

---

---

**Universiteit Utrecht**



*Department  
of Mathematics*

**Computing large-scale system  
eigenvalues most sensitive to parameter  
changes, with applications to power  
system small-signal stability**

by

**Joost Rommes and Nelson Martins**

---

---

Preprint

nr. 1364

June, 2007

---

---

# Computing large-scale system eigenvalues most sensitive to parameter changes, with applications to power system small-signal stability

Joost Rommes\* and Nelson Martins†

June, 2007

## Abstract

This paper describes a new algorithm, named the Sensitive Pole Algorithm, for the automatic computation of the eigenvalues (poles) most sensitive to parameter changes in large-scale system matrices. The effectiveness and robustness of the algorithm in tracing root-locus plots is illustrated by numerical results from the small-signal stability analysis of realistic power system models. The algorithm can be used in many other fields of engineering that also study the impact of parametric changes to linear system models.

## 1 Introduction

The transient response of a closed-loop linear system is strongly related to the location of the closed-loop poles, which depends on the value of the loop gain. It is therefore important for the designer to know how the closed-loop poles move in the  $s$ -plane as the loop gain (or any other relevant control loop parameter) is varied [23, 13]. The closed-loop poles are the eigenvalues of the system state matrix. The classical root-locus method is a graphical method that enables the designer to trace the closed-loop poles trajectories from the open-loop poles and zeros as the loop gain is changed [23], and helps to understand the effect of feedback and compensation on the closed-loop poles. The classical graphical method was intended for small system models, and has long been superseded by the direct trace of the root-locus plots by repeated application of the QR eigenvalue method [7] for a sequence of parametric changes. Since the tracing of root-loci requires repeated computation of the poles, and the system matrices can be very large, there is need for efficient eigenvalue methods. In this paper, the Sensitive Pole Algorithm (SPA) is presented for the computation of the most sensitive poles of large-scale systems.

Frequency response techniques are much employed in power system oscillation damping control [12, 15, 27, 26, 2, 14, 19, 17, 5, 16, 4, 3, 1, 32] and have the advantage of allowing verification through generator and excitation control field tests. Modal analysis (shapes, participations), transfer function pole residues, root-locus plots, and time response to a pulse or step, have also been extensively used for placing and tuning of power system stabilizers (PSSs), as well as for Power Oscillation Damping (POD) controllers that have been added to HVDC links and FACTS devices. The coordinated PSS design considering multiple operating scenarios, may be carried out much more automatically when employing the methods in [26, 5]. The design methods in [26, 5] would also greatly benefit from the availability of an efficient eigenvalue method to trace multi-parameter root-locus plots, becoming then applicable to large-scale power systems.

Detailed stability models for the North American eastern interconnection easily reach 25,000 states [16, 33], and are far too large to have their full eigensolution produced by a QR routine

---

\*Work performed at Mathematical Institute, Utrecht University, POBox 80010, 3508 TA, Utrecht, The Netherlands, <http://www.math.uu.nl/people/rommes>. Correspondence to [rommes@gmail.com](mailto:rommes@gmail.com)

†CEPEL, P.O.Box 68007, Rio de Janeiro, RJ - 20001 - 970, Brazil, [nelson@cepel.br](mailto:nelson@cepel.br)

[7] (since the computational (memory) costs depend cubically (quadratically) on the number of states). The much smaller Brazilian interconnection (BIPS) model currently has already more than 3,000 states, and its full eigensolution by the QR method requires about 10 minutes of CPU time on a powerful PC. The use of the QR method to trace root-locus plots, requiring thirty or more eigensolutions per set of parameters being changed, is therefore even for the moderately-sized BIPS model not a practical alternative. Obtaining rough traces of some critical root-locus branches of large power system models, using existing partial eigensolution methods that are not focused into this specific problem, can be very laborious [17, 5] and leads to barely acceptable results. A sparse eigenanalysis method that could efficiently compute the set of most sensitive poles for single or multiple parameter changes is therefore of high interest to power system dynamics and control as well as to other areas of engineering.

The Sensitive Pole Algorithm (SPA) presented in this paper is a new method for the computation of the poles most sensitive to parametric changes. Because it operates on the sparse descriptor matrices of the system (having a set of differential-algebraic equations) and computes the most sensitive poles automatically, it is very useful in root-locus studies of large-scale systems. The authors are not aware of any other eigensolution method, applicable to large-scale systems, for the computation of the most sensitive eigenvalues specifically. The robust performance of SPA is illustrated by numerical experiments with realistic power system models.

The outline of the paper is as follows. In Section 2 an overview of transfer function poles, eigenvalue problems, and eigenvalue derivatives is given. The Sensitive Pole Algorithm (SPA) is described in Section 3. Applications of SPA to large-scale practical power system models, showing the effectiveness and robustness of the SPA, are presented in Section 4. Section 5 concludes.

## 2 Transfer function poles, eigenvalues, and derivatives

The motivation for this paper comes mainly from dynamical systems  $(E, A, B, C, D)$ , that are of the form

$$\begin{cases} E\dot{\mathbf{z}}(t) &= A\mathbf{z}(t) + B\mathbf{u}(t) \\ \mathbf{o}(t) &= C^*\mathbf{z}(t) + D\mathbf{u}(t), \end{cases} \quad (1)$$

where  $A, E \in \mathbb{R}^{n \times n}$ ,  $E$  singular,  $B \in \mathbb{R}^{n \times m}$ ,  $C \in \mathbb{R}^{p \times n}$ ,  $\mathbf{z}(t) \in \mathbb{R}^n$  is the state vector,  $\mathbf{u}(t) \in \mathbb{R}^m$  is the input vector,  $\mathbf{o}(t) \in \mathbb{R}^p$  is the output vector, and  $D \in \mathbb{R}^{p \times m}$ . The corresponding transfer function  $H : \mathbb{C} \rightarrow \mathbb{C}^{p \times m}$  is defined as

$$H(s) = C^*(sE - A)^{-1}B + D, \quad (2)$$

with  $s \in \mathbb{C}$ . If  $m = p = 1$ , the system (1) is called a single-input single-output (SISO) system. If  $m, p > 1$ , system (1) is called a multi-input multi-output (MIMO) system. Throughout this paper,  $M^*$  denotes the complex-conjugate transpose of a matrix  $M$ .

The eigenvalues  $\lambda_i \in \mathbb{C}$  of the matrix pencil  $(A, E)$  are the poles of transfer function (2). Assuming that the pencil is nondefective, the right and left eigenvectors  $\mathbf{x}_i$  and  $\mathbf{y}_i$  corresponding to finite eigenvalues  $\lambda_i$  can be scaled so that  $\mathbf{y}_i^* E \mathbf{x}_i = 1$ . Furthermore, right and left eigenvectors corresponding to distinct eigenvalues are  $E$ -orthogonal:  $\mathbf{y}_i^* E \mathbf{x}_j = 0$  for  $i \neq j$ . The transfer function  $H(s)$  can be expressed as a sum of residue matrices  $R_i \in \mathbb{C}^{p \times m}$  over finite first-order poles [11]:

$$H(s) = \sum_{i=1}^{\tilde{n}} \frac{R_i}{s - \lambda_i} + D,$$

where the residues  $R_i$  are

$$R_i = (C^* \mathbf{x}_i)(\mathbf{y}_i^* B),$$

$\tilde{n} \leq n$  is the number of finite first-order poles, and the contribution of poles at infinity is assumed to be zero.

Let  $p \in \mathbb{R}$  be a parameter (e.g., gains or time constants of PSSs and PODs), and let  $A(p)$  and  $E(p)$  be matrices that depend on  $p$ . It is well known that the derivative of an eigenvalue  $\lambda$  of the

pencil  $(A(p), E(p))$ , with left and right eigenvectors  $\mathbf{y} \equiv \mathbf{y}(p)$  and  $\mathbf{x} \equiv \mathbf{x}(p)$ , to a parameter  $p$  is given by [21, 9]

$$\frac{\partial \lambda}{\partial p} = \frac{\mathbf{y}^* (\frac{\partial A}{\partial p} - \lambda \frac{\partial E}{\partial p}) \mathbf{x}}{\mathbf{y}^* E \mathbf{x}}. \quad (3)$$

The derivative (3) is usually called the sensitivity (coefficient) of  $\lambda$ . Note that the sensitivity of an eigenvalue can be real or complex, depending on whether the corresponding eigenvectors are real or complex. In the remainder of this paper it is assumed that  $\frac{\partial E}{\partial p} = 0$  (except for Section 3.4). In that case, with  $\mathbf{y}$  and  $\mathbf{x}$  scaled so that  $\mathbf{y}^* E \mathbf{x} = 1$ , the eigenvalue derivative (3) becomes

$$\frac{\partial \lambda}{\partial p} = \mathbf{y}^* \frac{\partial A}{\partial p} \mathbf{x}. \quad (4)$$

The larger the magnitude of the derivative (4), the more sensitive eigenvalue  $\lambda$  is to changes in parameter  $p$ . It is therefore instructive to view the problem of computing the most sensitive poles in a similar way as the problem of computing the most dominant poles ( $\lambda_i$  with large  $\|R_i\|_2$ ), which is solved by the Dominant Pole Algorithm [20, 29, 28]. In the following section the convergence to the most sensitive poles by the Sensitive Pole Algorithm is explained by its striking similarities (and in some cases even equivalence) to the Dominant Pole Algorithm.

### 3 The Sensitive Pole Algorithm

The Sensitive Pole Algorithm (SPA) is explained by considering the single parameter case first (Section 3.1). In Section 3.2, SPA is generalized to the multiple parameter case. To improve global convergence, SPA is extended with subspace acceleration in Section 3.3. The case  $\frac{\partial E}{\partial p} \neq 0$  is discussed in Section 3.4.

It is stressed that SPA is not an algorithm to compute just the sensitivities of eigenvalues, since these can readily be computed with the eigenvalues and corresponding left and right eigenvectors found by several methods, provided the matrix derivatives are available, see also, e.g., [6]. Eigenvalue sensitivity applied to power system dynamics and control is a topic on its own right, used in the design of many power system controllers [33, 18, 31, 24, 22, 34]. The sensitivities that have been used by engineers include: all controller parameters in power generators, HVDC links and FACTS devices, that impact relevant system dynamics [31]. The Sensitive Pole Algorithm is really intended for the computation of only those eigenvalues of large scale systems that are most sensitive to parameter changes. Tracing root-locus plots for parameter changes in controllers of critical equipment [27, 26, 14, 17], aiding the gain coordination of power oscillation damping controllers [26, 5], and determining parametric stability margins [35, 8], are some of the many potential applications of this method.

In the following  $A(p)$  is denoted by just  $A$ , for the sake of brevity and clarity, and a similar convention is followed for the eigenvalue  $\lambda \equiv \lambda(p)$  and the left and right eigenvectors  $\mathbf{y} \equiv \mathbf{y}(p)$  and  $\mathbf{x} \equiv \mathbf{x}(p)$ , respectively. For ease of notation, it is assumed in Sections 3.1–3.4 that  $A$  and  $E$  depend linearly on the parameter(s). Section 3.5 comments on nonlinear parameter dependency.

#### 3.1 A single parameter

Suppose that the derivative of  $A$  to parameter  $p$  has rank 1 and hence can be written as

$$\frac{\partial A}{\partial p} = \mathbf{b} \mathbf{c}^*, \quad (5)$$

where  $\mathbf{b}, \mathbf{c} \in \mathbb{R}^n$  are vectors (not related to  $B$  and  $C$  in (1)). Then the sensitivity of an eigenvalue  $\lambda$  with left and right eigenvectors  $\mathbf{y}$  and  $\mathbf{x}$  (with  $\mathbf{y}^* E \mathbf{x} = 1$ ) becomes

$$\frac{\partial \lambda}{\partial p} = \mathbf{y}^* \frac{\partial A}{\partial p} \mathbf{x} = (\mathbf{y}^* \mathbf{b})(\mathbf{c}^* \mathbf{x}) = (\mathbf{c}^* \mathbf{x})(\mathbf{y}^* \mathbf{b}). \quad (6)$$

In the right-hand side of (6) one recognizes the residues of the transfer function  $H(s) = \mathbf{c}^*(sE - A)^{-1}\mathbf{b}$ . Consequently, the most sensitive eigenvalues of the pencil  $(A(p), E)$  can be computed by applying the Dominant Pole Algorithm [20] to  $(E, A, \mathbf{b}, \mathbf{c})$ , with  $\mathbf{b}$  and  $\mathbf{c}$  defined by (5).

If  $\text{rank}(\frac{\partial A}{\partial p}) = k > 1$ , the derivative of  $A$  to  $p$  can be written as

$$\frac{\partial A}{\partial p} = \sum_{i=1}^k \mathbf{b}_i \mathbf{c}_i^*, \quad (7)$$

where  $\mathbf{b}_i, \mathbf{c}_i \in \mathbb{R}^n$  ( $i = 1, \dots, k$ ). The sensitivity of an eigenvalue  $\lambda$  with left and right eigenvectors  $\mathbf{y}$  and  $\mathbf{x}$  (with  $\mathbf{y}^* E \mathbf{x} = 1$ ) becomes

$$\frac{\partial \lambda}{\partial p} = \mathbf{y}^* \frac{\partial A}{\partial p} \mathbf{x} = \sum_{i=1}^k (\mathbf{y}^* \mathbf{b}_i) (\mathbf{c}_i^* \mathbf{x}).$$

Similar to the rank 1 case, the algorithm for computing the most sensitive poles follows from relating the sensitivities to the residues of a transfer function. To be more precise, consider the MIMO system  $(E, A, B, C)$ , where

$$\begin{aligned} B &= [\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_k], \\ C &= [\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_k], \end{aligned}$$

have as their columns the  $\mathbf{b}_i$  and  $\mathbf{c}_i$  ( $i = 1, 2, \dots, k$ ), respectively. The residues of the corresponding transfer function  $H(s) = C^*(sE - A)^{-1}B$  are defined by

$$R_i = (C^* \mathbf{x}_i) (\mathbf{y}_i^* B),$$

where the left and right eigenvectors  $\mathbf{y}_i$  and  $\mathbf{x}_i$  are scaled so that  $\mathbf{y}_i^* E \mathbf{x}_i = 1$ . It follows that the sensitivity of eigenvalue  $\lambda$  with eigenvectors  $\mathbf{y}$  and  $\mathbf{x}$  is given by

$$\frac{\partial \lambda}{\partial p} = \mathbf{y}^* \frac{\partial A}{\partial p} \mathbf{x} = \sum_{i=1}^k (\mathbf{y}^* \mathbf{b}_i) (\mathbf{c}_i^* \mathbf{x}) = \text{trace}(R), \quad (8)$$

where  $R = (C^* \mathbf{x}) (\mathbf{y}^* B)$  has rank  $k$ , and the trace of an  $n \times n$  matrix  $A$  is defined by  $\text{trace}(A) = \sum_{i=1}^n a_{ii}$ . With the index for dominance defined by  $|\text{trace}(R_i)|$ , the most sensitive eigenvalues of  $(A, E)$  could in principle be computed by SAMDP [28] (with adapted selection criterion) as the most dominant poles (according to (8)) of  $(E, A, B, C)$ . However, the Sensitive Pole Algorithm, presented in Alg. 1, exploits the freedom in SAMDP for the choice of the right-hand sides in step 5 and 6 (see Section IV.B in [28]): in iteration  $k$ , use the new information (i.e., left and right eigenvector approximations  $\mathbf{w}_k$  and  $\mathbf{v}_k$ ) to update the right-hand sides

$$\mathbf{b} = \frac{\partial A}{\partial p} \mathbf{v}_k = B(C^* \mathbf{v}_k) \text{ and } \mathbf{c} = \frac{\partial A}{\partial p}^* \mathbf{w}_k = C(B^* \mathbf{w}_k). \quad (9)$$

These updates are motivated as follows. To compute the most sensitive eigentriplet  $(\lambda, \mathbf{x}, \mathbf{y})$ , i.e., the eigenvalue with largest sensitivity  $\mathbf{y}^* \frac{\partial A}{\partial p} \mathbf{x}$ , one would ideally take  $\mathbf{b} = \frac{\partial A}{\partial p} \mathbf{x}$  and  $\mathbf{c} = \frac{\partial A}{\partial p}^* \mathbf{y}$ . In that case,  $\mathbf{b}$  and  $\mathbf{c}$  are fixed and SPA boils down to DPA, which converges to the dominant poles [30], i.e., the poles with largest residues  $R = (\mathbf{c}^* \mathbf{x}) (\mathbf{y}^* \mathbf{b}) = (\mathbf{y}^* \frac{\partial A}{\partial p} \mathbf{x})^2$  (see also [10]), which in turn are the most sensitive eigenvalues. However, the unknown eigenvectors  $\mathbf{y}$  and  $\mathbf{x}$  are exactly the ones looked after. Hence, in iteration  $k$ , the approximations  $\mathbf{w}_k$  and  $\mathbf{v}_k$  of left and right eigenvectors  $\mathbf{y}$  and  $\mathbf{x}$ , respectively, are the most obvious choices in (9), see also step 3 and 4 of Alg. 1. The initial vectors  $\mathbf{v}_0$  and  $\mathbf{w}_0$  should be chosen such that  $\mathcal{A}_p \mathbf{v}_0 \neq 0$  and  $\mathcal{A}_p^* \mathbf{w}_0 \neq 0$ , where  $\mathcal{A}_p = \frac{\partial A}{\partial p}$  (cf. step 1 of Alg. 1); if none of the row and column sums of  $\mathcal{A}_p$  are zero, a practical choice is  $\mathbf{v}_0 = \mathbf{w}_0 = [1, \dots, 1]^T \in \mathbb{R}^n$  (also random initial vectors will do in practice).

Note that the matrices  $B$  and  $C$  need not to be computed explicitly for SPA (but could be determined by computing the (thin) SVD [7] of  $\frac{\partial A}{\partial p}$ ). If  $\frac{\partial A}{\partial p}$  has rank one, SPA boils down to DPA, as described in the beginning of this section.

Matlab code for SPA is given in the Appendix, together with numerical results on a small example that illustrate the behavior of SPA.

---

**Algorithm 1** Sensitive Pole Algorithm (SPA)

---

**INPUT:** Pencil  $(A, E)$ ,  $\mathcal{A}_p \equiv \frac{\partial A}{\partial p}$ , initial pole estimate  $s_0$ , tolerance  $\epsilon \ll 1$

**OUTPUT:** Approximate sensitive pole  $\lambda$  and corresponding right and left eigenvectors  $\mathbf{x}$  and  $\mathbf{y}$

- 1: Set  $k = 0$ ,  $\mathbf{v}_0, \mathbf{w}_0 \in \mathbb{R}^n$  s.t.  $\mathcal{A}_p \mathbf{v}_0 \neq 0$  and  $\mathcal{A}_p^* \mathbf{w}_0 \neq 0$
- 2: **while** not converged **do**
- 3:    $\mathbf{b} = \mathcal{A}_p \mathbf{v}_k / \|\mathcal{A}_p \mathbf{v}_k\|_2$
- 4:    $\mathbf{c} = \mathcal{A}_p^* \mathbf{w}_k / \|\mathcal{A}_p^* \mathbf{w}_k\|_2$
- 5:   Solve  $\mathbf{v}_{k+1} \in \mathbb{C}^n$  from  $(s_k E - A) \mathbf{v}_{k+1} = \mathbf{b}$
- 6:   Solve  $\mathbf{w}_{k+1} \in \mathbb{C}^n$  from  $(s_k E - A)^* \mathbf{w}_{k+1} = \mathbf{c}$
- 7:   Compute the new pole estimate

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

- 8:   The pole  $\lambda = s_{k+1}$  with  $\mathbf{x} = \mathbf{v}_{k+1} / \|\mathbf{v}_{k+1}\|_2$  and  $\mathbf{y} = \mathbf{w}_{k+1} / \|\mathbf{w}_{k+1}\|_2$  has converged if

$$\|A\mathbf{x} - s_{k+1}E\mathbf{x}\|_2 < \epsilon$$

- 9:   Set  $k = k + 1$

10: **end while**

---

### 3.2 Multiple parameters

If  $A \equiv A(p_1, p_2, \dots, p_k)$  depends on multiple parameters  $p_1, \dots, p_k$ , the derivative (gradient) of  $\lambda$  is composed of the partial derivatives:

$$\nabla \lambda = (\mathbf{y}^* \frac{\partial A}{\partial p_1} \mathbf{x}, \dots, \mathbf{y}^* \frac{\partial A}{\partial p_k} \mathbf{x}).$$

To measure the sensitivity in a unit direction  $\mathbf{d} \in \mathbb{R}^k$ , that is, when changing parameter  $p_i$  by  $d_i$ , the directional derivative is used:

$$\text{sensitivity}(\lambda, \mathbf{d}) \equiv |\nabla \lambda \cdot \mathbf{d}|.$$

Since

$$\nabla \lambda \cdot \mathbf{d} = \sum_{i=1}^k d_i \mathbf{y}^* \frac{\partial A}{\partial p_i} \mathbf{x} = \mathbf{y}^* \mathcal{A}_p \mathbf{x},$$

where

$$\mathcal{A}_p = \sum_{i=1}^k d_i \frac{\partial A}{\partial p_i}, \quad (10)$$

SPA (Alg. 1) can be used to compute the eigenvalues of  $(A, E)$  most sensitive to changes in multiple parameters as well (with  $\mathcal{A}_p$  as in (10), cf. steps 3 and 4 in Alg. 1).

In practical applications the entries of  $\mathbf{d}$  are typically equal to the increments of the corresponding parameters (see Section 4.2 for an example from practice).

### 3.3 Subspace acceleration and deflation

To improve global convergence, SPA can be extended with subspace acceleration. The ingredients (search spaces, selection strategy, deflation) of the resulting Subspace Accelerated Sensitive Pole Algorithm (SASPA) are discussed briefly here. See [29, 28] for more details, including full code for Matlab, on subspace acceleration in the context of computing dominant poles.

The main idea behind subspace acceleration is that the approximate eigenvectors  $\mathbf{v}_{k+1}$  and  $\mathbf{w}_{k+1}$  (step 5 and 6 of Alg. 1) are kept in search spaces  $V$  and  $W$ , respectively. In the  $k$ -th iteration,

this leads to a projected eigenproblem ( $W^*AV, W^*EV$ ) of order  $k \ll n$ . This small problem can be solved exactly using the QZ method [7], leading to  $k$  approximate eigentriplets of  $(A, E)$ . Of these triplets, the most sensitive is selected, and the corresponding eigenvalue approximation is used as shift for the next iteration.

If an approximate eigentriplet  $(\lambda, \mathbf{x}, \mathbf{y})$  satisfies the convergence criterion  $\|A\mathbf{x} - \lambda E\mathbf{x}\|_2 < \epsilon$  (with  $\epsilon \ll 1$ ), deflation is used to avoid convergence to this eigentriplet again: the search spaces  $V$  and  $W$  are kept orthogonal to  $E^*\mathbf{y}$  and  $E\mathbf{x}$ , respectively, during the computation of other sensitive eigenvalues. In practical root-locus studies involving particular generator controllers, the number of sensitive eigenvalues of interest is often very low ( $< 5$ ), and their approximate location is known. However, in coordinated tuning of power oscillation damping controllers [26, 5, 35] in large power systems, the number of poles of interest may easily jump to 50 or more. If this information is not known in advance, a dynamic criterion can be to continue computation of sensitive poles until the relative difference in sensitivity between new poles and already found poles is above some value.

### 3.4 The case $\frac{\partial E}{\partial p} \neq 0$

If the matrix  $E$  depends on a parameter  $p$  as well, the derivative of  $\lambda$  is given by

$$\frac{\partial \lambda}{\partial p} = \frac{\mathbf{y}^* \left( \frac{\partial A}{\partial p} - \lambda \frac{\partial E}{\partial p} \right) \mathbf{x}}{\mathbf{y}^* E \mathbf{x}}. \quad (11)$$

The descriptor formulation in [31] utilizes sensitivity formula (11) in the design of HVDC damping controllers. Accordingly, the application of the SPA algorithm to such formulation would need to be based on (11) to effectively produce root locus plots. The presence of the (unknown)  $\lambda$  in the derivative (11) requires an update every iteration similar to the update for the case  $\text{rank}(\mathcal{A}_p) > 1$  (described in Section 3.1). Besides the eigenvector approximations  $\mathbf{w}_k$  and  $\mathbf{v}_k$  (cf. step 3 and 4 of Alg. 1), now also the eigenvalue estimate  $s_k$  is used to update  $\mathcal{A}_p$  every iteration (between step 2 and 3 in Alg. 1):

$$\mathcal{A}_p = \frac{\partial A}{\partial p} - s_k \frac{\partial E}{\partial p}.$$

In the multiple parameter case the update becomes

$$\mathcal{A}_p = \sum_{i=1}^k d_i \left( \frac{\partial A}{\partial p_i} - s_k \frac{\partial E}{\partial p_i} \right).$$

### 3.5 Nonlinear parameter dependency

If  $A(p)$  (and/or  $E(p)$ ) depends nonlinearly on parameter  $p$ , (SA)SPA can still be applied. Since the parameter  $p$  is present in the derivative of  $A(p)$  (and of  $E(p)$ ), the derivative needs to be evaluated at the present parameter value  $p = p_0$ :

$$\mathcal{A}_p = \left( \frac{\partial A}{\partial p} - s_k \frac{\partial E}{\partial p} \right) \Big|_{p=p_0}.$$

This generalizes readily to the multiple parameter case. Note that for the case  $\frac{\partial E}{\partial p} = 0$ , the derivative  $\mathcal{A}_p$  needs to be computed only once per parameter value; in the case  $\frac{\partial E}{\partial p} \neq 0$ , the derivative  $\mathcal{A}_p$  needs to be updated every iteration between step 2 and 3 of Alg. 1.

In power system dynamics, a general first-order block is of the form  $(K + sL)/(M + sN)$ , where  $L$  and  $N$  are time constants. A nonlinear term in  $N$  occurs in one or more coefficients of the state (descriptor) matrix as well as in its derivative with respect to  $N$ . To analyze the robustness of generator controllers, such as automatic excitation systems, speed-governors and power system stabilizers (PSSs), to changes in many different parameters, the formulations described in this section and Sections 3.2 and 3.4 may need to be combined.

Table 1: Sensitive poles for  $p = a_{116,115} = 1$  (see Fig. 3).

pole	$\mathbf{y}^* \frac{\partial A}{\partial p} \mathbf{x}$	symbol	hits SPA	hits RQI
$-22.1 + 4.64i$	$-1.24 + 2.00i$	x	48%	35%
$-5.37 + 5.64i$	$+0.10 + 1.23i$	circle	23%	4%
$+0.53 + 5.41i$	$-0.13 + 0.02i$	square	7%	3%
$-1.28 + 4.37i$	$-0.04 - 0.11i$	*	1%	3%

Many of the linear state matrix coefficients are nonlinear functions of the system power flow parameters. Determining the state matrix derivative with respect to changes in the system operating point is a topic of current interest but never a trivial task [33, 31, 22, 34, 35]. Using approximations of this derivative was proposed in [33, 34] and this might be suitable for use in online SPA implementations.

## 4 Numerical results

This section describes numerical results related to the small-signal stability analysis of power systems. First, the ability of (SA)SPA to find the most sensitive poles is illustrated by results obtained for a 41-state model of a 5-machine power system. Second, SASPA is used to compute root-locus plots for large-scale practical power system models. It is shown that SASPA is more efficient and more robust compared to existing methods.

All experiments were carried out in Matlab 7 on a MacBook Pro (2GHz Intel Core Duo, 1 GB RAM). Unless stated otherwise, all computations were done with sparse descriptor realizations. The dashed line in some figures denotes the 5% damping ratio border. In all experiments, initial vectors were chosen as  $\mathbf{v}_0 = \mathbf{w}_0 = [1, \dots, 1]^T \in \mathbb{R}^n$  (cf. step 1 of Alg. 1).

### 4.1 41-State power system

The descriptor realization of this 41-state system has dimension  $n = 136$ . The small sizes of the state space and descriptor realizations allows for the intensive dense computations that are needed to confirm the results obtained by SASPA, and for extensive comparison with other methods.

The power system stabilizer gain (Kpss) for one generator in this 5-machine power system can be controlled by element  $a_{116,115}$  of the descriptor matrix<sup>1</sup>. Figure 1 shows the spectrum of  $(A, E)$  with  $\text{Kpss} = a_{116,115}^1 = 1$  (squares, only showing finite eigenvalues), together with the spectra for  $(A^j, E)$  (asterisks), where

$$a_{116,115}^j = j, \quad (j = 2, \dots, 50).$$

This gives a clear indication of the most sensitive poles with respect to changes of  $a_{116,115}$ . The four most sensitive poles and their derivatives are also listed in Table 1, together with symbols used in the Figures 3–4. Figure 1 was obtained by calling the QZ method ( $\mathbf{eig}(A^j, E)$  ( $j = 1, \dots, 50$ )) in Matlab (within 4 s CPU time).

The derivative of  $A$  to Kpss is given by  $\mathcal{A}_p = \mathbf{e}_{116} \mathbf{e}_{115}^T$ , where  $\mathbf{e}_i$  is the  $i$ -th elementary vector. This means that computing the sensitive poles with (SA)SPA is equivalent to computing the sensitive poles with (SA)DPA [29] as the dominant poles of  $(E, A, \mathbf{b}, \mathbf{c})$ , where  $\mathbf{b} = \mathbf{e}_{116}$  and  $\mathbf{c} = \mathbf{e}_{115}$ . Figure 2 shows for  $(A^j, E)$  ( $j = 1, \dots, 50$ ) the six sensitive poles computed by SASPA, using initial shift  $s_0 = 1i$  for every run. Despite the naive choice for  $s_0$ , SASPA succeeds in tracing the most sensitive poles (cf. Fig. 1), showing the robustness of SASPA. As was confirmed by numerical experiments (not reported here), the very sporadic convergence to a less sensitive pole can be fixed by using as initial shifts for SASPA( $A^j, E$ ) the poles found by the previous run SASPA( $A^{j-1}, E$ ). SASPA required 40 s CPU time, against 4 s for  $\mathbf{eig}$ . However, due to the small system size, the costs for SASPA are dominated by overhead of keeping search spaces. For larger

<sup>1</sup>Row 116 contains the algebraic equation  $-\text{out}(t) + \text{Kpss} * \text{in}(t) = 0$ , with variables 'in' (for input to a gain block) and 'out' (for output of the gain block) ordered as 115 and 116, respectively, in the augmented state vector.

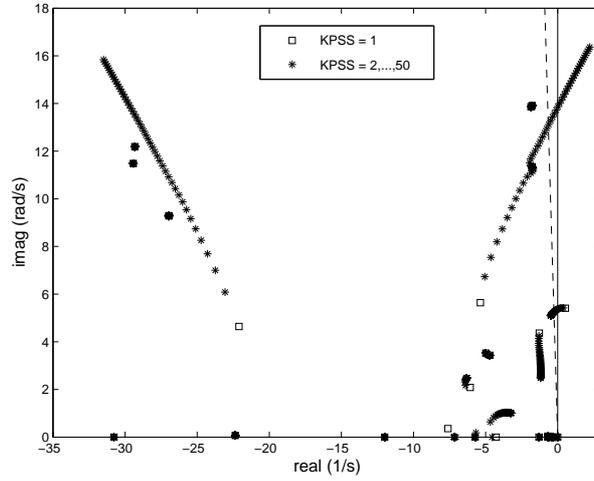


Figure 1: Full spectra of the 41-state power system matrices  $(A^j, E)$ , where  $a_{116,115}^j = j$  ( $j = 1, \dots, 50$ ), computed by the QZ method.

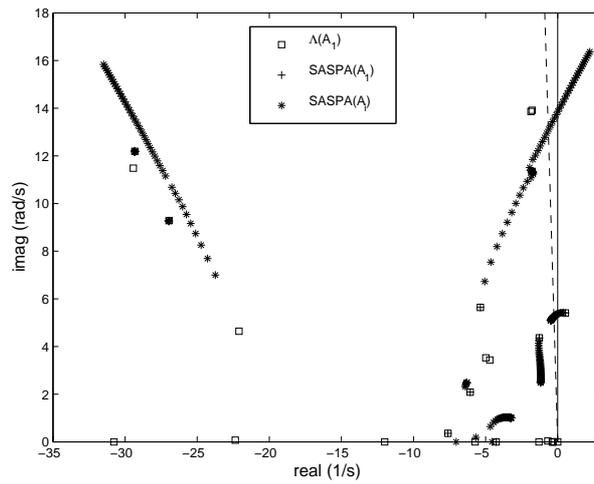


Figure 2: Sensitive pole traces for  $(A^j, E)$ , where  $a_{116,115}^j = j$  ( $j = 1, \dots, 50$ ), computed by SASPA with  $s_0 = 1i$  (six poles every run).

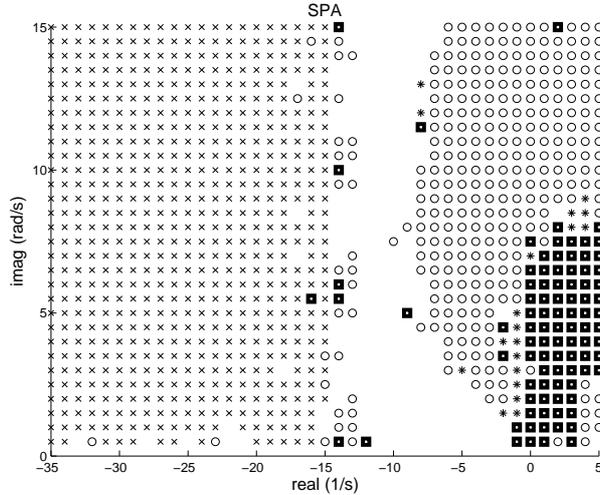


Figure 3: Sensitive pole convergence areas for SPA applied to the 41-state system with PSS gain  $a_{116,115} = 1$ , see also Table 1.

systems SASPA is faster than `eig` (QR/QZ method), as is confirmed by results in Section 4.2 (moreover, SASPA is applicable to large-scale systems, while the QR/QZ method is limited to systems of dimension  $O(10^3)$ ).

In Fig. 3 convergence areas for the most sensitive poles of  $(A^1, E)$  as obtained for SPA are displayed (note that the same results would be obtained for DPA applied to  $H(s) = \mathbf{c}^*(sE - A)^{-1}\mathbf{b}$ , with  $\mathbf{b} = \mathbf{e}_{116}$  and  $\mathbf{c} = \mathbf{e}_{115}$ , see the last part of Section 3.1). Fig. 4 shows the convergence areas obtained for two-sided RQI [25]. These figures should be interpreted as follows: a symbol at point  $(x, y)$  in the complex plane means that the respective method (SPA/RQI) started with  $s_0 = x + iy$  and converged to the pole corresponding to the symbol in Table 1 (for instance, an x means convergence to  $-22.1 \pm 4.64i$  in Table 1). Grid points without symbol denote convergence of the respective method to a less sensitive pole. Hence, the more x-es and circles, the better the performance of the method. Clearly, for SPA the convergence areas of the most sensitive poles are much larger, confirming its effectiveness and robustness (cf. the corresponding percentages in Table 1). Since two-sided RQI has an asymptotically cubic rate of convergence, against quadratic for SPA, the average number of iterations needed for convergence ( $\epsilon = 10^{-12}$ ) is lower for RQI (6.8 iterations against 9.8 for SPA), but SPA converges to poles that are more sensitive.

To summarize, these results show that SASPA is able to compute and trace the most sensitive poles in an effective and robust way, even without using a clever tracing strategy.

## 4.2 Large-scale power system I (1664 states)

The large system is a 1997 planning model for the Brazilian Interconnected Power System (BIPS/97) [17, 4, 29, 28]. The state-space realization has 1664 states, and the descriptor realization is of order  $n = 13250$ .

In the first experiment the PSS gain ( $K_{pss}$ ) of the Xingó generator ( $a_{4971,4970}$ ) is varied between 0 and 30, with increments of 0.5. Figure 5 shows the traces for the most sensitive poles as computed by SASPA (computing 10 poles per run, initial shift  $s_0 = 3i$ ). The CPU time for the 60 runs was 1450 s. A root-locus plot for all poles (Fig. 6) was produced by using the QR method for the state-space matrix, needing 7800 s (only computing eigenvalues). The sensitive poles are those that significantly shift with PSS gain, yielding the same critical root-locus branches that were tracked by the SASPA method, cf. Fig. 5. Clearly, SASPA is much faster, while producing qualitatively the same results. Moreover, SASPA also provides the eigenvalue derivatives, while the QR method

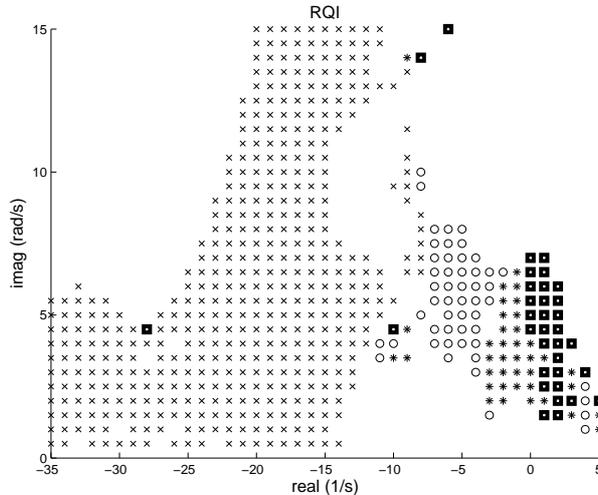


Figure 4: Sensitive pole convergence areas for two-sided RQI applied to the 41-state system with PSS gain  $a_{116,115} = 1$ , see also Table 1.

Table 2: Generators, PSS gain ranges, and increments for BIPS (cf. Figure 7 and 8).

generator	element	gain (min, max)	increment
Xingó	$a_{4971,4970}$	(0, 15)	0.5
Paulo Afonso IV	$a_{4901,4900}$	(0, 15)	0.5
Itaipu Kp	$a_{4973,4972}$	(0, 2.2)	0.0733
Itaipu Kw	$a_{4978,4977}$	(0, 10.35)	0.345

needs 33600 s to compute the left and right eigenvectors as well.

Sparse eigenmethods may be used in a continuation scheme [26, 17, 5] to compute root-locus plots, but generally require expert knowledge of the used methods and system. SASPA, on the other hand, automatically finds the most sensitive poles. Although SASPA can take (computational) advantage of eigentriplets computed for previous parameter values, this is not necessary. This makes SASPA also attractive for parallel computation, since SASPA runs for several parameter values can be executed in parallel (contrary to continuation type methods, where results for previous parameter values are needed).

The second example is a multiparameter root-locus, where the PSS gains for Xingó, Paulo Afonso IV, and Itaipu generators are varied simultaneously. The Itaipu PSS has two channels (power and frequency deviations), and the two associated gains, Kp and Kw, are varied simultaneously (see Table 2). Figure 7 shows the traces for the most sensitive critical poles as computed by SASPA (computing 20 poles per run, initial shift  $s_0 = 1i$ ). The CPU time for the 30 runs was 3000 s (only 1400 s when computing 10 poles per run; 20 poles were computed to trace the non-critical poles as well). SASPA succeeds in tracing both critical as non-critical sensitive poles, and is still faster than the QR method (3600 s), cf. Fig. 8.

### 4.3 Large-scale power system II (3172 states)

BIPS/07 is a year 2007 operations planning model developed by the Brazilian system operator ONS ([www.ons.com.br](http://www.ons.com.br)). The state-space realization has 3172 states, and the descriptor realization is of order  $n = 21476$ . Figure 9 shows the traces for the poles most sensitive to simultaneous variations in the Itaipu Kp and Kw gains, as computed by SASPA (computing 20 poles per run, initial shift  $s_0 = 2i$ ). The CPU time for the 30 SASPA runs was 4664 s. A single QR run to compute the full set of eigenvalues requires already 800 s, and computation of a 30 step root-locus plot with

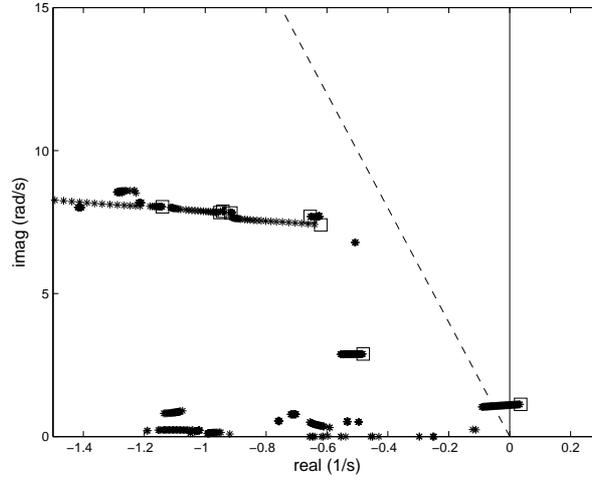


Figure 5: Root locus plot of sensitive poles of BIPS/97 computed by SASPA (Xingó PSS gain  $a_{4971,4970} = 0, 0.5, \dots, 30$ ). As the gain increases, the critical rightmost pole (starting in the right half of the complex plane) crosses the imaginary axis and the 5% damping ratio boundary. Squares denote the poles for  $a_{4971,4970} = 0$ .

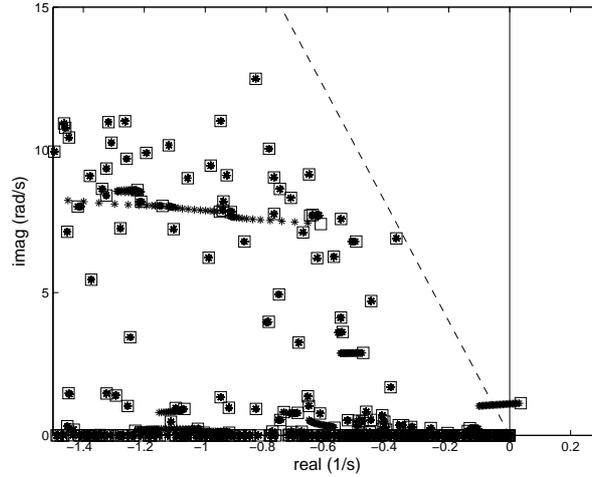


Figure 6: Root locus plot of poles of BIPS/97 computed by the QR method (Matlab's `eig`) (Xingó PSS gain  $a_{4971,4970} = 0, 1, \dots, 30$ ). As the gain increases, the critical rightmost pole (starting in the right half of the complex plane) crosses the imaginary axis and the 5% damping ratio boundary. Squares denote the poles for  $a_{4971,4970} = 0$ .

Table 3: CPU times (seconds) for computation of root-locus plots with SASPA and the QR method for BIPS/97 and BIPS/07. Notation: S=single parameter, M=multiple parameters,  $n$ =number of states,  $N$ =dimension of descriptor realization, steps=number of steps in the root-locus.

BIPS	S/M	$n$	$N$	steps	SASPA(#poles)	QR
97	S	1664	13250	60	1450 (10)	7800
97	M	1664	13250	30	1400 (10)	3600
97	M	1664	13250	30	3000 (20)	3600
07	M	3172	21476	30	4664 (20)	24000

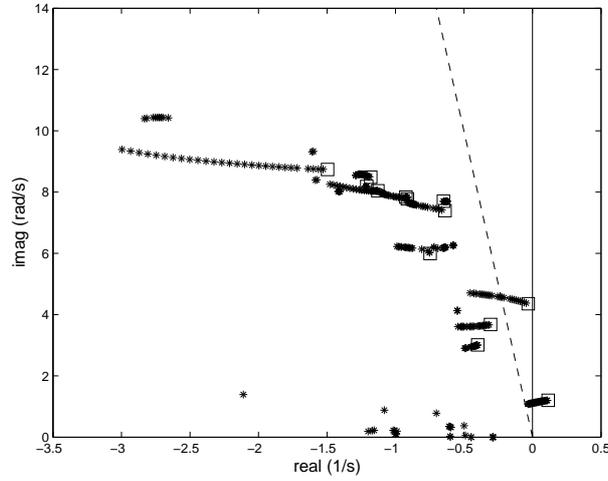


Figure 7: Root locus plot of sensitive poles of BIPS/97 computed by SASPA (multiple parameters, see Table 2 for gain ranges). Various poles become more damped as the PSS gains increase, the two critical rightmost poles crossing the imaginary axis and the 5% damping ratio boundary, respectively. Squares denote the poles for the system when the three PSSs in Table 2 have zero gain.

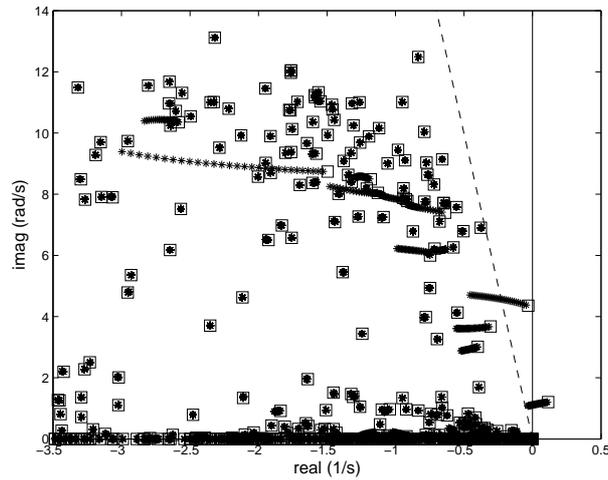


Figure 8: Root locus plot of poles of BIPS/97 computed by the QR method (multiple parameters, see Table 2 for gain ranges). Various poles become more damped as the PSS gains increase, the two critical rightmost poles crossing the imaginary axis and the 5% damping ratio boundary, respectively. Squares denote the poles for the system when the three PSSs in Table 2 have zero gain.

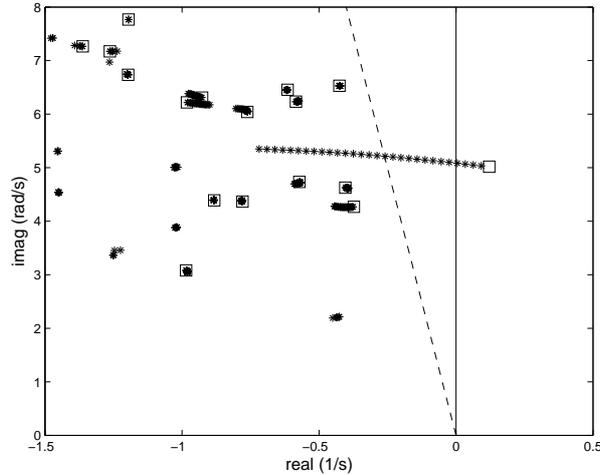


Figure 9: Root locus plot of sensitive poles of BIPS/07 (3172 states) computed by SASPA. As the Itaipu  $K_p$  and  $K_w$  gains increase (see Table 2 for gain ranges), the critical rightmost pole crosses the imaginary axis and the 5% damping ratio boundary, respectively. Squares denote the poles for the system when the PSS has zero gains in its two channels.

the QR method would take 24000 s (see also Table 3). Figure 9 indicates that the Itaipu PSS is essential to maintain positive damping to the major oscillatory mode associated with this large power plant (at slightly above 5 rad/s), but does not significantly impact other interarea modes. It also confirms that SASPA can be used to obtain root-locus plots for large power system models within reasonable CPU time, while application of the QR method is no longer practical.

## 5 Conclusions

The (subspace accelerated) Sensitive Pole Algorithm (SA)SPA is an effective, efficient, and robust method to compute the poles that are most sensitive to system parameter changes. Because SASPA operates on the sparse descriptor matrices of the system and computes the most sensitive poles automatically, it is very useful in root-locus studies of large-scale systems.

Root-locus plots, as their computation with SASPA is cost-effective for large-scale systems, will help to expedite small-signal stability studies on the verification of PSS and POD controller tuning and effectiveness to damp local and interarea modes during base case conditions and contingencies [26, 32, 33], among other applications.

Numerical experiments with realistic power system models confirmed that SASPA is a reliable method for computing and tracing the sensitive poles and showed significant speed-ups in computational times over existing methods such as the QR method. The authors envision applying (SA)SPA to help to determine generation rescheduling in critical power plants for online oscillation damping control [33, 34]. The efficient determination of the system matrix derivatives for numerous rescheduling alternatives is needed for the successful application of SA(SPA) in this online application.

Without any adaptation, SASPA can be used for sensitive eigenvalue computations that arise in several other engineering fields and numerical analysis as well.

Table 4: Iterates for SPA and RQI for  $A = \text{diag}(3\alpha, \alpha)$ , with  $\alpha = 1$ .

$k$	SPA		RQI	
	$s_k$	$\ A\mathbf{v}_k - s_k\mathbf{v}_k\ _2$	$s_k$	$\ A\mathbf{v}_k - s_k\mathbf{v}_k\ _2$
0	1.5	$O(10^0)$	1.5	$O(10^0)$
1	2	$O(10^0)$	1.2	$O(10^{-1})$
2	2.8	$O(10^{-1})$	1.0027	$O(10^{-2})$
3	2.999	$O(10^{-2})$	1	$O(10^{-4})$
4	3	$O(10^{-6})$	1	$O(10^{-13})$
5	3	$O(10^{-18})$	1	$O(10^{-18})$

## Acknowledgments

The authors would like to thank the anonymous referees for useful suggestions that helped to improve the readability of the paper.

## A SPA: Matlab code and example

The Matlab code for the Sensitive Pole Algorithm (cf. Alg. 1) is shown in Alg. 2. The following

---

### Algorithm 2 Matlab code for Sensitive Pole Algorithm (SPA)

---

```
function [lambda, x, y] = spa(A, E, dA, s0, tol)
n = size(A, 1) ; nres = 1 ; k = 0 ; s = s0 ;
v = ones(n,1) ; v = v / norm(v) ;
w = ones(n,1) ; w = w / norm(w) ;
while nres > tol
    b = dA*v ; b = b / norm(b) ;
    c = dA'*w ; c = c / norm(c) ;
    v = (s*E - A) \ b ;
    w = (s*E - A)' \ c ;
    s = s - c'*v / (w'*E*v) ;
    v = v / norm(v) ; w = w / norm(w) ;
    nres = norm( A*v - s*E*v ) ;
    k = k + 1 ;
end
lambda = s ; x = v ; y = w ;
```

---

small example illustrates the convergence behavior of SPA. Let  $A = \text{diag}(3\alpha, \alpha)$  and  $E = I$ , where  $\alpha$  is a parameter. Clearly,  $\lambda_1 = 3\alpha$  is the most sensitive eigenvalue:  $\frac{\partial \lambda_1}{\partial \alpha} = 3$ , while  $\frac{\partial \lambda_2}{\partial \alpha} = 1$ . Table 4 shows the iterates for SPA and RQI for  $\alpha = 1$ ,  $s_0 = 1.5$ , and  $\mathbf{v}_0 = \mathbf{w}_0 = [1, 1]^T$ . Having the same initial settings, SPA converges to the most sensitive eigenvalue  $\lambda_1 = 3$ , while RQI converges to  $\lambda_2 = 1$ , the eigenvalue nearest to the initial shift  $s_0 = 1.5$ .

## References

- [1] BOLLINGER, K., HURLEY, J., LARSEN, E., AND LEE, D. Power system stabilization via excitation control. No. 81EHO175-0-PWR. IEEE, 1981.
- [2] CHOW, J. H., SANCHEZ-GASCA, J. J., REN, H., AND WANG, S. Power system damping controller design using multiple input signals. *IEEE Control Syst. Mag.* 20, 4 (Aug 2000), 82–90.

- [3] CIGRE TF 38.01.17. Analysis and control of power system oscillations. Tech. Rep. 111, CIGRE, December 1996.
- [4] CIGRE TF 38.02.16. Impact of the interactions among power system controls. Tech. Rep. 166, CIGRE, July 2000.
- [5] FERRAZ, J. C. R., MARTINS, N., AND TARANTO, G. N. Coordinated stabilizer tuning in large power systems considering multiple operating conditions. In *IEEE/PES General Meeting* (June 2007), no. 07GM0304.
- [6] FREUND, R. W., AND FELDMANN, P. Small-signal circuit analysis and sensitivity computations with the PVL algorithm. *IEEE Trans. Circuits Syst. – II: Analog and Digital Signal Processing* 43, 8 (August 1996), 577–585.
- [7] GOLUB, G. H., AND VAN LOAN, C. F. *Matrix Computations*, third ed. John Hopkins University Press, 1996.
- [8] GOMES JR, S., MARTINS, N., AND PORTELA, C. Computing small-signal stability boundaries for large-scale power systems. *IEEE Trans. Power Syst.* 18, 2 (May 2003), 747–752.
- [9] HALEY, S. B. The generalized eigenproblem: pole-zero computation. *Proc. IEEE* 76, 2 (February 1988), 103–120.
- [10] HAMDAN, A. M. A., AND NAYFEH, A. H. Measures of modal controllability and observability for first- and second-order linear systems. *J. Guid. Contr. Dyn.* 12, 3 (May 1989), 421–428.
- [11] KAILATH, T. *Linear Systems*. Prentice-Hall, 1980.
- [12] KLEIN, M., ROGERS, G. J., MOORTY, S., AND KUNDUR, P. Analytical investigation of factors influencing power system stabilizers performance. *IEEE Trans. E.C. EC-7* (September 1992), 382–388.
- [13] KUO, B. C. *Automatic Control Systems*, 7 ed. Prentice-Hall, 1995.
- [14] LARSEN, E. V., AND BAKER, D. H. Experience with small-signal stability analysis of power systems. In *Eigenanalysis and frequency domain methods for system dynamic performance*, no. 90TH0292-3-PWR. IEEE, 1990, pp. 67–76.
- [15] LARSEN, E. V., AND SWANN, D. A. Applying power system stabilizers, Part I: General concepts, Part II: Performance objectives and tuning concepts, Part III: Practical considerations. *IEEE Trans. PAS 100* (1981), 3017–3046.
- [16] LIMA, L. T. G. Challenges in small-signal analysis of large interconnected power systems. In *IEEE/PES General Meeting* (June 2007), no. 07GM01371.
- [17] MARTINS, N., DE ANDRADE BARBOSA, A., FERRAZ, J. C. R., DOS SANTOS, M. G., BERGAMO, A. L. B., YUNG, C. S., DE OLIVEIRA, V. R., AND DE MACEDO, N. J. P. Retuning stabilizers for the North-South Brazilian interconnection. In *Proceedings of Power Engineering Society Summer Meeting* (July 1999).
- [18] MARTINS, N., AND LIMA, L. T. G. Determination of suitable locations for power system stabilizers and static var compensators for damping electromechanical oscillations in large scale power systems. *IEEE Trans. Power Syst.* 5, 4 (Nov 1990), 1455–1469.
- [19] MARTINS, N., AND LIMA, L. T. G. Eigenvalue and frequency domain analysis of small-signal stability problems. In *Eigenanalysis and frequency domain methods for system dynamic performance*, no. 90TH0292-3-PWR. IEEE, 1990, pp. 17–33.
- [20] MARTINS, N., LIMA, L. T. G., AND PINTO, H. J. C. P. Computing dominant poles of power system transfer functions. *IEEE Trans. Power Syst.* 11, 1 (Feb 1996), 162–170.

- [21] MURTHY, D. V., AND HAFTKA, R. T. Derivatives of eigenvalues and eigenvectors of a general complex matrix. *Int. J. Num. Meth. Eng.* 26 (January 1988), 293–311.
- [22] NAM, H.-K., KIM, Y.-K., SHIM, K.-S., AND LEE, K. Y. A new eigen-sensitivity theory of augmented matrix and its applications to power system stability analysis. *IEEE Trans. Power Syst.* 15, 1 (February 2000), 363–369.
- [23] OGATA, K. *Modern Control Engineering*, 2 ed. Prentice-Hall, 1990.
- [24] PAGOLA, F. L., PÉREZ-ARRIAGA, I. J., AND VERGHESE, G. C. On sensitivities, residues and participations: applications to oscillatory stability analysis and control. *IEEE Trans. Power Syst.* 4, 1 (February 1989), 278–285.
- [25] PARLETT, B. N. The Rayleigh quotient iteration and some generalizations for nonnormal matrices. *Math. Comp.* 28, 127 (July 1974), 679–693.
- [26] POURBEIK, P., AND GIBBARD, M. Simultaneous coordination of power system stabilizers and FACTS device stabilizers in a multimachine power system for enhancing dynamic performance. *IEEE Trans. Power Syst.* 13 (May 1998), 473–478.
- [27] ROGERS, G. *Power System Oscillations*. Kluwer Academic Publishers, 2000.
- [28] ROMMES, J., AND MARTINS, N. Efficient computation of multivariable transfer function dominant poles using subspace acceleration. *IEEE Trans. Power Syst.* 21, 4 (Nov 2006), 1471–1483.
- [29] ROMMES, J., AND MARTINS, N. Efficient computation of transfer function dominant poles using subspace acceleration. *IEEE Trans. Power Syst.* 21, 3 (Aug. 2006), 1218–1226.
- [30] ROMMES, J., AND SLEIJPEN, G. L. G. Convergence of the dominant pole algorithm and Rayleigh quotient iteration. Preprint 1356, Utrecht University, 2006. Submitted.
- [31] SMED, T. Feasible eigenvalue sensitivity for large power systems. *IEEE Trans. Power Syst.* 8, 2 (May 1993), 555–563.
- [32] UHLEN, K., ELENUS, S., NORHEIM, I., JYRINSALO, J., ELOVAARA, J., AND LAKERVI, E. Application of linear analysis for stability improvements in the Nordic power transmission system. In *IEEE/PES General Meeting* (July 2003).
- [33] WANG, L., HOWELL, F., KUNDUR, P., CHUNG, C. Y., AND XU, W. A tool for small-signal security assessment of power systems. In *Proc. Int. Conf. Power Ind. Comput. Appl.* (May 21–24 2001), pp. 246–252.
- [34] WANG, L., HOWELL, F., KUNDUR, P., CHUNG, C. Y., AND XU, W. Generation rescheduling methods to improve power transfer capability constrained by small-signal stability. *IEEE Trans. Power Syst.* 19, 1 (February 2004), 524–530.
- [35] YANG, D., AND AJJARAPU, V. Critical eigenvalues tracing for power system analysis via continuation of invariant subspaces and projected Arnoldi method. *IEEE Trans. Power Syst.* 22, 1 (February 2007), 324–332.