boldsymbol{N}_p \boldsymbol{v}_{r0} \neq 0$ and $\boldsymbol{N}_p^* \boldsymbol{w}_{r0} \neq 0$

4> **while** $i < l$ **do**

5> **while** not converge **do**

a. $\boldsymbol{b}_r = \boldsymbol{N}_p \boldsymbol{v}_{rm} \big/ \left\| \boldsymbol{N}_p \boldsymbol{v}_{rm} \right\|_2$

b. $\boldsymbol{c}_r = \boldsymbol{N}_p^* \boldsymbol{w}_{rm} \big/ \left\| \boldsymbol{N}_p^* \boldsymbol{w}_{rm} \right\|_2$

c. Calculate $\boldsymbol{v}_{rm+1} \in C^k$ from $(s_m \boldsymbol{U}_k - \boldsymbol{N}_k) \boldsymbol{v}_{rm+1} = \boldsymbol{b}_r$

d. Calculate $\boldsymbol{w}_{rm+1} \in C^k$ from $(s_m \boldsymbol{U}_k - \boldsymbol{N}_k)^* \boldsymbol{w}_{rm+1} = \boldsymbol{c}_r$

e. Renew eigenvalue $s_{m+1} = \dfrac{\boldsymbol{w}_{rm+1}^* \boldsymbol{N}_k \boldsymbol{v}_{rm+1}}{\boldsymbol{w}_{rm+1}^* \boldsymbol{U}_k \boldsymbol{v}_{rm+1}}$

f. $\boldsymbol{x}_{ri} = \boldsymbol{v}_{rm+1} \big/ \left\| \boldsymbol{v}_{rm+1} \right\|_2$ and $\boldsymbol{y}_{ri} = \boldsymbol{w}_{rm+1} \big/ \left\| \boldsymbol{w}_{rm+1} \right\|_2$

g. If $\left\| \boldsymbol{N}_k \boldsymbol{x}_{ri} - s_{m+1} \boldsymbol{U}_k \boldsymbol{x}_{ri} \right\|_2 < \varepsilon$, then $i = i + 1$ and eigenvalue of matrix $\boldsymbol{N}_k$ is $\lambda_i = s_{m+1}$, compute the eigenvectors $\boldsymbol{x}_i = \boldsymbol{V}_k \boldsymbol{x}_{ri}$ and $\boldsymbol{y}_i = \boldsymbol{W}_k \boldsymbol{y}_{ri}$; λ_i is eigenvalue of Matrix ***A,*** once it satisfies $\left\| \boldsymbol{A} \boldsymbol{x}_i - \lambda_i \boldsymbol{x}_i \right\|_2 < \varepsilon$ or $\left\| \boldsymbol{A}^* \boldsymbol{y}_i - \lambda_i^* \boldsymbol{y}_i \right\|_2 < \varepsilon$

h. set $m = m + 1$

6> **end while**

7> Compute the matrix $\boldsymbol{T}_i = \dfrac{1}{\boldsymbol{y}_{ri}^* \boldsymbol{x}_{ri}} \boldsymbol{x}_{ri} \boldsymbol{y}_{ri}^*$,

8> Compute the matrix $\boldsymbol{T}_i = \boldsymbol{I} - \boldsymbol{T}_i$

9> Update sensitivity index $\boldsymbol{A}_p = \boldsymbol{T}_i \boldsymbol{A}_p \boldsymbol{T}_i$

10> Compute the matrix $\boldsymbol{N}_p = \boldsymbol{W}_k^* \boldsymbol{A}_p \boldsymbol{V}_k$

11> **end while**

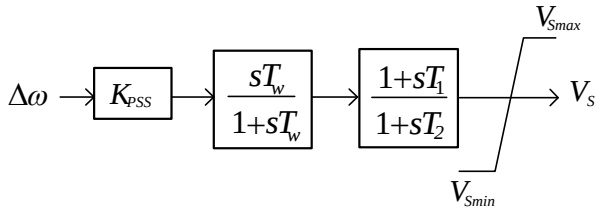

Fig. 2. PSS model installed on Generator G_2

This study first sets the shift point of damping ratio of 50% with frequency 2.0Hz, i.e. $\sigma = (-7.26, 12.57)$. The proposed method succeeds in computing the four most sensitive eigenvalues in the first 4 iterations, i.e. three times of deflation processes, as shown in Table I. Their sensitivities and state variables with maximum participation factor (PF) for these modes are tabulated (in Table I); V_{R2} denotes exciter input of generator G_2; δ_2, δ_3 denote rotor angles of generators G_2 and G_3, respectively; E'_{q_1} denotes the q-axis component of the voltage behind transient reactance X'_d.

The proposed method requires 0.1560s CPU time to calculate four sensitive eigenvalues while SASPA [36] takes 0.3120s to fulfill the same task. Comparison of computation costs is given in Table II. In the Iteration column, the item in the summation denotes the number of iteration steps required for obtaining each sensitive eigenvalue. For TSA-SPA, the first item of the summation denotes the number of iterations in n-dimension for building Krylov subspaces for a given size of reduced matrix k and each item in parentheses denotes iterations executed for obtaining one sensitive eigenvalue from the reduced matrix. It can be seen that the number of LU factorizations is one in TSA-SPA, and it increases with the number of wanted eigenvalues in SASPA. Thus, TSA-SPA can save computational expense in LU factorization. Also, as shown in Iteration of Table II, even when the number of wanted eigenvalues (l) rises, TSA-SPA needs the same number of iterations for building the Krylov subspaces. Besides, it can be observed that iterations in k dimensions for finding the most sensitive eigenvalues are not more than that needed in SASPA, in which iterations are executed in n dimensions (each item in the summation denotes the number of iterations for obtaining one sensitive eigenvalue). This property enables TSA-SPA to have relatively stable computation cost. The reason for TSA-SPA having this property could be that TSA-SPA focuses on a specific area, as dictated by shift point, while SASPA scans the whole complex plane, using the shift strategy. It also needs to be mentioned that the CPU time for obtaining sensitive eigenvalues by TSA-SPA does not show obvious variation with change in the shift point, while for SASPA, the CPU time may vary with initial shift selection. For example, if the number of wanted eigenvalues is 4 and the initial shifts are selected as (-7.26, 12.57), (0, 1) and (1, 0), the CPU time is almost constant for TSA-SPA, while for SASPA, it takes 0.3120s, 0.4212s and 0.5304s, respectively.

TABLE I
SENSITIVE EIGENVALUES FOR K_{PSS} OF PSS INSTALLED AT GENERATOR G_2

Deflation number	Eigenvalue (symbol)	Sensitivity	State variable with max. PF
0	-5.8025+j7.7888 (Dot ●)	-0.313 - j0.151	V_{R2}
1	-0.0339+j8.5278 (Circle ○)	0.148 + j0.153	δ_2
2	-0.6452+j12.7459 (Star *)	0.074 - j0.001	δ_3
3	-0.4487+j1.1950 (Diamond ◆)	-0.003-j0.0145	E'_{q_1}

TABLE II
COMPARISON OF COMPUTATION COST OF TSA-SPA WITH SASPA

Method	k	l	Num. of LU factor-izations	Iteration	CPU time (s)
TSA-SPA	8	4	1	33+(4+6+4+6)	0.1560
	8	5	1	33+(4+6+4+6+10)	0.1872
	8	6	1	33+(4+6+4+6+10+2)	0.2340
	10	4	1	35+(4+5+6+10)	0.1716
	10	5	1	35+(4+5+6+10+6)	0.2028
	10	6	1	35+(4+5+6+10+6+6)	0.2596
SASPA	-	4	16	4+7+9+9	0.3120
	-	5	19	4+7+9+9+10	0.4056
	-	6	27	4+7+9+9+10+7	0.5148

Robustness of the proposed method is tested by comparing its convergence area (Fig. 4) with SPA (Fig. 3). A symbol at point (x, y) in complex plane means that the respective methods start with an initial eigenvalue estimate $s_0 = x + jy$ and converge to the eigenvalue corresponding to the symbol in Table I. Symbol × in Fig. 3 denotes less sensitive eigenvalues, i.e. eigenvalues with absolute value of sensitivity less than 0.05. It can be observed that SPA can find several sensitive eigenvalues, depending on the selection of shift. It is noted that SPA can find another real sensitive eigenvalue -16.4684 (with sensitivity 0.203 marked by ■). With change in the shift around the eigenvalue, TSA-SPA can also find it, as seen in Fig. 5. For TSA-SPA, in all cases, the proposed method obtains the same sensitive eigenvalue (-5.8025 + j7.7888), which is closer to the shift point, and there are no bad computation results (less sensitive eigenvalues). The proposed method is clearly much more robust to the initial shift selection than SPA.

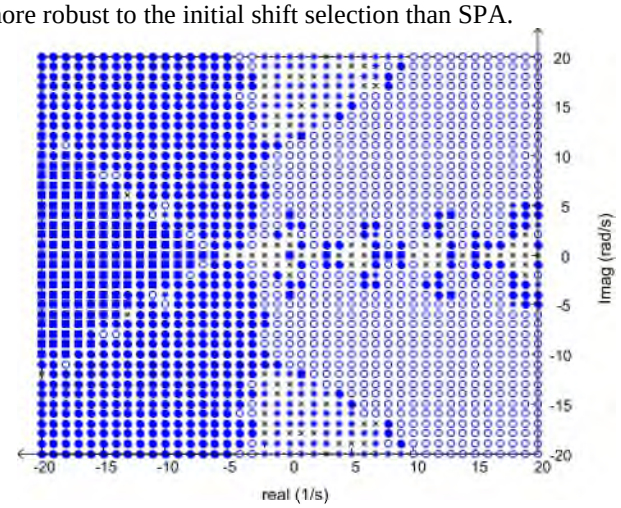

Fig. 3. Sensitive eigenvalue convergence areas for SPA in the 9-bus system

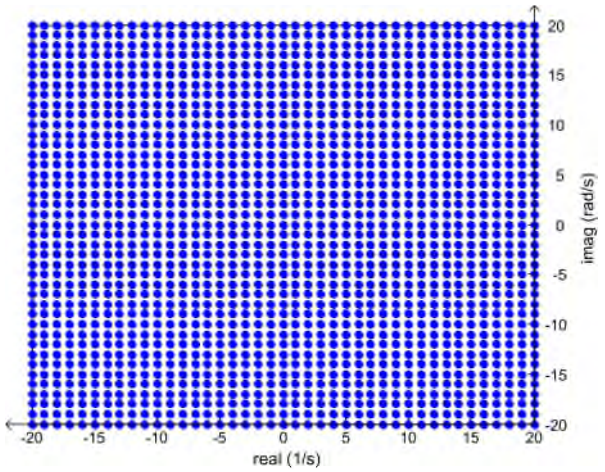

Fig. 4. Sensitive eigenvalue convergence areas for TSA-SPA in the 9-bus system

Besides, the influence of the change of shift point on the calculated eigenvalues is also studied (Fig. 5). A symbol at point (*x*, *y*) in complex plane means the proposed method starts with a shift point $\sigma = x + \mathrm{j}y$ and converges to the eigenvalue corresponding to the symbol in Table I. Similar to Figs. 3 and 4, symbol × in Fig. 5 denotes the less sensitive eigenvalues, i.e. eigenvalues with absolute value of sensitivity less than 0.05. For the shift-invert transformation, eigenvalues obtained are symmetric to the real axis and shift point σ has great influence on eigenvalues computed, but the sensitive eigenvalues in a region can always be found with reasonable shift points. If the sensitive eigenvalues locate far from the shift points, no eigenvalue will be obtained, i.e. symbol× in Fig. 5. In other words, the shift points can be used to select the region to be studied, demonstrating the flexibility of the proposed algorithm, and it would be a very useful function, especially in large-scale power system.

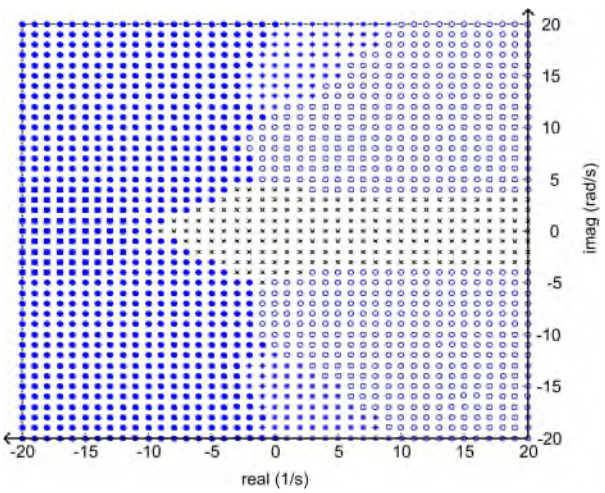

Fig. 5. Eigenvalues obtained by TSA-SPA for different shift points in the 9-bus system

V. Simulation Results of Large-Scale Power Systems

A. A large-scale power system with 37587 states

In this section, TSA-SPA is applied in a large-scale power system which represents the entire eastern US-Canada interconnected system. This system has 37587 state variables while the augmented Jacobian matrix has the order of 108331. Control parameter of the system is selected to be gain K_{PSS} of PSS at generator G_{87456}. The PSS model is selected as IEEEST [37], as shown in Fig. 6. In this study, the first block of the PSS, the torsional filter, is neglected and the input signal of the PSS is the generator's rotor speed deviation. Based on available settings of the PSS in the system, simulation is carried out, with the result showing that TSA-SPA can selectively compute sensitive eigenvalues with proper shift selection.

This simulation is carried out on PC (1.4-GHz Intel Core Duo, 2GB RAM) and the program is written in Fortran 90 based on the available open-source code of ARPACK. Initial vectors are chosen as $\boldsymbol{v}_{r0} = \boldsymbol{w}_{r0} = [1,\ldots,1] \in R^k$, and tolerance is set to be 10^{-6}. Similar to the setting of the small test system, the size of Krylov subspace is also set to be twice that of the reduced matrix. The Krylov subspace dimension is set to be 120 and the reduced matrix has the order of 60. The region of frequencies from 0.2Hz to 2.0Hz and damping ratios from 1% to 8%, which includes both inter-area and local modes, is studied. All eigenvalues in this region are calculated with IRAM, using different shift points. 1558 eigenvalues are calculated; the top ten sensitive eigenvalues are as tabulated in Table III. TSA-SPA is then applied with a shift point of damping ratio of 5% with frequency 1.7Hz, i.e. (-0.5347, 10.6814), to calculate the sensitive local modes. The computing time is 191s for computing three most sensitive modes by deflation procedure, as shown in Table IV. TSA-SPA can selectively calculate these sensitive eigenvalues out of 1558 eigenvalues locating in the region of interest.

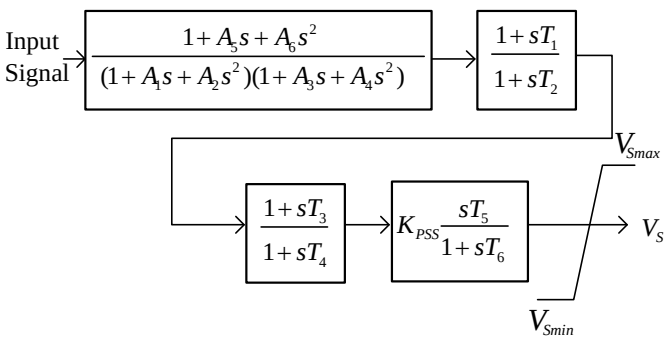

Fig. 6. IEEEST PSS model

The influence of the change of eigenvalue estimate on calculated eigenvalues in case of large-scale power systems is also studied. As shown in Fig. 7, TSA-SPA is also very robust to the eigenvalue estimate shifting, similar to the small power system in Section IV. Since eigenvalues obtained are symmetric to the real axis, only eigenvalues with positive imaginary parts are illustrated.

The capability of TSA-SPA to find the most sensitive inter-area modes is also tested. When a shift point with damping ratio of 5% and frequency of 0.4Hz, i.e. (-0.1258, 2.5133), is assumed, TSA-SPA can obtain the four highest sensitivity inter-area modes sequentially by deflation procedure (as shown in Table V). When a suitable shift point is selected, eigenvalues with high sensitivity and which are closer to the shift point can be obtained directly. The obtained eigenvalues of inter-area modes for different shift points are shown in Fig. 8. Although highly sensitive eigenvalues in a region can also be calculated sequentially, using deflation procedure, selection of the shift point is an effective means to determine the region for eigenvalue searching.

TABLE III
SENSITIVE EIGENVALUES (ELECTROMECHANICAL MODES) OBTAINED BY IRAM IN THE SYSTEM WITH 37587 STATES

Ranking number	Eigenvalue	Eigenvalue sensitivity (10^{-2})
1	-0.9130 + j10.8538	-6.5792 - j0.0335
2	-0.1651 + j11.4636	-1.8722 - j0.0211
3	-0.9171 + j 7.3984	-1.1613 - j0.0114
4	-0.8253 + j6.3804	-0.5710 - j0.0143
5	-0.7280 + j7.7364	-0.1954 - j0.0342
6	-0.1134 + j2.6185	-0.1346 - j0.0415
7	-0.1231 + j2.3018	0.0492 - j0.0043
8	-0.2225 + j1.6383	-0.0234 + j0.0051
9	-0.4807 + j6.2210	-0.0217 - j0.0074
10	-0.1261 + j2.8236	-0.0182 + j0.0055

TABLE IV
SENSITIVE LOCAL MODES OBTAINED BY TSA-SPA BY SHIFTING POINT (-0.5347, 10.6814) IN THE SYSTEM WITH 37587 STATES

Deflation number	Eigenvalue	Sensitivity (10^{-2})	State variable with max PF
0	-0.9130 + j10.8538	-6.5792 - j0.0335	ω_{87456}
1	-0.1651 + j11.4636	-1.8722 - j0.0211	ω_{87458}
2	-0.6325 + j10.5189	-0.0123 - j0.0029	ω_{62032}

TABLE V
SENSITIVE INTER-AREA MODES OBTAINED BY TSA-SPA BY SHIFTING POINT (-0.1258, 2.5133) IN THE SYSTEM WITH 37587 STATES

Number	Eigenvalue	Sensitivity (10^{-2})	State variable with max PF
1	-0.1134 + j2.6185 (Diamond ◆)	-0.1346 - j0.0415	ω_{87456}
2	-0.1231 + j2.3018 (Square ■)	0.0492 - j0.0043	ω_{87456}
3	-0.2225 + j1.6383 (Triangle ▲)	-0.0234 + j0.0051	ω_{87456}
4	-0.1261 + j2.8236 (Dot ●)	-0.0182 + j0.0055	ω_{87456}

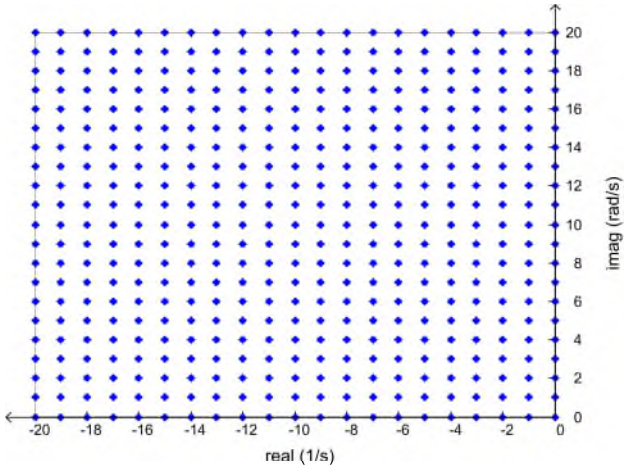

Fig. 7. Sensitive eigenvalue convergence areas for TSA-SPA in the system with 37587 states

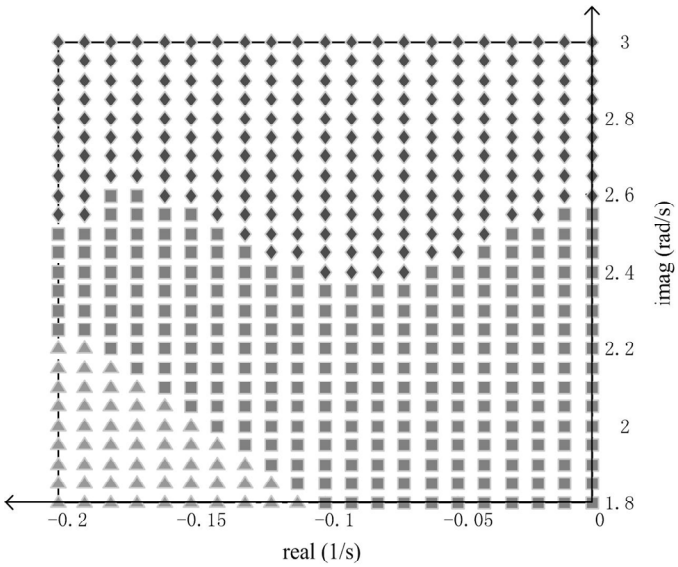

Fig. 8. Sensitive inter-area modes obtained by TSA-SPA for different shift points in the system with 37587 states

B. A large-scale power system with 4381 states

A large-scale system in the interconnected North-Central China region is studied, using TSA-SPA and SASPA. This system has 4381 states with 654 generators and 1645 load nodes. The gain K_{PSS} of PSS at generator G_{EGAO}, which locates at the centre of the grid with capacity of 2600MW, is selected as the control parameter of the system. The PSS model is IEEEST, as shown in Fig. 6, and the input signal is generator's rotor speed deviation. Comparison of computational requirement of TSA-SPA and SASPA for finding the four most sensitive eigenvalues is studied. It is noted that the available open source SASPA is written in Matlab language and copes with the dense state matrix; thus it takes a long time to solve the linear problem, i.e. Steps 5 and 6 in Alg. 1. To speed up the computation, the sparse descriptor in [38] is applied. Besides, SASPA is implemented in Matlab 7.1, in which ARPACK is a library of Matlab software. Therefore, TSA-SPA and SASPA, implemented in Fortran 90 and MATLAB 7.1, respectively, lend themselves to reasonable comparison.

A shift point with damping ratio of 3% and frequency of 0.7 Hz, i.e. (-0.132, 4.3982), is assumed for both TSA-SPA and SASPA. SASPA spans the subspace with the order of 4 while TSA-SPA spans the Krylov subspaces of dimension 120 with the reduced matrix having the order of 60. The achieved sensitive eigenvalues are shown in Table VI. Number of iterations required for building the subspace in TSA-SPA is greater than what is required for SASPA but the computation speed of TSA-SPA is over 14 times faster than that of SASPA. It is clear that TSA-SPA has the advantage in computational speed when sensitive eigenvalues in a specified region are of interest, although SASPA might be faster when scanning of the whole complex plane is required. TSA-SPA focuses on a specific area by spanning the subspaces around a given shift with assumption of adequate subspace size while SASPA scans the whole complex plane by shift strategy and spans the subspaces subsequently, with high enough precision, which can guarantee the accuracy of the obtained results. Besides, TSA-SPA can find several sensitive eigenvalues around a given shift while SASPA can locate the more scattered sensitive eigenvalues, as shown in Table VI. With enough scanning, TSA-SPA can also identify an adequate number of scattered sensitive eigenvalues associated with control parameters. This illustrates that TSA-SPA can provide an effective tool to compute sensitive eigenvalues in a specified region of interest in complex plane.

TABLE VI
COMPARISON OF TSA-SPA AND SASPA ON SENSITIVE INTER-AREA MODES CALCULATION IN THE SYSTEM WITH 4381 STATES

Method	Num.	Eigenvalue	Sensitivity (10^{-2})	Iteration	Time (s)
TSA-SPA	1	-0.0573 + j4.1627	-8.1206 - j0.0215	342	12
	2	-0.0425 + j4.3658	-4.3395 - j0.0412		
	3	-0.1074 + j4.1354	-0.0536 - j0.0023		
	4	-0.1241 + j4.6528	-0.04232 + j0.0003		
SASPA	1	-0.0425 + j4.3658	-4.3395 - j0.0412	197	174
	2	-0.0572 + j4.1626	-8.1206- j0.0215		
	3	-1.5846 + j2.3652	-0.1135- j0.0321		
	4	-0.6578 + j7.6984	-0.2576- j0.0076		

VI. CONCLUSION

The proposed TSA-SPA algorithm is effective and robust in computing sensitive eigenvalues to control parameters. Pre-condition transformation enables the proposed method to focus on sensitive eigenvalues in a specified region of interest in complex plane so that electromechanical oscillation modes and/or control modes can be found at a satisfactory speed and reliability. Other sensitive eigenvalues in the region can also be calculated with the deflation technique, which enhances the flexibility of stability analysis of power systems. Numerical experiments confirm that the proposed method is very reliable for arbitrary initial eigenvalue estimate. The proposed method also circumvents the risk of breakdown occurrence in the nonsymmetric Lanczos method.

REFERENCES

[1] F. P. Demello and C. Concordia, "Concepts of synchronous machine stability as affected by excitation control," *IEEE Trans. Power App. Syst.*, vol. PAS-88, pp. 316-329, 1969.

[2] C.Y. Chung, L. Wang, F. Howell and P. Kundur, "Generation rescheduling methods to improve power transfer capability constrained by small-signal stability," *IEEE Trans. Power Syst.*, vol. 19, no. 1, pp. 524-530, Feb. 2004.

[3] D. A. Pierre, "A perspective on adaptive control of power systems," *IEEE Trans. Power Syst.*, vol. 2, pp. 387-396, May 1987.

[4] G. Rogers, *Power System Oscillations*. Norwell, MA: Kluwer, 2000.

[5] J. Z. Yang, C. W. Liu, and W. G. Wu, "A hybrid method for the estimation of power system low-frequency oscillation parameters," *IEEE Trans. Power Syst.*, vol. 22, pp. 2115-2123, Nov 2007.

[6] P. Kundur, M. Klein, G. J. Rogers, and M. S. Zywno, "Application of power system stabilizers for enhancement of overall system stability," *IEEE Trans. Power Syst.*, vol. 4, pp. 614-626, May 1989.

[7] D. R. Ostojic, "Stabilization of multimodal electromechanical oscillations by coordinated application of power system stabilizers," *IEEE Trans. Power Syst.*, vol. 6, pp. 1439-1445, Nov 1991.

[8] P. Pourbeik and M. J. Gibbard, "Simultaneous coordination of power system stabilizers and FACTS device stabilizers in a multimachine power system for enhancing dynamic performance," *IEEE Trans. Power Syst.*, vol. 13, pp. 473-478, May 1998.

[9] D. J. Trudnowski, D. A. Pierre, J. R. Smith, and R. Adapa, "Coordination of multiple adaptive PSS units using a decentralized control scheme," *IEEE Trans. Power Syst.*, vol. 7, pp. 294-300, Feb 1992.

[10] G. E. Boukarim, S. P. Wang, J. H. Chow, G. N. Taranto, and N. Martins, "A comparison of classical, robust, and decentralized control designs for multiple power system stabilizers," *IEEE Trans. Power Syst.*, vol. 15, pp. 1287-1292, Nov 2000.

[11] C. Y. Chung, K. W. Wang, C. T. Tse, X. Y. Bian, and A. K. David, "Probabilistic eigenvalue sensitivity analysis and PSS design in multimachine systems," *IEEE Trans. Power Syst.*, vol. 18, pp. 1439-1445, Nov 2003.

[12] N. Mithulananthan, C. A. Canizares, J. Reeve, and G. J. Rogers, "Comparison of PSS, SVC, and STATCOM controllers for damping power system oscillations," *IEEE Trans. Power Syst.*, vol. 18, pp. 786-792, May 2003.

[13] B. Chaudhuri, R. Majumder, and B. C. Pal, "Application of multiple-model adaptive control strategy for robust damping of interarea oscillations in power system," *IEEE Trans. Control Syst. Technol.*, vol. 12, pp. 727-736, Sep 2004.

[14] R. A. Ramos, L. F. C. Alberto, and N. G. Bretas, "A new methodology for the coordinated design of robust decentralized power system damping controllers," *IEEE Trans. Power Syst.*, vol. 19, pp. 444-454, Feb 2004.

[15] L. J. Cai and I. Erlich, "Simultaneous coordinated tuning of PSS and FACTS damping controllers in large power systems," *IEEE Trans. Power Syst.*, vol. 20, pp. 294-300, Feb 2005.

[16] D.P. Ke, C.Y. Chung and Y. Xue, "An eigenstructure-based performance index and its application to control design for damping intera-area oscillations in power systems," *IEEE Trans. Power Syst.*, vol. 26, pp. 2317-2380, Nov. 2011.

[17] D.P. Ke and C.Y. Chung, "An inter-area mode oriented pole-shifting method with coordination of control effects for robust tuning of power oscillation damping controllers", *IEEE Trans. Power Syst., Early Access*.

[18] N. Uchida and T. Nagao, "A new eigen-analysis method of steady-state stability studies for large power systems: S matrix method," *IEEE Trans. Power Syst.*, vol. 3, pp. 706-714, May 1988.

[19] Y. Q. Lin, L. Bao, and Y. M. Wei, "Model-order reduction of large-scale second-order MIMO dynamical systems via a block second-order Arnoldi method," *Int. J. Comput. Math.*, vol. 84, pp. 1003-1019, 2007.

[20] H. S. Najafi and E. Khaleghi, "A new restarting method in the Arnoldi algorithm for computing the eigenvalues of a nonsymmetric matrix," *Appl. Math. Comput.*, vol. 156, pp. 59-71, Aug 25 2004.

[21] D. Yang and K. Ajjarapu, "Critical eigenvalues tracing for power system analysis via continuation of invariant subspaces and projected Arnoldi method," *IEEE Trans. Power Syst.*, vol. 22, pp. 324-332, Feb 2007.
[22] Y. Saad, "Chebyschev acceleration techniques for solving nonsymmetric eigenvalue problems," *Math.Comp.*, vol. 42, pp. 567-588, 1984.
[23] B. Nouromid, B. N. Parlett, and R. L. Taylor, "Lanczos versus subspace iteration for solution of eigenvalue problems," *Int. J. Numer. Methods Eng.*, vol. 19, pp. 859-871, 1983.
[24] L. Wang and A. Semlyen, "Application of sparse eigenvalue techniques to the small-signal stability analysis of large power systems," *IEEE Trans. Power Syst.*, vol. 5, pp. 635-642, May 1990.
[25] N. Martins, L. T. G. Lima, and H. J. C. P. Pinto, "Computing dominant poles of power system transfer functions," *IEEE Trans. Power Syst.*, vol. 11, pp. 162-167, Feb 1996.
[26] J. Rommes and N. Martins, "Computing large-scale system eigenvalues most sensitive to parameter changes, with applications to power system small-signal stability," *IEEE Trans. Power Syst.*, vol. 23, pp. 434-442, May 2008.
[27] J. Rommes and N. Martins, "Efficient computation of multivariable transfer function dominant poles using subspace acceleration," *IEEE Trans. Power Syst.*, vol. 21, pp. 1471-1483, Nov 2006.
[28] J. Rommes and N. Martins, "Efficient computation of transfer function dominant poles using subspace acceleration," *IEEE Trans. Power Syst.*, vol. 21, pp. 1218-1226, Aug 2006.
[29] J. Cullum and T. Zhang, "Two-sided Arnoldi and nonsymmetric Lanczos algorithms," *SIAM J. Matrix Anal. Appl.*, vol. 24, pp. 303-319, Dec 19 2002.
[30] R. B. Lehoucq, D. C. Sorensen and C. Yang, "ARPACK User's Guide: Solution of large-scale eigenvalue problems by implicitly restarted arnoldi methods," *SIAM: Philadelphia, PA*, 1998.
[31] Y. Saad, *Numerical Methods for Large Eigenvalue Problems.* Manchester: Manchester University Press, 1992.
[32] D. Kressner, *Numerical methods for general and structured eigenvalue problems*. Berlin: Springer, 2005.
[33] K. Meerbergen and D. Roose, "Matrix transformations for computing rightmost eigenvalues of large sparse non-symmetric eigenvalue problems," *SIMA. J. Numer. Anal.*, vol. 16, pp. 297-346, Jul 1996.
[34] R. B. Morgan, "On restarting the Arnoldi method for large nonsymmetric eigenvalue problems," *Mathematics of Computation*, vol. 65, pp. 1213-1230, Jul 1996.
[35] Peter W. Sauer and M. A. Pai, *Power System Dynamics and Stability*. Upper Saddle River, N.J.: Prentice Hall, 1998.
[36] J. Rommes, *Available from http://sites.google.com/site/rommes/software*.
[37] P. T. Inc, "PSS/E Program Operation Manual," 2001.
[38] P. Kundur, *Power System Stability and Control.* McGraw-Hill, Inc., 1994.

C. Y. Chung (M'01–SM'07) received the B.Eng. degree (with First Class Honors) and the Ph.D. degree in electrical engineering from The Hong Kong Polytechnic University, Hong Kong, China, in 1995 and 1999, respectively. After his Ph.D. graduation, he worked in Department of Electrical and Computer Engineering at the University of Alberta, Edmonton, AB, Canada, and Powertech Labs, Inc., Surrey, BC, Canada. Currently, he is the Convener of the Power Systems Research Group and an Associate Professor in the Department of Electrical Engineering, The Hong Kong Polytechnic University. His research interests include power system stability/control, planning and operation, computational intelligence applications and power markets.

Dr Chung was the Chair of IEEE Hong Kong Joint Chapter of PES/IAS/PELS/IES in 2007-2009. During his tenure as chair, the Joint Chapter received the 2008 IEEE PES Outstanding Small Chapter Award and the 2009 IEEE IAS Outstanding Small Joint Chapter Award. He was the Technical Chairman of IET APSCOM2009 International Conference, and Honorary Secretary of IEEE DRPT2004 International Conference and IEEE IAS 2005 Annual Meeting. Currently, he is the Chair of IEEE Hong Kong Section.

Bo Dai (M'02) was born in Beijing, China, in February 1977. She received the M.Sc. and Ph.D. degrees in electrical engineering from North China Electric Power University, Beijing, China, in 2002 and 2009 respectively.

Currently, she is working in Department of Electrical Engineering, The Hong Kong Polytechnic University Her research interests include power system stability and control.