Eigen-Analysis of Large Delayed Cyber-Physical Power System by Time Integration-Based Solution Operator Discretization Methods

Hua Ye, *Member, IEEE*, Qianying Mou, *Student Member, IEEE*, Xinlei Wang, and Yutian Liu, *Senior Member, IEEE*

***Abstract*—In this paper, solution operator discretization methods with linear multistep and implicit Runge–Kutta (SOD-LMS/IRK) are presented for efficient eigen-analysis of large delayed cyber-physical power system (DCPPS). First, the time integration-based discretization generates highly structured approximate matrices to the solution operator. Exploitation of the structure ensures low computational burden and high efficiency in solving the matrix-inversion-vector product involved in eigenvalue computation. Second, the implicitly restarted Arnoldi algorithm is employed to compute critical eigenvalue from the solution operator's discretized matrices. SOD-LMS/IRK are endowed with scalability in analyzing very large DCPPS by fully utilizing the inherent sparsity in the augmented system state matrices. The main contribution of the presented SOD-LMS/IRK is the improved efficiency and scalability of existing SOD-PS (pseudo-spectral collocation) and SOD-LMS/IRK. Numerical results on the 16-generator 68-bus test system, a 516-bus, and a 33028-bus real-life large transmission systems validate the effectiveness of the proposed SOD-LMS/IRK.**

***Index Terms*—Cyber-physical power system, eigen-analysis, small signal stability, solution operator discretization, spectral discretization, time delay, wide-area measurement system.**

Nomenclature

DCPPS	Delayed cyber-physical power system.
SOD	Solution operator discretization.
IGD	Infinitesimal generator discretization.
EIGD/ IIGD	Explicit/iterative infinitesimal generator discretization.
MIVP	Matrix-inversion-vector product.
MVP	Matrix-vector product.
SOD-LMS/IRK/PS	Solution operator discretization with linear multistep/implicit Runge-Kutta/ pseudo-spectral collocation.
IRA	Implicitly restarted Arnoldi algorithm.
WADC	Wide-area damping controller.
WAMS	Wide-area measurement system.

Manuscript received June 19, 2017; revised November 15, 2017 and February 20, 2018; accepted April 11, 2018. Date of publication April 13, 2018; date of current version October 18, 2018. This work was supported in part by the Natural Science Foundation of China under Grants 51107073 and 51677107 and in part by the Young Scholar Program of Shandong University under Grant 2016WLJH06. Paper no. TPWRS-00919-2017. *(Corresponding author: Qianying Mou.)*

The authors are with the Key Laboratory of Power System Intelligent Dispatch and Control of Ministry of Education, Shandong University, Ji'nan 250061, China (e-mail: yehua@sdu.edu.cn; muqy@mail.sdu.edu.cn; wxinlei008@163.com; liuyt@sdu.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TPWRS.2018.2826576

I. Introduction

WAMS deployed at the transmission level provides wide-area situational awareness of the physical power system in real time [1], [2]. In sharp contrast with local signals, wide-area measurements, such as active power on the tie-lines and relative rotor angles/speeds, show good observability of some significant interarea oscillation modes. By using wide-area measurements as remote feedback control signals, WADCs have a significant potential in stabilizing power system against inter-area low frequency oscillations [3]. However, processing and transmitting wide-area measurements in WAMS introduce unneglectable communication delays. Typically, they are in the range of a few milliseconds to several hundred milliseconds [4], [5]. The time delay impacts on system's small signal stability [6]–[10] and WADC design [11], [12] are far from trivial. Challenges posed by time delay in WAMS reveal the need of efficient stability analysis algorithms and robust controller design methods for DCPPS.

In recent years, there are increased interests in spectral discretization-based methods for eigen-analysis of the physical power system with inclusion of time delays. The methods allow one to compute eigenvalues of DCPPS from the spectra of the infinitesimal generator and solution operator associated with DCPPS. For IGD-based methods presented in [13]–[15], only a set of eigenvalues surrounding a given shift point can be captured. Multiple runs of the methods are required to scan critical oscillation modes of DCPPS located in the vicinity of the imaginary axis. To avoid the eigenvalue scanning, SOD-PS was presented in [16] to compute a set of critical eigenvalues with damping ratios less than a given threshold by one run of the method. Nevertheless, the efficiency of SOD-PS is constrained by iteratively solving the high-dimensional MIVP in generating Krylov sequences. In [17], two time integration-based methods, i.e., SOD-LMS/IRK, were presented for eigen-analysis of DCPPS. A salient characteristic of the methods is the highly structured approximate matrices to the solution operator. However, it is questionable whether the methods still work when the size of the system scales up. This is because the QR algorithm

adopted in [17] calls for prohibitive memory. In addition, the inherent sparsity in the augmented system state matrices has not been exploited.

From the viewpoint of the authors, it is believed that the advantages of SOD-PS and SOD-LMS/IRK are complementary. Based on this finding, scalable and efficient SOD-LMS/IRK methods are presented by applying the computational framework presented in [16] to SOD-LMS/IRK in [17]. First, the theoretical foundation of SOD-LMS/IRK is established. The reason why SOD-LMS/IRK gains superior efficiency to SOD-PS is analyzed. Second, how to determine the discretization step-length and select LMS/IRK integration methods are intensively discussed. Third, SOD-LMS/IRK are efficiently implemented and endowed with scalability by exploiting the inherent sparsity in augmented system matrices. Last, comparisons among SOD-LMS, SOD-IRK and SOD-PS are conducted through numerical studies on the 16-generator 68-bus test system and two real-life large transmission systems.

The main contribution of this paper is the improved scalability and efficiency of previous SOD-based methods in [16] and [17]. The scalability of SOD-LMS/IRK is achieved by fully exploiting the inherent sparsity in augmented system state matrices. SOD-LMS is provided with the highest efficiency by directly solving the MIVP involved in sparse eigenvalue computation. SOD-IRK gains higher efficiency than SOD-PS by cutting down the burden of iteratively solving the involved MIVP. In addition, theoretical analyses are intensively conducted, including efficiency comparison among various SOD-based methods, determination of the step-length for time integration-based discretization, and appropriate use of different LMS/IRK methods. At last, emphasis should be placed on the unique feature of SOD-LMS/IRK. As has been proved in [16], SOD-based eigen-analysis methods can efficiently calculate the least damped oscillation modes of DCPPS. Therefore, they are capable of determining the small signal stability of DCPPS in a faster manner when compared with EIGD in [15].

The remainder of the paper is organized as follows. Section II formulates the eigenvalue problem of DCPPS. Section III establishes the theoretical foundation of time integration-based SOD methods. Section IV gives guidance on determining the discretization step-length and selecting LMS/IRK integration methods. SOD-LMS/IRK are efficiently implemented in Section V. The effectiveness of SOD-LMS/IRK is validated in Section VI, followed by Section VII which concludes the paper.

II. DCPPS Modeling and Its Eigen-Problem

The dynamics of DCPPS around an equilibrium $(\boldsymbol{x}(0), \boldsymbol{y}(0))$ can be described by the following set of delayed differential algebraic equations.

$$\begin{cases} \Delta\dot{\boldsymbol{x}}(t) = \boldsymbol{A}_0\Delta\boldsymbol{x}(t) + \boldsymbol{B}_0\Delta\boldsymbol{y}(t) \\ \qquad\qquad + \sum_{i=1}^{m}(\boldsymbol{A}_i\Delta\boldsymbol{x}(t-\tau_i) + \boldsymbol{B}_i\Delta\boldsymbol{y}(t-\tau_i)) \\ \boldsymbol{0} = \boldsymbol{C}_0\Delta\boldsymbol{x}(t) + \boldsymbol{D}_0\Delta\boldsymbol{y}(t) \\ \boldsymbol{0} = \boldsymbol{C}_i\Delta\boldsymbol{x}(t-\tau_i) + \boldsymbol{D}_i\Delta\boldsymbol{y}(t-\tau_i),\ i = 1,\ \ldots,\ m \end{cases} \tag{1}$$

where t is the time instant. $\boldsymbol{x}(t) \in \mathbb{R}^{n\times 1}$ is the state variable vector, consisting of state variables of dynamic components, e.g., generators and the associated excitation systems, HVDC systems and FACTS devices, as well as supplementary damping controllers on them, etc. $\boldsymbol{y}(t) \in \mathbb{R}^{l\times 1}$ is the algebraic variable vector, i.e., bus voltage. $\tau_i\,(i = 1, \ldots, m)$ are discrete time delays, $0 < \tau_1 < \cdots < \tau_m \triangleq \tau_{\max}$. $\boldsymbol{A}_i \in \mathbb{R}^{n\times n}$, $\boldsymbol{B}_i \in \mathbb{R}^{n\times l}$, $\boldsymbol{C}_i \in \mathbb{R}^{l\times n}$ and $\boldsymbol{D}_i \in \mathbb{R}^{l\times l}\,(i = 0, \ldots, m)$ are augmented system matrices, which are highly sparse. In addition, $\boldsymbol{C}_i = \boldsymbol{C}_0$ and $\boldsymbol{D}_i = \boldsymbol{D}_0$ for $i = 1, \ldots, m$.

Since matrix $\boldsymbol{D}_0$ is non-singular, (1) can be further reduced to the following delayed differential equations by eliminating $\Delta\boldsymbol{y}(t)$ and $\Delta\boldsymbol{y}(t-\tau_i)\,(i = 1, \ldots, m)$.

$$\begin{cases} \Delta\dot{\boldsymbol{x}}(t) = \tilde{\boldsymbol{A}}_0\Delta\boldsymbol{x}(t) + \sum_{i=1}^{m}\tilde{\boldsymbol{A}}_i\Delta\boldsymbol{x}(t-\tau_i),\ t \geq 0 \\ \Delta\boldsymbol{x}(t) = \Delta\boldsymbol{x}(0) \triangleq \boldsymbol{\varphi},\ t \in [-\tau_{\max},\ 0] \end{cases} \tag{2}$$

where $\boldsymbol{\varphi}$ is the initial system state. $\tilde{\boldsymbol{A}}_0 \in \mathbb{R}^{n\times n}$ and $\tilde{\boldsymbol{A}}_i \in \mathbb{R}^{n\times n}\,(i = 1, \ldots, m)$ are the dense matrix of system states and highly sparse matrices of delayed system states, respectively. They can be formulated by the augmented system matrices $\boldsymbol{A}_i$, $\boldsymbol{B}_i$, $\boldsymbol{C}_0$ and $\boldsymbol{D}_0\,(i = 0, \ldots, m)$.

$$\tilde{\boldsymbol{A}}_0 = \boldsymbol{A}_0 - \boldsymbol{B}_0\boldsymbol{D}_0^{-1}\boldsymbol{C}_0 \tag{3}$$

$$\tilde{\boldsymbol{A}}_i = \boldsymbol{A}_i - \boldsymbol{B}_i\boldsymbol{D}_0^{-1}\boldsymbol{C}_0. \tag{4}$$

The characteristic equation corresponding to (2) is transcendental and has an infinite number of eigenvalues.

$$\left(\tilde{\boldsymbol{A}}_0 + \sum_{i=1}^{m}\tilde{\boldsymbol{A}}_i\mathrm{e}^{-\lambda\tau_i}\right)\boldsymbol{v} = \lambda\boldsymbol{v} \tag{5}$$

where λ and $\boldsymbol{v}$ are the eigenvalue and the corresponding right eigenvector, respectively.

In this paper, two time integration-based SOD methods are presented to compute a reduced set of critical oscillation modes of large DCPPS from (5) so that the small signal stability of the system is efficiently and reliably determined.

III. Theoretical Foundation of Time Integration-Based SOD Methods

This section presents the theoretical foundation of time integration-based SOD methods, including definition and expression of the solution operator, Kronecker product reformulation of approximate matrices generated by SOD-LMS/IRK, as well as analyses of the unique features of time integration-based SOD methods.

A. Solution Operator

The solution operator $\boldsymbol{\mathcal{T}}(h)$ associated with DCPPS is the operator transforming the initial condition $\boldsymbol{\varphi}$ at time instant t to the solution segment at a later time instant $t + h$ [18], where h is the transfer step-length satisfying $0 \leq h \leq \tau_{\max}$.

$$(\boldsymbol{\mathcal{T}}(h)\boldsymbol{\varphi})(t) = \Delta\boldsymbol{x}_h(t) = \Delta\boldsymbol{x}(t+h),\ t \in [-\tau_{\max}, 0]. \tag{6}$$

Corresponding to (2), $\boldsymbol{\mathcal{T}}(h)$ has two segments.

1) Time Integration: When $t \in [-h, 0]$ and $t + h \in [0, h]$, $\Delta \boldsymbol{x}_h(t)$ is the solution to the following ordinary differential equation

$$\Delta \dot{\boldsymbol{x}}_h(t) = \tilde{\boldsymbol{A}}_0 \Delta \boldsymbol{x}_h(t) + \sum_{i=1}^{m} \tilde{\boldsymbol{A}}_i \Delta \boldsymbol{x}_h(t - \tau_i) \; t \in [-h, 0] \quad (7)$$

which follows from the Picard's Existence and Uniqueness Theorem [19].

2) Shift: For $t \in [-\tau_{\max}, -h]$, $\Delta \boldsymbol{x}_h(t)$ is always the initial condition part of (2) since $t + h \leq 0$. $\boldsymbol{\mathcal{T}}(h)$ in this case is a shift.

In summary, the expression of $\boldsymbol{\mathcal{T}}(h)$ can be explicitly reformulated as follows.

$$\Delta \boldsymbol{x}_h(t) = \begin{cases} \boldsymbol{\varphi}(0) + \displaystyle\int_0^t \left(\tilde{\boldsymbol{A}}_0 \Delta \boldsymbol{x}_h(s) + \sum_{i=1}^{m} \tilde{\boldsymbol{A}}_i \Delta \boldsymbol{x}_h(s - \tau_i) \right) \mathrm{d}s \\ \qquad\qquad\qquad\qquad t \in [-h,\, 0] \\ \boldsymbol{\varphi}(t + h) \qquad\qquad\qquad\qquad t \in [-\tau_{\max},\, -h]. \end{cases} \quad (8)$$

For small signal stability analysis of DCPPS, it is the eigenvalue λ rather than the solution $\Delta \boldsymbol{x}_h(t)$ that interests power engineers. To obtain λ, one can compute the nonzero eigenvalues μ of $\boldsymbol{\mathcal{T}}(h)$ instead by utilizing the following spectral mapping [16].

$$\begin{cases} \mu = \mathrm{e}^{\lambda h}, \mu \in \sigma(\boldsymbol{\mathcal{T}}(h)) \backslash \{0\} & (9) \\ \lambda = \dfrac{1}{h} \ln \mu = \dfrac{1}{h} (\ln |\mu| + \mathrm{j} \arg \mu) & (10) \end{cases}$$

where $|\mu|$ and $\arg \mu$ are modulus and principal argument of μ, respectively. $\arg \mu$ can be computed as $\arcsin(\frac{\mathrm{Im}(\mu)}{|\mu|}) \mathrm{mod} \frac{\pi}{h}$ [20]. $\sigma(\cdot)$ denotes the spectrum. $\backslash$ denotes the operation of set difference.

Since finding eigenvalues of $\boldsymbol{\mathcal{T}}(h)$ is an infinite-dimensional problem on a Banach space, two time integration schemes, i.e., LMS and IRK, are presented in the following sections to discretize $\boldsymbol{\mathcal{T}}(h)$ first. Then, a reduced set of eigenvalues are calculated from the finite-dimensional approximate matrices to $\boldsymbol{\mathcal{T}}(h)$.

B. SOD-LMS

The LMS discretization of $\boldsymbol{\mathcal{T}}(h)$ is achieved by evaluating $\Delta \boldsymbol{x}(t + h)$ shown by (8) at $N + k$ grid points of Ω_N, as illustrated by Fig. 1(a).

At point $t = \theta_0$, the linear k-step method with a step-length of h is applied to the time integration segment of (8) [21]. At the remaining points, the discretization of the second segment of (8) is straightforward by applying the shift property [22].

By representing the system states at points of $h + \Omega_N$ in terms of the states at points of Ω_N, the discrete counterpart of (8) is obtained. The resultant coefficient matrix $\boldsymbol{T}_N \in \mathbb{R}^{(N+k)n \times (N+k)n}$

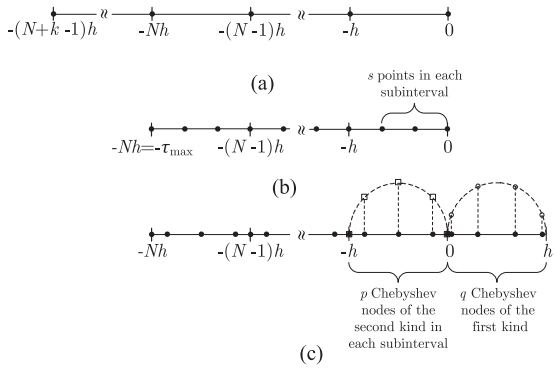


Fig. 1. Different schemes for discretizing $\boldsymbol{\mathcal{T}}(h)$. (a) SOD-LMS. The set Ω_N consists of $N + k$ equally spaced discrete points defined by $\theta_j = -jh$, where $j = 0, 1, \ldots, N + k - 1$, $h = \tau_{\max}/N$, k is the number of steps in the LMS method. (b) SOD-IRK. The set Ω_{Ns} consists of Ns points defined by $\theta_{j,q} = -jh + c_q h$, where $j = 1, \ldots, N$, $q = 1, \ldots, s$, c_q are abscissae of an s stage IRK method. (c) SOD-PS [16]. The sub-intervals $[-(j+1)h, -jh]$ $(j = 0, \ldots, N-1)$ and $[0, h]$ are respectively discretized by the shifted p and q nodes of the second and first kind Chebyshev polynomials. Essentially, the discrete points are projections of equally spaced points on the upper half of the unit circle.

is an approximant to $\boldsymbol{\mathcal{T}}(h)$.

$$\boldsymbol{T}_N = \begin{bmatrix} & \boldsymbol{\Gamma}_N & \\ \hline \boldsymbol{I}_n & & \boldsymbol{0} \\ & \ddots & \vdots \\ & \boldsymbol{I}_n \;\; \boldsymbol{0} & \end{bmatrix} \quad (11)$$

where $\boldsymbol{\Gamma}_N \in \mathbb{R}^{n \times (N+k)n}$ is the first block rows of $\boldsymbol{T}_N$. $\boldsymbol{I}_n \in \mathbb{R}^{n \times n}$ is an identity matrix.

By applying the methodology presented in [13], [15], $\boldsymbol{\Gamma}_N$ in the multiple time delay case can be explicitly expressed in terms of system state matrices $\tilde{\boldsymbol{A}}_i$ $(i = 0, \ldots, m)$.

$$\boldsymbol{\Gamma}_N = \boldsymbol{R}_N^{-1} \boldsymbol{\Sigma}_N \quad (12)$$

$$\boldsymbol{R}_N = \alpha_k \boldsymbol{I}_n - h \beta_k \tilde{\boldsymbol{A}}_0 \quad (13)$$

$$\boldsymbol{\Sigma}_N = \boldsymbol{\ell}_{m+1}^{\mathrm{T}} \otimes \boldsymbol{I}_n + \sum_{i=0}^{m} \boldsymbol{\ell}_i^{\mathrm{T}} \otimes \tilde{\boldsymbol{A}}_i \quad (14)$$

where $\otimes$ denotes the Kronecker product. α_j and β_j $(j = 0, \ldots, k)$ are coefficients of linear k-step method. $\boldsymbol{\ell}_i \in \mathbb{R}^{(N+k)\times 1}$ $(i = 0, \ldots, m+1)$ are constant Lagrange interpolation vectors and dependent on α_j and β_j $(j = 1, \ldots, k)$, as well as the interpolation coefficients.

C. SOD-IRK

The principle of SOD-IRK is similar to that of SOD-LMS [23]. As Fig. 1(b) illustrates, the N sub-intervals of length h are further discretized by the abscissae of an s-stage IRK method, resulting in a set Ω_{Ns} of Ns grid points.

For mnemonic convenience, the coefficients of IRK, a_{ij}, b_i and c_i, $i, j = 1, \ldots, s$, are usually arranged in the Butcher's tableau $(\boldsymbol{A}, \boldsymbol{b}, \boldsymbol{c})$ [24].

By applying IRK to evaluate system states at $\theta_{0,q}$ $(q = 1, \ldots, s)$ and using the shift property to assess system states at the remaining points of Ω_{Ns}, the approximate matrix to $\boldsymbol{\mathcal{T}}(h)$ is derived.

$$\boldsymbol{T}_{Ns} = \begin{bmatrix} & \boldsymbol{\Gamma}_{Ns} & \\ \hdashline \boldsymbol{I}_{sn} & & \boldsymbol{0} \\ & \ddots & \vdots \\ & \boldsymbol{I}_{sn} & \boldsymbol{0} \end{bmatrix} \tag{15}$$

The Kronecker product reformulation of the first block rows $\boldsymbol{\Gamma}_{Ns} \in \mathbb{R}^{sn \times Nsn}$ is given as follows.

$$\boldsymbol{\Gamma}_{Ns} = \boldsymbol{R}_{Ns}^{-1} \boldsymbol{\Sigma}_{Ns} \tag{16}$$

$$\boldsymbol{R}_{Ns} = \boldsymbol{I}_{sn} - \boldsymbol{A} \otimes (h\tilde{\boldsymbol{A}}_0) \tag{17}$$

$$\boldsymbol{\Sigma}_{Ns} = \boldsymbol{L}_0 \otimes \boldsymbol{I}_n + \sum_{i=1}^{m} (h\boldsymbol{A}\boldsymbol{L}_i) \otimes \tilde{\boldsymbol{A}}_i \tag{18}$$

where $\boldsymbol{L}_i \in \mathbb{R}^{s \times Ns}$ $(i = 0, \ldots, m)$ are constant Lagrange interpolation matrices.

D. Features of Time Integration-Based SOD Methods

The unique features of SOD-LMS/IRK can be derived from the expressions of the discretized matrices $\boldsymbol{T}_N$ and $\boldsymbol{T}_{Ns}$ as well as the principles of multistep and one-step algorithms [24].

1) $\boldsymbol{T}_N$ and $\boldsymbol{T}_{Ns}$ Are Highly Structured: They are block matrices and highly sparse. Their sub-diagonal blocks are identity matrices with dimensions of n and sn, respectively.

2) $\boldsymbol{\Gamma}_n$ and $\boldsymbol{\Gamma}_{Ns}$ Are Explicit: They are explicit about system state matrices $\tilde{\boldsymbol{A}}_i$ $(i = 0, \ldots, m)$. The explicitness allows one to fully exploit the inherent sparsity in augmented system matrices (presented in (3) and (4)) and makes the methods particularly suitable for solving large DCPPS.

3) SOD-LMS/IRK Are More Efficient Than SOD-PS: No Kronecker product is involved in $\boldsymbol{R}_N$ because LMS needs to evaluate (2) once to obtain $\Delta\boldsymbol{x}(0)$. In contrast, IRK requires s evaluations and hence Kronecker product emerges in $\boldsymbol{R}_{Ns}$. For large DCPPS, the direct inverse of $\boldsymbol{R}_{Ns}$ is unavailable [25]. Thus, iterative methods should be employed by SOD-IRK to compute the product between the inverse of $\boldsymbol{R}_{Ns}$ and a vector (MIVP), leading to more computational burden and lower efficiency than SOD-LMS.

Nevertheless, SOD-IRK is more efficient than SOD-PS in [16], whose efficiency is also constrained by the iterative solution of MIVP involved in eigenvalue computation. It can be seen from Fig. 1(b) and (c) that the sizes of the MIVPs in SOD-IRK and SOD-PS are sn and Nn, respectively. Since $s \ll N$, SOD-IRK gains lower computational burden and higher efficiency in iteratively solving the involved MIVP when compared to SOD-PS.

IV. Determination of Step-Length and LMS/IRK Methods

This section discusses how to determine the step-length h and select LMS/IRK methods for fine discretizing $\boldsymbol{\mathcal{T}}(h)$.

A. Determination of Step-Length h

Small step-length h can ensure preservation of the stability properties of DCPPS by SOD-LMS/IRK. However, the sizes of $\boldsymbol{T}_N$ and $\boldsymbol{T}_{Ns}$ are inversely proportional to h. Thus, a large h is always expected to reduce the burden of computing eigenvalues from $\boldsymbol{T}_N$ and $\boldsymbol{T}_{Ns}$.

To this aim, a step-length heuristic is proposed in [21] by virtue of the correspondence between the absolute stability regions of LMS/IRK (the region in the $h\lambda$ plane where all roots ζ of the methods' characteristic equation satisfy $|\zeta(h\lambda)| \leq 1$) [24] scaled by $1/h$ and the delay-independent stability of DCPPS (i.e., stability for all values of $\tau_i > 0, i = 1, \ldots, m$).

$$h = 0.9 \frac{\rho_\varepsilon}{||\tilde{\boldsymbol{A}}_0|| + \sum_{j=1}^{m} ||\tilde{\boldsymbol{A}}_j||} \tag{19}$$

where 0.9 is a safety factor. ε is a number small enough so that $\varepsilon/h \ll 1$. Given a small value ε, ρ_ε is the safety radius of the absolute stability region of the LMS/IRK methods [21].

It should be emphasized that the step-length h determined by (19) is a sufficient condition under which a given LMS/IRK method retains the delay-independent stability of DCPPS. Considering $\tau_{\max}$ is typically less than 1.0 s, the setting $N \geq 10$ or $h \leq 0.1$ is sufficient to ensure that critical oscillation modes of DCPPS can be recovered by SOD-LMS/IRK.

B. Selection of LMS/IRK Methods

LMS and IRK are essentially two large classes of time integration methods. It is of great importance to investigate which method can guarantee the accuracy of discretizing $\boldsymbol{\mathcal{T}}(h)$ so that the small signal stability of DCPPS can be accurately and reliably captured.

1) Selection of LMS/IRK Methods: It is well-understood that the physical power system is a stiff system with significantly different time scales. In addition, it has been revealed that the stiffness of DCPPS is independent of time delays, which can be validated by visualization of the sequences of eigenvalues with decreasing real parts (see [14], [16]).

It follows from [24] that *explicit* time integration methods do not work for problems described by stiff differential equations, e.g., the explicit Adams-Bashforth (AB) LMS method. In the case study sections of this paper, the performances of two classic implicit LMS methods, i.e., implicit Adams-Moulton (AM) and backward differentiation formulas (BDF), in approximating $\boldsymbol{\mathcal{T}}(h)$ are intensively investigated. Furthermore, numerical results on discretizing solution operator by typical IRK methods, including Radau-IA, Radau-IIA and Gauss, are presented.

2) Selection of the Step k and Stage s: The rules of thumb on selecting the step k of LMS and the stage s of IRK are given as follows by analyzing the local error in estimating the rightmost eigenvalue of DCPPS, i.e., $\mathcal{O}(h^p)$, in terms of the order p of LMS/IRK methods [24]. Generally, p varies proportionally with k or s. Specifically, for the k-step AB, AM and BDF methods, $p = k,\ k+1,\ k$, respectively, where $k = 1$–6. For the s-stage Radau-IA, Radau-IIA and Gauss methods, $p = 2s-1,\ 2s-1,\ 2s$, respectively, where $s = 1, 2, 3$.

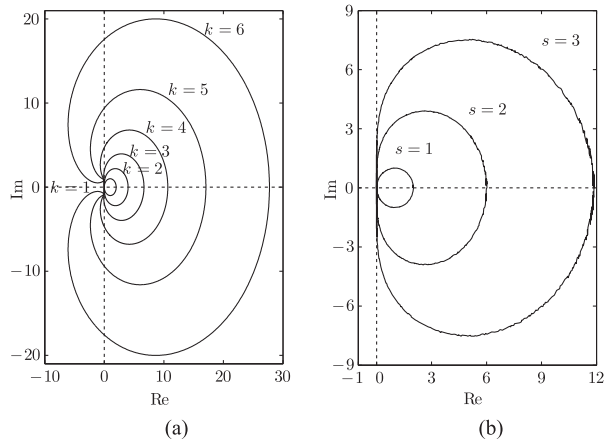


Fig. 2. Boundaries of the absolute stability regions for LMS and IRK. For each method, the unbounded absolute stability regions are the exterior of curves depicted. (a) LMS with BDF of $k = 1$–6. (b) IRK with Radua IA (IIA) of $s = 1$–3.

It is natural to choose a higher order of p to reduce the estimation error $\mathcal{O}(h^p)$. However, the rationale behind (19) is that the absolute stability regions of the LMS/IRK methods scaled by $1/h$ mimic the left-half plane up to some accuracy ε. In this sense, the LMS/IRK methods with *unbounded* stability regions *completely or nearly* covering the left-half plane is highly suggested, including BDF methods with $k = 2$–4 and all IRK methods, as depicted in Fig. 2.

V. Eigen-Analysis of Large DCPPS by SOD-LMS/IRK

In this section, a reduced set of critical oscillation modes of DCPPS are efficiently computed from the discretized matrices $\boldsymbol{T}_N$ and $\boldsymbol{T}_{Ns}$ generated by SOD-LMS/IRK, where the rotation-and-multiplication preconditioning technique and the sparse eigenvalue algorithm IRA are applied.

A. Rotation-and-Multiplication Preconditioning

To efficiently compute critical oscillation modes of DCPPS by SOD-based methods, the rotation-and-multiplication preconditioning technique presented in [16] is indispensable. The preconditioning enhances the dispersion of the eigenvalues μ of $\boldsymbol{\mathcal{T}(h)}$, leading to an accelerated convergence rate of IRA used to compute them. It is achieved by first rotating the complex plane in the anti-clockwise direction by $\theta\,(=\arcsin(\zeta))$ radian and then multiplying the plane by $\frac{1}{\alpha}$ $(\alpha > 1)$. By using the preconditioning, the eigenvalues of DCPPS with damping ratios less than the threshold are mapped to those of $\boldsymbol{\mathcal{T}(h)}$ located outside the unitary circle and can be computed with priority. The preconditioned eigenvalue λ' relates to λ of DCPPS by:

$$\lambda' = \alpha\lambda \mathrm{e}^{-\mathrm{j}\theta}. \tag{20}$$

Correspondingly, the preconditioned characteristic equation of DCPPS can be obtained by substituting λ in (4) with $\frac{1}{\alpha}\mathrm{e}^{\mathrm{j}\theta}\lambda'$.

$$\left(\tilde{\boldsymbol{A}}_0' + \sum_{i=1}^{m} \tilde{\boldsymbol{A}}_i' \mathrm{e}^{-\lambda'\tau_i'}\right)\boldsymbol{v} = \lambda'\boldsymbol{v} \tag{21}$$

where

$$\begin{cases} \tilde{\boldsymbol{A}}_0' = \alpha\tilde{\boldsymbol{A}}_0 \mathrm{e}^{-\mathrm{j}\theta},\ \tilde{\boldsymbol{A}}_i' = \alpha\tilde{\boldsymbol{A}}_i \mathrm{e}^{-\mathrm{j}\theta} & (22) \\ \tau_i' = \dfrac{\tau_i \mathrm{e}^{\mathrm{j}\theta}}{\alpha} \approx \dfrac{\tau_i}{\alpha},\ i = 1, \ldots, m. & (23) \end{cases}$$

Considering the delay approximation made in (23), λ' is essentially an approximant to the true solution to (21). It should be noted that the impact of delay approximation on accurately estimating λ' has been analyzed in [16].

B. Explicit Expressions of $\boldsymbol{T}_{N'}$ and $\boldsymbol{T}_{N's}$

After the preconditioning, $\boldsymbol{T}_N$ and $\boldsymbol{T}_{Ns}$ become $\boldsymbol{T}_{N'}$ and $\boldsymbol{T}_{N's}$, respectively.

$$\boldsymbol{T}_{N'} = \begin{bmatrix} \boldsymbol{\Gamma}_{N'} \\ \hdashline \boldsymbol{I}_{(N'+k-1)n} \quad \boldsymbol{0} \end{bmatrix},\ \boldsymbol{T}_{N's} = \begin{bmatrix} \boldsymbol{\Gamma}_{N's} \\ \hdashline \boldsymbol{I}_{(N'-1)sn} \quad \boldsymbol{0} \end{bmatrix} \tag{24}$$

where $N' = \lceil \tau_{\max}/(\alpha h) \rceil$, $\boldsymbol{T}_{N'} \in \mathbb{C}^{(N'+k)n \times (N'+k)n}$ and $\boldsymbol{T}_{N's} \in \mathbb{C}^{N'sn \times N'sn}$.

The block rows $\boldsymbol{\Gamma}_{N'} \in \mathbb{C}^{n \times (N'+k)n}$ and $\boldsymbol{\Gamma}_{N's} \in \mathbb{C}^{sn \times N'sn}$ are obtained by replacing $\tilde{\boldsymbol{A}}_i$ in (12)–(14) and (16)–(18) by $\tilde{\boldsymbol{A}}_i'\,(i = 0, \ldots, m)$. In addition, $\boldsymbol{\ell}_i\,(i = 0, \ldots, m+1)$ and $\boldsymbol{L}_j\,(j = 0, \ldots, m)$ are updated as $\boldsymbol{\ell}_i' \in \mathbb{R}^{(N'+k)\times 1}$ and $\boldsymbol{L}_j' \in \mathbb{R}^{N's\times 1}$ respectively by taking (23) into consideration.

$$\boldsymbol{\Gamma}_{N'} = \boldsymbol{R}_{N'}^{-1}\boldsymbol{\Sigma}_{N'}' \tag{25}$$

$$\boldsymbol{R}_{N'} = \alpha_k \boldsymbol{I}_n - h\beta_k \tilde{\boldsymbol{A}}_0' \tag{26}$$

$$\boldsymbol{\Sigma}_{N'} = (\boldsymbol{\ell}_{m+1}')^{\mathrm{T}} \otimes \boldsymbol{I}_n + \sum_{i=0}^{m} (\boldsymbol{\ell}_i')^{\mathrm{T}} \otimes \tilde{\boldsymbol{A}}_i' \tag{27}$$

$$\boldsymbol{\Gamma}_{N's} = \boldsymbol{R}_{N's}^{-1}\boldsymbol{\Sigma}_{N's} \tag{28}$$

$$\boldsymbol{R}_{N's} = \boldsymbol{I}_{sn} - \boldsymbol{A} \otimes (h\tilde{\boldsymbol{A}}_0') \tag{29}$$

$$\boldsymbol{\Sigma}_{N's} = \boldsymbol{L}_0' \otimes \boldsymbol{I}_n + \sum_{i=1}^{m} (h\boldsymbol{A}\boldsymbol{L}_i') \otimes \tilde{\boldsymbol{A}}_i'. \tag{30}$$

The eigenvalue μ' of $\boldsymbol{T}_{N'}$ or $\boldsymbol{T}_{N's}$ relates λ' by:

$$\mu' = \mu^{\alpha \mathrm{e}^{-\mathrm{j}\theta}} = \mathrm{e}^{\lambda' h},\ \lambda' = \frac{1}{h}\ln\mu'. \tag{31}$$

C. Sparse Eigenvalue Computation Using IRA

Here IRA [26] is employed to compute a given number of eigenvalues μ' with the largest moduli from $\boldsymbol{T}_{N'}$ and $\boldsymbol{T}_{N's}$.

1) Overview of Efficient Implementation of SOD-LMS: In application of IRA to SOD-LMS, the most computationally expensive operation is iteratively generating Krylov sequences. Let $\boldsymbol{q}_j \in \mathbb{C}^{(N'+k)n\times 1}$ be the jth Krylov sequence. Then, $\boldsymbol{q}_{j+1}$ is computed as:

$$\boldsymbol{q}_{j+1} = \boldsymbol{T}_{N'} \cdot \boldsymbol{q}_j. \tag{32}$$

Since the sub-diagonal blocks of $\boldsymbol{T}_{N'}$ are $\boldsymbol{I}_n$, the last $(N'+k-1)n$ entries of $\boldsymbol{q}_{j+1}$ can be readily obtained by forward shifting n entries of $\boldsymbol{q}_j$. Thus, the key to compute $\boldsymbol{q}_{j+1}$ is

identified as determination of its first n entries.

$$\boldsymbol{z} = \boldsymbol{\Sigma}_{N'} \cdot \boldsymbol{q}_j \tag{33}$$

$$\boldsymbol{q}_{j+1}(1:n,1) = \boldsymbol{R}_{N'}^{-1} \cdot \boldsymbol{z}. \tag{34}$$

where $\boldsymbol{z} \in \mathbb{C}^{n\times 1}$ is an intermediate vector.

In the following, (33) and (34) are efficiently implemented by exploiting the sparsity in augmented system matrices.

2) Compute $\boldsymbol{z} = \boldsymbol{\Sigma}_{N'} \cdot \boldsymbol{q}_j$: First, $\boldsymbol{q}_j$ is converted to a matrix $\boldsymbol{Q} \in \mathbb{C}^{n\times(N'+k)}$ by taking every n entries as columns.

$$\boldsymbol{q}_j = \mathrm{vec}(\boldsymbol{Q}). \tag{35}$$

Then $\boldsymbol{z}$ can be efficiently computed by transforming the memory-consuming Kronecker products into ordinary MVPs.

$$\begin{aligned}\boldsymbol{z} = \boldsymbol{\Sigma}_{N'} \cdot \boldsymbol{q}_j &= \left((\boldsymbol{\ell}'_{m+1})^{\mathrm{T}} \otimes \boldsymbol{I}_n + \sum_{i=0}^{m} (\boldsymbol{\ell}'_i)^{\mathrm{T}} \otimes \tilde{\boldsymbol{A}}'_i \right) \mathrm{vec}(\boldsymbol{Q}) \\ &= \boldsymbol{Q}\boldsymbol{\ell}'_{m+1} + \tilde{\boldsymbol{A}}'_0(\boldsymbol{Q}\boldsymbol{\ell}'_0) + \sum_{i=1}^{m} \tilde{\boldsymbol{A}}'_i(\boldsymbol{Q}\boldsymbol{\ell}'_i).\end{aligned} \tag{36}$$

The cost of calculating $\boldsymbol{z}$ is dominated by $\tilde{\boldsymbol{A}}'_0(\boldsymbol{Q}\boldsymbol{\ell}'_0) \triangleq \alpha \mathrm{e}^{-\mathrm{j}\theta}\boldsymbol{z}_0$, where $\boldsymbol{z}_0 = \tilde{\boldsymbol{A}}_0(\boldsymbol{Q}\boldsymbol{\ell}'_0)$ can be efficiently solved from the following augmented equation by taking (3) into account.

$$\begin{bmatrix} \boldsymbol{z}_0 \\ \boldsymbol{0} \end{bmatrix} = \begin{bmatrix} \boldsymbol{A}_0 & \boldsymbol{B}_0 \\ \boldsymbol{C}_0 & \boldsymbol{D}_0 \end{bmatrix} \begin{bmatrix} \boldsymbol{Q}\boldsymbol{\ell}'_0 \\ \boldsymbol{w} \end{bmatrix} \tag{37}$$

where $\boldsymbol{w}$ is an $l \times 1$ ancillary vector.

3) Compute $\boldsymbol{q}_{j+1}(1:n,\,1) = \boldsymbol{R}_{N'}^{-1} \cdot \boldsymbol{z}$: First, $\boldsymbol{R}_{N'}$ is reformulated by inserting (22) and (3) into (26).

$$\boldsymbol{R}_{N'} = \boldsymbol{A}'_0 - \boldsymbol{B}'_0\boldsymbol{D}_0^{-1}\boldsymbol{C}_0 \tag{38}$$

where $\boldsymbol{A}'_0$ and $\boldsymbol{B}'_0$ have identical sparsity with $\boldsymbol{A}_0$ and $\boldsymbol{B}_0$, respectively.

$$\begin{cases} \boldsymbol{A}'_0 = \alpha_k \boldsymbol{I}_n - h\beta_k\alpha\boldsymbol{A}_0\mathrm{e}^{-\mathrm{j}\theta} \\ \boldsymbol{B}'_0 = -h\beta_k\alpha\boldsymbol{B}_0\mathrm{e}^{-\mathrm{j}\theta}. \end{cases} \tag{39}$$

Accordingly, $\boldsymbol{q}_{j+1}(1:n,\,1)$ can be directly solved from the following augmented equation.

$$\begin{bmatrix} \boldsymbol{A}'_0 & \boldsymbol{B}'_0 \\ \boldsymbol{C}_0 & \boldsymbol{D}_0 \end{bmatrix} \begin{bmatrix} \boldsymbol{q}_{j+1}(1:n,\,1) \\ \boldsymbol{w} \end{bmatrix} = \begin{bmatrix} \boldsymbol{z} \\ \boldsymbol{0} \end{bmatrix}. \tag{40}$$

In summary, the computational complexity of one IRA iteration in SOD-LMS is characterized by an MVP in solving (33) and an MIVP in solving (34), which is almost the same as the EIGD method presented in [15].

4) Efficient Implementation of SOD-IRK: Considering the structure similarity in the discretized matrices $\boldsymbol{T}_{N'}$ and $\boldsymbol{T}_{N's}$, it is straightforward to generalize the sparse implementation of SOD-LMS as presented in the above subsections to SOD-IRK. The difference lies in the variant of (34), i.e.,

$$\boldsymbol{q}_{j+1}(1:sn,1) = \boldsymbol{R}_{N's}^{-1} \cdot \boldsymbol{z}. \tag{41}$$

Unlike $\boldsymbol{R}_{N'}$, the direct inversion of $\boldsymbol{R}_{N's}$ is unavailable since it is essentially the sum of two Kronecker products [25]. Thus, (41) can only be computed by iterative solvers [27] until the convergence is reached.

$$\boldsymbol{R}_{N's} \cdot \boldsymbol{q}_{j+1}^{(k)}(1:sn,1) = \boldsymbol{z} \tag{42}$$

where $\boldsymbol{q}_{j+1}^{(k)}(1:sn,1)$ denotes the solution to $\boldsymbol{q}_{j+1}(1:sn,1)$ after the kth iteration. Here the left-hand-side of (42) can be efficiently implemented by applying the unique property of Kronecker product [15], resulting in a matrix product $\tilde{\boldsymbol{A}}'_0(\boldsymbol{Q}\boldsymbol{A}) \triangleq \tilde{\boldsymbol{A}}'_0[\tilde{\boldsymbol{q}}_1, \ldots, \tilde{\boldsymbol{q}}_s]$. The ordinary MVPs $\tilde{\boldsymbol{A}}'_0\tilde{\boldsymbol{q}}_i$ $(i = 1, \ldots, s)$ can be efficiently computed in the same way as (37).

D. Restore λ From μ'

Once an eigenvalue μ' is obtained from $\boldsymbol{T}_{N'}$ or $\boldsymbol{T}_{N's}$ by IRA, the estimate to eigenvalue of DCPPS can be transformed back from the inversion of (20) and (31) by:

$$\hat{\lambda} = \frac{1}{\alpha}\mathrm{e}^{\mathrm{j}\theta}\lambda' = \frac{1}{\alpha h}\mathrm{e}^{\mathrm{j}\theta}\ln\mu'. \tag{43}$$

The corresponding eigenvector $\hat{\boldsymbol{v}}$ can be directly estimated by the first n entries of the corresponding Krylov sequence. Actually, $\hat{\lambda}$ and $\hat{\boldsymbol{v}}$ are still estimates of the true solution $\{\lambda, \boldsymbol{v}\}$. The roots of the estimation errors and the corresponding countermeasures are as follows.

1) Solution Operator Discretization: The accuracy of $\boldsymbol{T}_{N'}$ or $\boldsymbol{T}_{N's}$ in approximating $\boldsymbol{\mathcal{T}}(h)$ determines the accuracy of $\hat{\lambda}$ and $\hat{\boldsymbol{v}}$, which can be guaranteed by fine granularity of the discretization, i.e., small h and large N.

2) Delay Approximations in Rotation-and-Multiplication Preconditioning: The estimation error introduced by (23) can only be eliminated by utilizing the Newton's method [14], where $\hat{\lambda}$ and $\hat{\boldsymbol{v}}$ are taken as the initial guesses. The true solution pair $\{\lambda, \boldsymbol{v}\}$ of (5) are accordingly obtained.

VI. Case Studies

In this section, the accuracy, efficiency and scalability of time integration-based SOD-LMS/IRK methods are intensively studied on the 16-machine 68-bus test system and two real-life large transmission systems.

A. Case 1: The 16-Generator 68-Bus Test System

The details on the test system and two WADCs installed on G_2 and G_5 can be found in [16], [28]. The feedback and output time delays are assumed to be $\tau_{\mathrm{f}1} = 150$ ms, $\tau_{\mathrm{o}1} = 90.5$ ms, $\tau_{\mathrm{f}2} = 70$ ms and $\tau_{\mathrm{o}2} = 37.4$ ms. The number of state variables is $n = 200$. All tests are carried out on a 3.4 GHz desktop computer with 8 GB RAM. The convergence tolerances for IRA and Newton correction are 10^{-6}.

Firstly, the basis of SOD-LMS/IRK, i.e., the accuracy of $\boldsymbol{T}_N$ and $\boldsymbol{T}_{Ns}$ in approximating $\boldsymbol{\mathcal{T}}(h)$, is validated. Fig. 3 shows the eigenvalues $\hat{\mu}$ of $\boldsymbol{T}_N$ and $\boldsymbol{T}_{Ns}$ with the largest moduli against the accurate eigenvalues μ of $\boldsymbol{\mathcal{T}}(h)$, which are computed by SOD-PS in [16] and corrected by Newton iterations. Parameter settings are: $h = 0.0075$ s, $N = 20$, $\alpha = 1$ and $\theta = 0$; SOD-LMS with BDF of $k = 3$; SOD-IRK with Radau-IIA of $s = 2$; SOD-PS with $p = q = 3$.

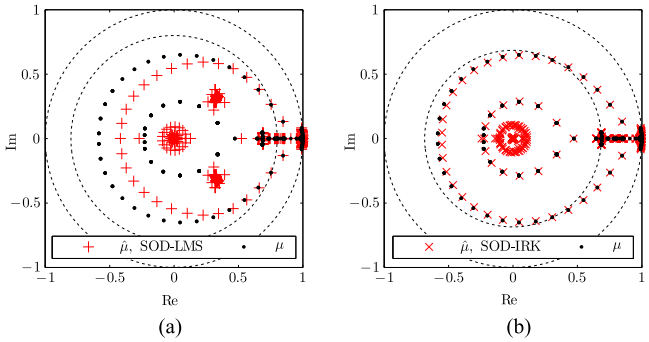


Fig. 3. Estimates $\hat{\mu}$ to accurate eigenvalues μ of $\boldsymbol{\mathcal{T}}(h)$ computed from $\boldsymbol{T}_N$ by SOD-LMS (left) and from $\boldsymbol{T}_{Ns}$ by SOD-IRK (right). μ are obtained by SOD-PS in [16] with Newton correction and (9).

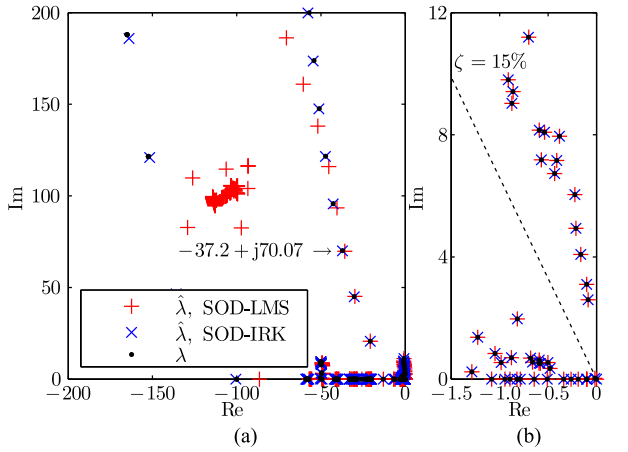


Fig. 4. Estimates $\hat{\lambda}$ to accurate eigenvalues λ of DCPPS computed by SOD-LMS and SOD-IRK. Figure (b) is a zoomed-in view of figure (a). λ are obtained by SOD-PS in [16] with Newton correction.

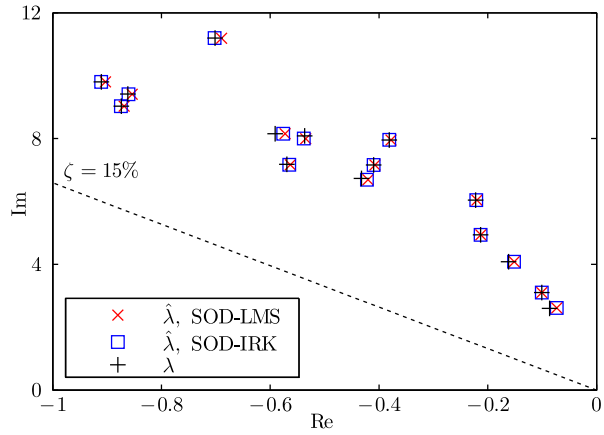


Fig. 5. Estimates $\hat{\lambda}$ to accurate eigenvalues λ of DCPPS computed by SOD-LMS and SOD-IRK with $\alpha = 2$ and $\theta = 8.63°$. λ are computed by SOD-PS in [16] with Newton correction.

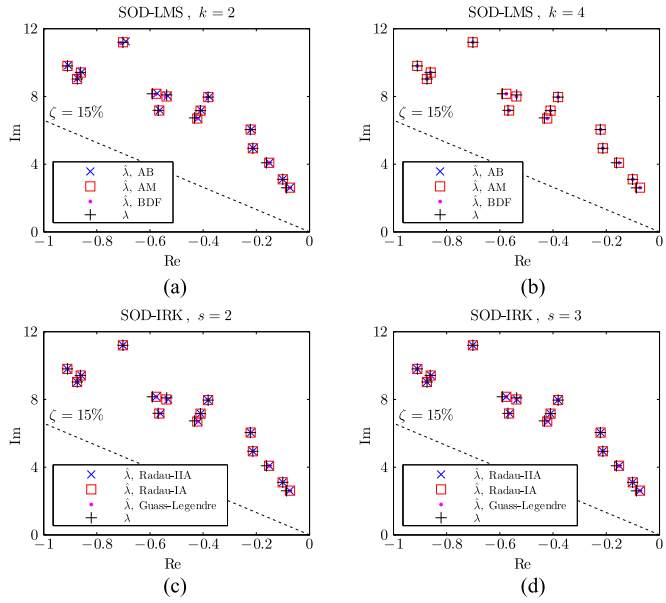


Fig. 6. Estimates $\hat{\lambda}$ to accurate eigenvalues λ of DCPPS computed from $\boldsymbol{T}_{N'}$ by SOD-LMS and from $\boldsymbol{T}_{N's}$ by SOD-IRK. In figures (a) and (b), SOD-LMS methods are AB, AM and BDF with $k = 2$ and 4, respectively. In figures (c) and (d), SOD-IRK methods are Radau-IIA, Radau-IA and Gauss with $s = 2$ and 3, respectively. λ are computed by SOD-PS in [16] with Newton correction.

It can be seen from Fig. 3 that SOD-LMS and SOD-IRK can accurately compute a subset of μ, which are located outside the inner circles. The radii of the inner circles are 0.8006 and 0.6854, respectively. All spurious eigenvalues with smaller moduli are located inside the inner circles. Since the largest modulus is 0.9999, the DCPPS can be determined to be stable. The eigenvalue estimations of DCPPS are obtained by substituting $\hat{\mu}$ into $\hat{\lambda} = \frac{1}{h}\ln\hat{\mu}$, as depicted in Fig. 4. It can be seen that $\hat{\lambda}$ computed by SOD-IRK is in close proximity to λ in the whole depicted region. In contrast, SOD-LMS can accurately capture eigenvalues located on the right of $\mathrm{Re}(\lambda) = -37.2$, which are sufficient for small signal stability analysis of DCPPS.

Secondly, by applying the rotation-and-multiplication preconditioning with $\alpha = 2$ and $\theta = 8.63°$, SOD-based methods are capable of estimating all electromechanical oscillation modes with priority, as shown in Fig. 5. Due to delay approximations made in (23), some estimates $\hat{\lambda}$ with large sensitivities to time delays slightly differ from their true values λ. By applying Newton correction, the biases between the estimations and the accurate eigenvalues can be readily removed. The estimates converge to their accurate counterparts.

Thirdly, the accuracy and efficiency of SOD with different LMS/IRK methods are intensively investigated. Part of the results are given in Fig. 6 and Table I. In all cases, $N' = 15$, $h = 0.005$ s, $\alpha = 2$ and $\theta = 8.63°$. The number of eigenvalues required to be computed is $r = 15$. It can be seen from Fig. 6(b) that AB with $k > 2$ cannot capture any eigenvalue of the system. Due to small step-length used to discretize the solution operator, μ of $\boldsymbol{\mathcal{T}}(h)$ are closely distributed, which causes about 40–50 IRA iterations in eigenvalue computation, as can be seen in column 10 of Table I. Nevertheless, in all cases, SOD-LMS consumes less than 3 s. It is the most efficient method because: 1) MIVP in IRA iterations is directly solved; 2) The dimensions of the discretized matrix $\boldsymbol{T}_{N'}$ to $\boldsymbol{\mathcal{T}}(h)$ are much smaller when comparing with SOD-IRK/PS.

Fourthly, efficiencies of SOD-LMS/IRK in this paper and in [17] are compared. The CPU time consumptions in cases

TABLE I
EFFICIENCY COMPARISON BETWEEN DIFFERENT SOD METHODS ($N' = 15$, $h = 0.005$ s, $\alpha = 2$, $\theta = 8.63°$)

Methods	Cases	LMS/IRK Methods	k	s	p	q	Dim	r	N_{IRA}	Time(s)	Time(s) /N_{IRA}
Proposed SOD-LMS	1	BDF	2	–	–	–	3400	15	38	2.07	0.054
	2	BDF	3	–	–	–	3600	15	44	2.38	0.054
	3	BDF	4	–	–	–	3800	15	39	2.45	0.063
	4	BDF	5	–	–	–	4000	15	41	2.67	0.065
	5	AM	5	–	–	–	4000	15	46	2.93	0.064
	6	AM	2	–	–	–	3400	15	44	2.31	0.053
	7	AM	3	–	–	–	3600	15	46	2.30	0.050
	8	AM	4	–	–	–	3800	15	46	2.74	0.060
	9	AB	2	–	–	–	3400	15	52	2.60	0.050
Proposed SOD-IRK	10	Radau-IIA	–	2	–	–	6000	15	39	5.92	0.152
	11	Radau-IIA	–	3	–	–	9000	15	41	7.78	0.190
	12	Radau-IA	–	2	–	–	6200	15	36	5.73	0.159
	13	Radau-IA	–	3	–	–	9200	15	43	9.33	0.217
	14	Gauss	–	2	–	–	9000	15	49	9.36	0.191
	15	Gauss	–	3	–	–	12000	15	40	10.24	0.256
SOD-PS in [16]	16	–	–	–	6	3	9200	15	43	16.44	0.382
	17	–	–	–	3	3	18200	15	39	25.25	0.647
SOD-LMS in [17][1,2]	18	BDF	2	–	–	–	3400	15	36	2.38	0.066
	19	BDF	3	–	–	–	3600	15	39	2.48	0.064
	20	BDF	4	–	–	–	3800	15	41	2.91	0.071
	21	BDF	5	–	–	–	4000	15	49	3.71	0.076
SOD-IRK in [17][1,2]	22	Radau-IIA	–	2	–	–	6000	15	42	33.21	0.791
	23	Radau-IIA	–	3	–	–	9000	15	44	81.24	1.846
Cayley-IRA (w/o delay)	24	–	–	–	–	–	200	15	4	0.29	0.073

[1]The GPU-based parallelization of QR is substituted by single-core CPU-based IRA, accompanying with LU factorization to directly solve (34) and (41).
[2]The sparsity in the lower block rows of $\boldsymbol{T}_{N'}$ and $\boldsymbol{T}_{N's}$, i.e., identity matrices, are utilized in the implementation, resulting in improved efficiency.

18–23 are listed in Table I. By comparing cases 18–21 with cases 1–4, it is clear that the computational burdens in solving (34) by SOD-LMS in the paper and in [17] are nearly the same. However, in cases 22 and 23, SOD-IRK in [17] consumes about 5 and 10 times of CPU time of the presented SOD-IRK in cases 10 and 11. It means that the iterative solution of (41) by the presented SOD-IRK is more efficient than SOD-IRK in [17].

Lastly, in the case of system without delay, IRA with Cayley transform (Cayley-IRA) [26], which has the similar rationale to the rotation-and-multiplication preconditioning, is used to compute $r = 15$ eigenvalues with the damping ratios less than 15%. As can be seen from case 24 in Table I, the computation costs less than 0.3 s. Nevertheless, the CPU time per iteration is almost the same as SOD-LMS, indicating comparative computational burden.

In summary, SOD-LMS/IRK/PS can accurately find a set of stability-determining eigenvalues μ of $\boldsymbol{\mathcal{T}}(h)$ with the largest moduli, corresponding to the eigenvalues λ of DCPPS with the smallest damping ratios. SOD-LMS features in its high efficiency. SOD-IRK/PS can capture more eigenvalues λ with greater real parts. However, the extra eigenvalues captured by SOD-IRK/PS provide no more useful information for small signal stability analysis of DCPPS.

B. Case 2: A 516-Bus Real-Life Transmission System

In this section, the scalability of SOD-LMS/IRK in analyzing large DCPPS is studied on the Shandong power grid with 516 buses, 114 generators, 936 transformers and transmission lines,

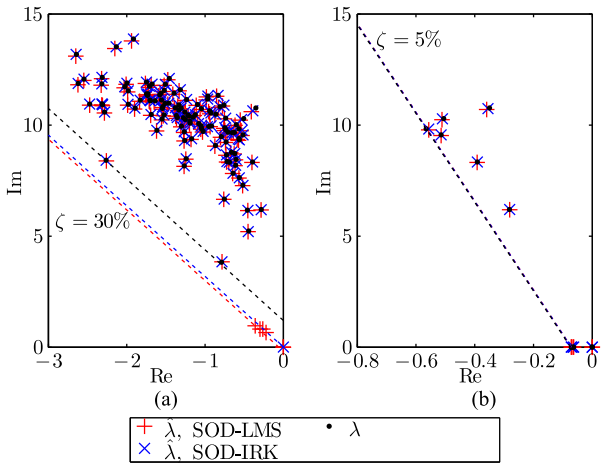


Fig. 7. Estimates $\hat{\lambda}$ of the system computed by SOD-LMS with BDF of $k = 2$ and SOD-IRK with Radau-IIA of $s = 2$. (a) $r = 120$, $\theta = 17.46°$, $\zeta = 30\%$. (b) $r = 12$, $\theta = 2.87°$, $\zeta = 5\%$. The dashed lines in red, blue and black are with identical slope of ζ and with the smallest Y-intercepts determined by three clusters of eigenvalues computed by SOD-LMS, SOD-IRK and SOD-PS in [16] with Newton correction.

and 299 loads [16]. The total delays in two wide-area damping control channels are 90 ms and 100 ms, respectively. The number of state variables of the system is 1128.

The criterion of scalability adopted here is *the inexistence of any spurious eigenvalues located on the right-hand side of the desired electromechanical oscillation modes*. With the aim, SOD methods with different LMS and IRK schemes, including AB, AM, BDF, Radau-IIA, Radau-IA and Gauss, are intensively studied. In all studies, $h = 0.005$ s, $N' = 10$, $\alpha = 2$. The accurate eigenvalues computed by SOD-PS in [16] with $p = q = 3$ are taken as benchmarks.

Considerable numerical studies reveal that SOD-LMS methods with BDF of $k = 2$–4 and all SOD-IRK methods are capable of accurately analyzing the real-life large system. As shown in Fig. 7(a), all electromechanical oscillation modes of the system are estimated by applying $\theta = 17.46°$ (i.e., $\zeta = 30\%$) and computing $r = 120$ eigenvalues. The red dashed line has the smallest Y-intercept because 5 eigenvalues near from the origin are captured by SOD-LMS. In contrast, SOD-IRK and SOD-PS obtain 4 and 5 spurious eigenvalues located to the left of $\mathrm{Re}(\lambda) = -35$, respectively. In Fig. 7(b), all eigenvalues with damping ratios less than $\zeta = 5\%$ (i.e., $\theta = 2.86°$) are captured for reliably and fast determining the system's stability against time delays. The small biases between the estimates and the accurate eigenvalues, as shown in Fig. 7, can be readily removed by the Newton's method.

Fig. 8 shows 120 estimates $\hat{\lambda}$ computed by SOD-LMS with AB of $k = 2$ and with AM of $k = 4$. It is clear that besides the desired low frequency oscillation modes, some spurious eigenvalues are also computed by the two SOD-LMS methods. Therefore, SOD-LMS methods with AM and AB are not suitable for reliably analyzing the small signal stability of large DCPPS.

Table II summarizes the computational time of different test cases. It can be concluded that SOD-LMS is always the most efficient method among various SOD methods. In addition, it

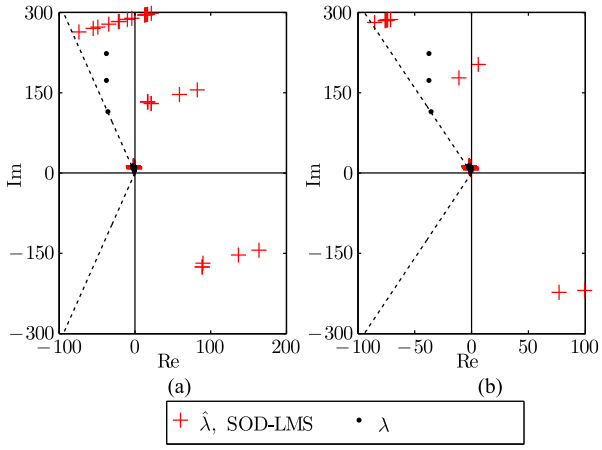


Fig. 8. $r = 120$ estimates $\hat{\lambda}$ of the system computed by SOD-LMS with AB and AM. (a) AB with $k = 2$. (b) AM with $k = 4$. λ are computed by SOD-PS in [16] with Newton correction.

TABLE II
EFFICIENCY COMPARISON IN ANALYZING THE 516-BUS SYSTEM

Methods	Cases	LMS/IRK Methods	k	s	p	q	Dim	θ (°)	r	N_{IRA}	Time(s)	Time(s) /N_{IRA}
Proposed SOD-LMS	1	BDF	2	–	–	–	13536	17.46	120	19	210.35	11.071
	2	BDF	3	–	–	–	14664	17.46	120	15	190.87	12.725
	3	BDF	4	–	–	–	15792	17.46	120	18	235.65	13.092
	4	BDF	2	–	–	–	13536	2.87	12	776	100.21	0.129
	5	BDF	3	–	–	–	14664	2.87	12	867	118.03	0.136
	6	BDF	4	–	–	–	15792	2.87	12	346	31.97	0.092
Proposed SOD-IRK	7	Radau-IIA	–	2	–	–	22560	17.46	120	17	350.99	20.646
	8	Radau-IIA	–	3	–	–	33840	17.46	120	16	498.19	31.137
	9	Radau-IIA	–	2	–	–	22560	2.87	12	810	278.90	0.344
	10	Radau IIA	–	3	–	–	33840	2.87	12	722	393.16	0.545
	11	Radau-IA	–	2	–	–	23688	17.46	120	17	383.49	22.558
	12	Radau-IA	–	3	–	–	34968	17.46	120	17	549.17	32.304
	13	Radau-IA	–	2	–	–	23688	2.87	12	717	279.60	0.390
	14	Radau-IA	–	3	–	–	34968	2.87	12	719	397.84	0.553
	15	Gauss	–	2	–	–	33840	17.46	100	16	496.82	31.051
	16	Gauss	–	3	–	–	45120	17.46	100	17	671.66	39.509
	17	Gauss	–	2	–	–	33840	2.87	12	634	334.35	0.527
	18	Gauss	–	3	–	–	45120	2.87	12	888	564.89	0.636
SOD-PS in [16]	19	–	–	–	3	3	34968	17.46	120	16	526.09	32.881
	20	–	–	–	6	3	68808	17.46	120	16	267.26	16.704
	21	–	–	–	3	3	34968	2.87	12	931	503.18	0.540
	22	–	–	–	6	3	68808	2.87	12	751	816.21	1.087
SOD-LMS in [17]	23	BDF	2	–	–	–	13536	17.46	120	18	220.22	12.234
	24	BDF	3	–	–	–	14664	17.46	120	15	207.28	13.819
	25	BDF	4	–	–	–	15792	17.46	120	18	253.45	14.081
	26	BDF	2	–	–	–	13536	2.87	12	831	211.29	0.254
	27	BDF	3	–	–	–	14664	2.87	12	764	194.28	0.254
	28	BDF	4	–	–	–	15792	2.87	12	372	57.83	0.155
SOD-IRK in [17]	29	Radau-IIA	–	2	–	–	22560	17.46	120	14	1046.17	74.726
	30	Radau-IIA	–	3	–	–	33840	17.46	120	13	2320.26	178.482
	31	Radau-IIA	–	2	–	–	22560	2.87	12	684	3539.90	5.175
	32	Radau-IIA	–	3	–	–	33840	2.87	12	Failed	–	–
Cayley-IRA (w/o delay)	33	–	–	–	–	–	1128	17.46	120	5	17.39	3.478
	34	–	–	–	–	–	1128	2.87	12	212	35.68	0.168

is noted that in cases of $\theta = 2.87°$ and $\zeta = 5\%$, all three SOD methods need a great number of IRA iterations due to several eigenvalues closely located on the negative real axis (see Fig. 7(b)). It also can be found from Table II that the SOD-LMS methods in the paper and in [17] consume nearly the same CPU time, while the presented SOD-IRK consumes $\frac{1}{6}$ to $\frac{1}{3}$ CPU time of SOD-IRK in [17]. Compared with the no delay cases 33 and 34, SOD-LMS costs 1 to 4 times of CPU time per iteration of Cayley-IRA.

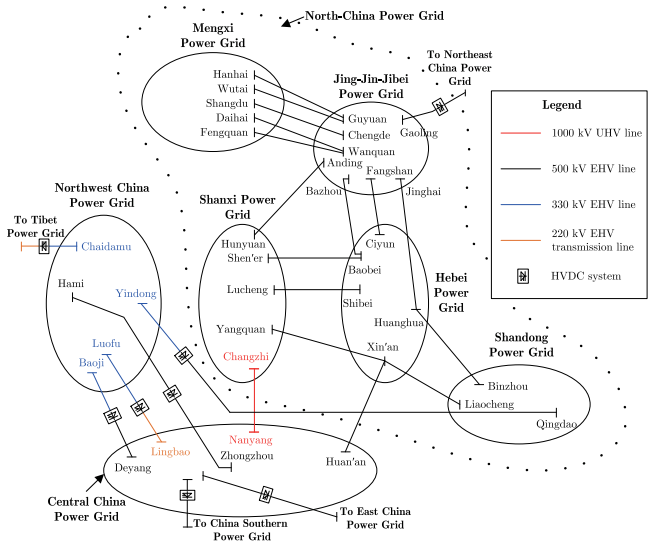


Fig. 9. Illustrative diagram of the ultra-high-voltage (UHV) North China-Central China interconnected power grid.

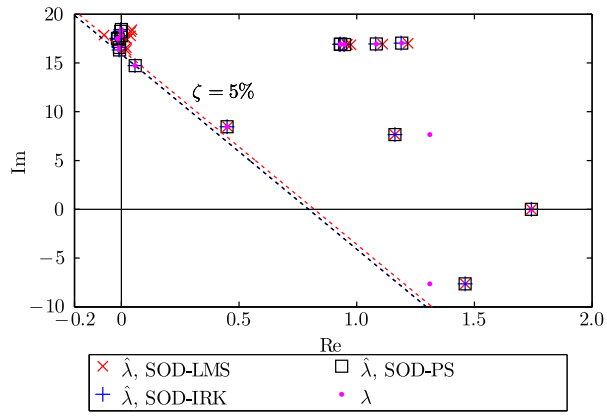


Fig. 10. $r = 20$ estimates $\hat{\lambda}$ to the unstable plant/local modes of the system by SOD-LMS, SOD-IRK and SOD-PS in [16], respectively. λ are computed by SOD-PS with Newton correction.

C. Case 3: A 33028-Bus Real-Life Interconnected System

The scalability and efficiency of SOD-LMS/IRK are further studied on the ultra-high-voltage (UHV) North China-Central China interconnected power grid, as illustrated in Fig. 9. The grid has 33028 buses, 2405 generators, 1991 induction motors and 16 HVDC transmission systems. The feedback and output time delays of two WADCs are assumed to be $\tau_{\text{f1}} = 120$ ms, $\tau_{\text{o1}} = 100$ ms, $\tau_{\text{f2}} = 100$ ms and $\tau_{\text{o2}} = 80$ ms. The number of system state variables is $n = 80577$.

Fig. 10 shows three sets of $r = 20$ estimates to unstable plant/local modes of the system by SOD-LMS with BDF, SOD-IRK with Radau-IIA and SOD-PS in [16]. The modes are dominated by hydraulic generators which are with small rated capacities in Mountain areas of Central China and connected to the main grid through long distance transmission lines. It can

TABLE III
EFFICIENCY COMPARISON IN ANALYZING THE 33028-BUS SYSTEM

Cases	Methods	k	s	p	q	N	h	Dim	θ (°)	r	N_{IRA}	Time(s)	Time(s) /N_{IRA}
1	Proposed SOD-LMS	3	–	–	–	20	0.006	1047501	2.87	20	177	7359	41.58
2	Proposed SOD-IRK	–	2	–	–	20	0.006	1611540	2.87	20	164	19543	119.16
3	SOD-PS in [16]	–	–	3	3	20	0.006	2497887	2.87	20	155	45009	290.38
4	SOD-LMS in [17]	3	–	–	–	20	0.006	1047501	2.87	20	Failed	–	–
5	SOD-IRK in [17]	–	2	–	–	20	0.006	1611540	2.87	20	Failed	–	–
6	Cayley-IRA (w/o delay)	–	–	–	–	–	–	80577	2.87	20	172	1416	8.23

be seen from the figure that the estimates obtained by SOD-LMS coincide with SOD-IRK and SOD-PS except those with $\mathrm{Im}(\hat{\lambda}) > 15$.

Table III summarizes the CPU time consumed by the three methods. It can be concluded that SOD-LMS is the most efficient, followed by SOD-IRK and then SOD-PS. Specifically, SOD-LMS and SOD-IRK respectively spend about $\frac{1}{6}$ and $\frac{1}{2}$ CPU time of SOD-PS. On the contrary, neither SOD-LMS nor SOD-IRK in [17] is capable of handling the system in this case because the inherent sparsity in the augmented state matrices is not exploited. Compared with the system without delay, SOD-LMS spends about 5 times of CPU time while it is 13 times the matrix dimension of Cayley-IRA.

VII. CONCLUSIONS

This paper presents scalable and efficient SOD-LMS and SOD-IRK for efficiently analyzing large DCPPS. The presented methods improve the scalability and efficiency of existing SOD-LMS/IRK in [17] and SOD-PS in [16]. Important conclusions are summarized as follows.

1) The SOD-LMS/IRK are endowed with scalability in efficiently analyzing large DCPPS by fully utilizing the inherent sparsity in augmented system state matrices. The methods feature in efficiently calculating critical eigenvalues of DCPPS with damping ratios less than a prespecified threshold by one run of the methods.
2) SOD-LMS/IRK generate highly structured approximate matrices to the solution operator. Exploitation of the structure ensures high efficiency of the methods. SOD-LMS is provided with the *highest* efficiency by directly solving the matrix-inversion-vector product (MIVP) involved in sparse eigenvalue computation. SOD-IRK gains higher efficiency than SOD-PS by cutting down the burden of iteratively solving the involved MIVP.
3) The backward differentiation formulas (BDF) LMS methods of $k = 2$–4 are recommended in analyzing the small signal stability of large DCPPS. The unbounded stability regions of these methods almost cover the whole left-half complex plane, leading to smaller dimensions of the discretized matrices to the solution operator and hence less computational burden in eigenvalue computation.

Application of SOD-LMS for designing WADCs with consideration of time delay impacts is part of our future research.

ACKNOWLEDGMENT

The authors also would like to thank the editor and anonymous reviewers for the valuable comments and suggestions.

REFERENCES

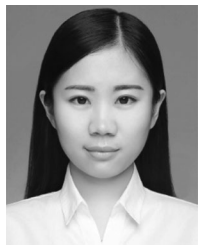
[1] A. G. Phadke and J. S. Thorp, *Synchronized Phasor Measurements and Their Applications*. New York, NY, USA: Springer, 2008.
[2] J. Zhao, Y. Zhang, P. Zhang, X. Jin, and C. Fu, "Development of a WAMS based test platform for power system real time transient stability detection and control," *Protection Control Modern Power Syst.*, vol. 1, no. 1, pp. 1–11, Jun. 2016. doi: 10.1186/s41601-016-0013-1.
[3] F. Bai *et al.*, "A measurement-based approach for power system instability early warning," *Protection Control Modern Power Syst.*, vol. 1, no. 1, pp. 1–9, Jun. 2016. doi: 10.1186/s41601-016-0014-0.
[4] J. W. Stahlhut, T. J. Browne, G. T. Heydt, and V. Vittal, "Latency viewed as a stochastic process and its impact on wide area power system control signals," *IEEE Trans. Power Syst.*, vol. 23, no. 1, pp. 84–91, Feb. 2008.
[5] F. Zhang, Y. Sun, L. Cheng, X. Li, J. H. Chow, and W. Zhao, "Measurement and modeling of delays in wide-area closed-loop control systems," *IEEE Trans. Power Syst.*, vol. 30, no. 5, pp. 2426–2433, Sep. 2014.
[6] H. Wu, K. S. Tsakalis, and G. T. Heydt, "Evaluation of time delay effects to wide-area power system stabilizer design," *IEEE Trans. Power Syst.*, vol. 19, no. 4, pp. 1935–1941, Nov. 2004.
[7] F. Milano and M. Anghel, "Impact of time delays on power system stability," *IEEE Trans. Circuit Syst. I*, vol. 59, no. 4, pp. 889–900, Apr. 2012.
[8] V. Bokharaie, R. Sipahi, and F. Milano, "Small-signal stability analysis of delayed power system stabilizers," in *Proc. Power Syst. Comput. Conf.*, Wroclaw, Poland, Aug. 18–22, 2014, pp. 1–7.
[9] F. Milano and I. Dassios, "Small-signal stability analysis for non-index 1 Hessenberg form systems of delay differential-algebraic equations," *IEEE Trans. Circuit Syst. I*, vol. 63, no. 9, pp. 1521–1530, Sep. 2016.
[10] F. Milano, "Delay-based numerical stability of the partitioned-solution approach," in *Proc. IEEE Power Energy Soc. General Meet.*, Boston, MA, USA, Jul. 17–21, 2016, pp. 1–5.
[11] S. Zhang and V. Vittal, "Design of wide-area power system damping controllers resilient to communication failures," *IEEE Trans. Power Syst.*, vol. 28, no. 4, pp. 4292–4300, Nov. 2013.
[12] J. Li, Z. Chen, D. Cai, W. Zhen, and Q. Huang, "Delay-dependent stability control for power system with multiple time-delays," *IEEE Trans. Power Syst.*, vol. 31, no. 3, pp. 2316–2326, May 2016.
[13] H. Ye, W. Gao, Q. Mou, and Y. Liu, "Iterative infinitesimal generator discretization-based method for eigen-analysis of large delayed cyber-physical power system," *Electr. Power Syst. Res.*, vol. 143, no. 1, pp. 389–399, Feb. 2017.
[14] H. Ye, Y. Liu, and P. Zhang, "Efficient eigen-analysis for large delayed cyber-physical power system using explicit infinitesimal generator discretization," *IEEE Trans. Power Syst.*, vol. 31, no. 3, pp. 2361–2370, May 2016.
[15] H. Ye, Q. Mou, and Y. Liu, "Enabling highly efficient spectral discretization-based eigen-analysis methods by Kronecker product," *IEEE Trans. Power Syst.*, vol. 32, no. 5, pp. 4148–4150, Sep. 2017.
[16] H. Ye, Q. Mou, and Y. Liu, "Calculation of critical oscillation modes for large delayed cyber-physical power system using pseudo-spectral discretization of solution operator," *IEEE Trans. Power Syst.*, vol. 32, no. 6, pp. 4464–4476, Nov. 2017.
[17] F. Milano, "Small-signal stability analysis of large power systems with inclusion of multiple delays," *IEEE Trans Power Syst.*, vol. 31, no. 4, pp. 3257–3266, Jul. 2016.
[18] O. Diekmann, S. A. v. Gils, S. M. V. Lunel, and H. O. Walther, *Delay Equations. Functional, Complex, and Nonlinear Analysis*. New York, NY, USA: Am. Math. Soc., 1995.
[19] D. Breda, S. Maset, and R. Vermiglio, *Stability of Linear Delay Differential Equations: A Numerical Approach With MATLAB*. New York, NY, USA: Springer, 2015.
[20] K. Engelborghs and D. Roose, "Numerical computation of stability and detection of Hopf bifurcations of steady-state solutions of delay differential equations," *Adv. Comput. Math.*, vol. 10, nos. 3/4, pp. 271–289, May 1999.

[21] K. Engelborghs and D. Roose, "On stability of LMS methods and characteristic roots of delay differential equations," *SIAM J. Numer. Anal.*, vol. 40, no. 2, pp. 629–650, Aug. 2002.
[22] E. Jarlebring, "The spectrum of delay-differential equations: Numerical methods, stability and perturbation," PhD dissertation, TU Braunschweig, Braunschweig, Germany, 2008.
[23] D. Breda, "Solution operator approximations for characteristic roots of delay differential equations," *Appl. Numer. Math.*, vol. 56, no. 3/4, pp. 305–317, Mar./Apr. 2006.
[24] E. Hairer and G. Wanner, *Solving Ordinary Differential Equations II: Stiff and Differential-Algebraic Problems*. Springer Series in Computational Mathematics 14, 2nd ed. BerlinGermany: Springer-Verlag, 1996.
[25] C. Canuto, V. Simoncini, and M. Verani, "On the decay of the inverse of matrices that are sum of Kronecker products," *Linear Algebra Appl.*, vol. 452, no. 1, pp. 21–39, Jul. 2014.
[26] G. Angelidis and A. Semlyen, "Improved methodologies for the calculation of critical eigenvalues in small signal stability analysis," *IEEE Trans. Power Syst.*, vol. 11, no. 3, pp. 1209–1217, Aug. 1996.
[27] R. Barrett, *et al*., *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods*. Philadelphia, PA, USA: SIAM, 1994.
[28] G. Rogers, *Power System Oscillations*. New York, NY, USA: Kluwer Academic Publishers, 2000.

Hua Ye (M'13) received the B.Eng. and Ph.D. degrees in electrical engineering from Shandong University, Ji'nan, China, in 2003 and 2009, respectively.

He is currently a Professor with the Key Laboratory of Power System Intelligent Dispatch and Control, Ministry of Education, Shandong University, Ji'nan, China. His research interests include power system dynamic stability analysis and control, and cyber-physical systems.

Qianying Mou (S'16) received the B. Eng. degree in electrical engineering from Hu'nan University, Changsha, China, in 2014. She is currently working toward the Ph.D. degree in electrical engineering with Shandong University, Ji'nan, China. Her research interests include power system stability analysis and control.



Xinlei Wang received the B.Eng. degree in electrical engineering from China University of Petroleum, Qingdao, China, in 2014. She is currently working toward the M.S. degree in electrical engineering with Shandong University, Ji'nan, China. Her research interests include power system stability analysis and control.

Yutian Liu (SM'96) received the B.Eng. and M.S. degrees in electrical engineering from Shandong University of Technology, Ji'nan, China, in 1984 and 1990, respectively, and the Ph.D. degree in electrical engineering from Xi'an Jiaotong University, Xi'an, China, in 1994.

He is currently a Professor with the Key Laboratory of Power System Intelligent Dispatch and Control of Ministry of Education, Shandong University, Ji'nan, China. His research interests include power system analysis and control and artificial intelligence applications in power system.