

PI-Controlled Variable Time-Step Power System Simulation Using an Adaptive Order Differential Transformation Method

Kaiyang Huang, *Student Member, IEEE*, Yang Liu, *Member, IEEE*, Kai Sun, *Fellow, IEEE*, Feng Qiu, *Senior Member, IEEE*.

Abstract—Dynamic simulation plays a crucial role in power system transient stability analysis, but traditional numerical integration-based methods are time-consuming due to the small time step sizes. Other semi-analytical solution methods, such as the Differential Transformation method, often struggle to select proper orders and steps, leading to slow performance and numerical instability. To address these challenges, this paper proposes a novel adaptive dynamic simulation approach for power system transient stability analysis. The approach adds feedback control and optimization to selecting the step and order, utilizing the Differential Transformation method and a proportional-integral control strategy to control truncation errors. Order selection is formulated as an optimization problem resulting in a variable-step-optimal-order method that achieves significantly larger time step sizes without violating numerical stability. It is applied to three systems: the IEEE 9-bus, 3-generator system, IEEE 39-bus, 10-generator system, and a Polish 2383-bus, 327-generator system, promising computational efficiency and numerical robustness for large-scale power system is demonstrated in comprehensive case studies.

Index Terms—Differential transformation, PI control, variable step-size and order, time domain simulation.

I. INTRODUCTION

MODERN Power systems are characterized by their vast scale and intricate composition, including transmission lines, transformers, generators, loads, and controllers. This complex structure, combined with the diverse and dynamic behavior of the system's elements, can result in a wide range of security issues. In contrast to direct stability analysis such as direct methods [1]–[4], which are based on non-linear system theories such as Lyapunov stability analysis, time-domain simulation-based methods [5]–[9] are able to provide more detailed and accurate dynamics of a power system for its dynamic security assessment (DSA). This flexibility allows for a more comprehensive and accurate assessment of system stability, compared to other methods, and enables the identification of potential issues that may be overlooked by other approaches [3], [4] which are critical for power system operators and engineers in making informed decisions to ensure

the system's safe and reliable operation. However, while time-domain simulation can provide detailed and accurate dynamics, it is time-consuming due to the large-scale power systems models compared with direct methods. Time-domain simulation involves solving the system's mathematical model described by Ordinary Differential Equations (ODEs) or Differential Algebraic Equations (DAEs) [10] as an Initial Value Problem (IVP). However, traditional numerical integration schemes used in commercial software, such as Runge-Kutta (RK) methods [11] for ODEs, or Runge-Kutta-Newton-Raphson methods [10] for DAEs often suffer from computational burdens or stability issues [12]–[14]. Therefore, advanced time-domain simulation techniques are required to maintain stability in modern power systems. Therefore, developing advanced time domain simulation techniques is crucial for ensuring the stability and reliability of modern power systems.

Over the years, significant advancements and improvements have been achieved in time domain simulation to enhance their accuracy, efficiency, and stability. Various advanced simulation techniques have been developed such as simulations based on parallel approaches [8], [15], high-performance computing [16], machine learning [9], [17] and Semi-Analytical Solution (SAS) methods [18]–[21], to provide faster and more accurate simulation results. SAS methods are promising simulation technologies that have recently gained attention. These methods involve dividing the computational burden of solving power system differential equations or algebraic equations into two stages which combine numerical and analytical tools to solve the system's equations, resulting in a more efficient and accurate simulation. In the precomputing stage, an approximate and analytical solution is derived for the power system differential or algebraic equations model based on different spectrums. This solution is an explicit or implicit expression of state variables, the initial state, and parameters and is accurate in a specific time interval, the length of which depends on the system model and the order of the SAS expression. In the online stage, the SAS is evaluated by solving a precomputing model, which usually transforms IVP to algebraic equations with different orders. This process is repeated until the desired simulation length is reached. A variety of SAS methods have been investigated in the literature, such as the Adomian decomposition method [22], the power series method [18], the holomorphic embedding method [23], the differential transformation method [19]–[21], etc. These methods have shown promising results in power system simulations and have been tested and validated on

This work was supported by the Advanced Grid Modeling program of U.S. DOE Office of Electricity under grant DE-OE0000875.

K. Huang and K. Sun are with the Department of Electrical Engineering and Computer Science, University of Tennessee, Knoxville, TN 37996 USA (e-mail: khuang12@vols.utk.edu; kaisun@utk.edu).

Y. Liu and F. Qiu are with Argonne National Laboratory, Lemont, IL 60439 (e-mail: yang.liu100@anl.gov; fqiu@anl.gov).

large-scale power systems [24]. Several examinations [19], [24] show that the Differential Transformation (DT) based time domain simulation has a better speed performance than some RK methods when high accuracy is required for the power system simulation.

Basically, the order and step size are determined case by case and fixed based on experiences. From [19], the DT method is able to increase the step size by using a high-order SAS. And, as a semi-analytical simulation method, the DT method is able to adjust its step size more easily on the fly of the simulation than numerical methods such as RK methods. However, the theoretical linkage between the step size and the order of the SAS, or in other words the optimal pair of step size and the order, has not been studied well. To ensure numerical stability, a step size much smaller than its threshold is often employed in simulations. In fact, how to optimally adjust both the step size and order on the fly of the simulation is not a trivial problem. Variable step and variable order strategies should not be too complex to ensure the efficiency of the simulation, and also the accuracy of the simulation should be acceptable. These requirements will be met by the proposed strategies. One advantage of using the DT method is that the recursive computation process is performed offline, resulting in faster online iteration. Moreover, analyzing error estimation is easier when compared to the embedded RK method [11]. The DT algorithm can adjust the step size based on truncated error without the need to embed another order's solver for error prediction. This feature is demonstrated in a simple case in [21], where an adaptive time window is utilized in a multi-energy system. The aforementioned strategy can be extended to a PI-controlled strategy [25], which performs better in stiff systems and reduces rapid changes in step size. Unlike traditional numerical methods, the DT method can smoothly adjust orders during the simulation, allowing for the design of an adaptive order that works in conjunction with the step strategy to achieve improved performance.

The contributions of this paper are summarized as follows. First, an automatic PI-controlled variable step strategy for the DT method, referred to as the Variable-Step DT (VS-DT) method, is proposed for power system simulations using a fixed order SAS to achieve high-speed performance with good accuracy, even for large systems. Second, an optimally variable order strategy is proposed to enable using the optimal order in the VS-DT method, which is referred to as the Variable-Step-Optimal-Order DT (VSOO-DT) method. By formulating and solving an optimization problem, the optimal SAS order is determined efficiently for the VS-DT method. The variable step and variable order strategies together provide an optimal candidate pair of the step size and order at each time step on the fly of the simulation, ensuring a good balance between numerical stability and simulation speed. Furthermore, the numerical stability of the proposed methods as well as the selections of all parameters are also theoretically analyzed. Finally, by case studies, the two proposed methods are examined through a variety of tests. The test results show, in particular, that compared to similar methods, the proposed VS-DT and VSOO-DT methods achieve more robust and efficient simulations by automatically varying the step size or optimizing

the order as well. Moreover, the proposed two strategies suggest a general framework for simulations using adaptive step sizes and optimized solution orders, not limited to the DT method. Such a framework can be easily generalized to other semi-analytical simulation methods or numerical methods whose orders are adjustable.

The structure of this paper is as follows. In section II, the DT method is reformulated. In Section III we propose a novel PI-controlled variable step size and variable order DT algorithm and analyze the numerical stability. In Section IV we formulate the changing of orders as an optimization problem and a heuristic method is applied to relax the original problem. Finally, in section V, results of time domain simulation based on the new algorithm for several IEEE classical systems are presented to show the scalability and efficiency of the proposed method. We conclude in Section VI by summarizing several advantages of the new strategy and many promising ways to enhance the approach. The detailed power system model is provided in the Appendix.

II. DIFFERENTIAL TRANSFORMATION METHOD

Consider a dynamic system which is Lipschitz on \mathbb{R}^{m_1} :

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}), \quad (1)$$

where \mathbf{x} is in \mathbb{R}^{m_1} . Assuming that the solution $\phi(t, \mathbf{x}_0)$ is sufficiently smooth in a neighborhood $\mathcal{B}_\delta(t_n, \mathbf{x}_0)$, we can expand it using the Taylor series up to a certain order K within $\mathcal{B}_\delta(t_n, \mathbf{x}_0)$:

$$\phi(t, \mathbf{x}_0) = \sum_{k=0}^K \frac{1}{k!} \frac{\partial^k \phi(t, \mathbf{x}_0)}{\partial t^k} \Big|_{t=t_n} (t - t_n)^k + \mathbf{R}(\zeta)(t - t_n)^{K+1}, \quad (2)$$

where $\mathbf{R}(\zeta)$ is related to order K , t_n and t . Substituting (2) into (1), and leveraging the property of the linear independence of polynomials, we can equate the coefficients of the same order, resulting in:

$$(k+1)\mathbf{X}(k+1) = \frac{d\mathbf{f}}{d\mathbf{x}}\mathbf{X}(k) + \mathbf{F}_{\text{non}}(k), \quad (3)$$

where

$$\mathbf{X}(k) = \frac{1}{k!} \frac{\partial^k \phi(t, \mathbf{x}_0)}{\partial t^k} \Big|_{t=t_n}$$

and $\mathbf{F}_{\text{non}}(k)$ is a nonlinear function from $k = 0, \dots, K$. A recursive rule to obtain a semi-analytical solution in a single step is defined in (3). However, when the system dimension becomes large, solving $\mathbf{F}_{\text{non}}(k)$ directly can be computationally complicated due to the high-order partial derivatives. Therefore, further modification is required. To lift the original system into a linear form, we define intermediate variables $\mathbf{y} \in \mathbb{R}^{m_2}$ to eliminate $\mathbf{F}_{\text{non}}(k)$ in a linear form. Suppose $\mathbf{y} = \mathbf{g}(\mathbf{x})$, then, let $\mathbf{z} = (\mathbf{x}^\top, \mathbf{y}^\top)^\top \in \mathbb{R}^{m_1+m_2}$, we can rewrite \mathbf{f} as an augmented system $\mathbf{f}_{\text{aug}}: \mathbb{R}^{m_1+m_2} \rightarrow \mathbb{R}^{m_1}$ such that $\mathbf{f}_{\text{aug}}(\mathbf{x}, \mathbf{g}(\mathbf{x})) = \mathbf{f}(\mathbf{x}) = \mathbf{f}_{\text{aug}}(\mathbf{z})$, $\forall \mathbf{x} \in \mathbb{R}^{m_1}$, by proper selecting intermediate variables, $\mathbf{f}_{\text{aug}}(\mathbf{z})$ is an vector linear function with respect to $\mathbf{z} = (\mathbf{x}^\top, \mathbf{y}^\top)^\top$. Specifically, we assume that \mathbf{y} is a vector function of \mathbf{x} , and we can derive a DT rule offline for the

nonlinear algebraic function $\mathbf{y} = \mathbf{g}(\mathbf{x})$. Using the chain rule of derivatives, we can create more than one layer of variables from \mathbf{y} to \mathbf{x} . The iteration rule can be obtained from the Taylor series of (2) by both sides due to the linear independence of polynomials:

$$\mathbf{Y}(k) = \mathbf{G}(\mathbf{X}(k), \mathbf{X}(k-1), \dots, \mathbf{X}(0)), \quad (4)$$

where \mathbf{G} is a DT function from $(\mathbf{X}(k), \mathbf{X}(k-1), \dots, \mathbf{X}(0))$ to $\mathbf{Y}(k)$ and

$$\mathbf{Y}(k) = \left. \frac{1}{k!} \frac{d^k \mathbf{y}(t)}{dt^k} \right|_{t=t_n}.$$

Based above, a closed form of (3) can be obtained:

$$\begin{aligned} (k+1)\mathbf{X}(k+1) &= \frac{d\mathbf{f}_{\text{aug}}}{d\mathbf{z}} \mathbf{Z}(k) = \mathbf{J}\mathbf{Z}(k), \\ \mathbf{Y}(k) &= \mathbf{G}(\mathbf{X}(k), \mathbf{X}(k-1), \dots, \mathbf{X}(0)), \\ \mathbf{Z}(k) &= (\mathbf{X}(k), \mathbf{Y}(k)) \end{aligned} \quad (5)$$

from $k = 0, \dots, K-1$. The matrix \mathbf{J} remains a constant matrix, while the higher-order nonlinear terms $\mathbf{F}_{\text{non}}(k)$ have been eliminated and transformed into a more manageable derived vector function \mathbf{G} with expanded dimensions. In [19], various forms of nonlinear functions, such as square functions and sine functions, have been derived. Moreover, the DT iteration formula for the detailed and classical models (39) for power system dynamic simulations has been proposed in [19], [20]. This iteration process takes the form of (5), providing a K^{th} order approximation solution of ϕ in $\mathcal{B}_\delta(t_n, \mathbf{x}_0)$. Finally, an algorithm can be designed to solve the IVP (1):

Algorithm 1 Original DT Method

Require: $t_0, t_e, h, K, \mathbf{x}_0$.

- 1: initialization $t = t_0, \mathbf{X}(0) = \mathbf{x}_0$,
 - 2: $\mathbf{Y}(0) = \mathbf{G}(\mathbf{X}(0)), \mathbf{Z}(0) = (\mathbf{X}(0), \mathbf{Y}(0)), n = 0$.
 - 3: **while** $t + h < t_e$ **do**
 - 4: **while** $k < K$ **do**
 - 5: $\mathbf{Y}(k) \leftarrow \mathbf{G}(\mathbf{X}(k), \mathbf{X}(k-1) \dots \mathbf{X}(0))$
 - 6: $\mathbf{Z}(k) \leftarrow (\mathbf{X}(k), \mathbf{Y}(k))$
 - 7: $\mathbf{X}(k+1) \leftarrow (k+1)^{-1} \mathbf{J}\mathbf{Z}(k)$
 - 8: $k \leftarrow k + 1$
 - 9: **end while**
 - 10: $\mathbf{x}(t_{n+1}) \leftarrow \sum_{k=0}^K \mathbf{X}(k)h^k$
 - 11: $\mathbf{X}(0) \leftarrow \mathbf{x}(t_{n+1})$.
 - 12: $t_{n+1} \leftarrow t + h, t \leftarrow t + h, n \leftarrow n + 1, k \leftarrow 0$
 - 13: **end while**
 - 14: **return** $\mathbf{x}(t), t_0 \leq t \leq t_e$
-

Remark 1. Algorithm 1 provides a general way to implement the DT method into dynamic simulations on an ODE model. For a more general DAE model, a similar process can be implemented [20]. In such cases, different from iterative numerical methods, a DT-based dynamic simulation is conducted in a recursive manner using a high-order power series, i.e., the SAS, in which higher-order mathematical terms can be recursively derived from low-order terms. Thus, a high-order approximation of the true solution is achieved, compared with traditional low-order numerical methods, e.g., the RK4 method.

Furthermore, there are two main approaches to solving the DAE model: simultaneous and partitioned methods. Simultaneous methods handle both differential and algebraic equations together, while partitioned methods alternate their solutions. While the partitioned method is widely used, solutions based on partitioned methods can face numerical issues due to the interfacing of different variables and it is difficult to use large time step sizes. In contrast, benefiting from the non-iterative and recursive nature of the SAS, the numerical error induced by the partitioning [14] can be avoided by the DT method.

Remark 2. While higher order accuracy and numerical stability can be provided by the DT method on a power system modeled by DAEs or ODEs, the resulting solution of (1) is supposed to be continuously differentiable in a desired order for the existence and convergence of the power-series approximation. In general, this assumption can easily be met considering the smooth response of state variables in a power system model even with switchable controllers considering control criteria or limits. The vector field $\mathbf{f}(\mathbf{x})$ of the power system model is at least piecewise smooth, so the DT algorithm is still well-defined and the corresponding SAS can be derived for each smooth sub-region of the vector field in the power series form of $\mathbf{x}(t_{n+1}) = \sum_{k=0}^K \mathbf{X}(k)h^k$. Then, the SAS is evaluated over consecutive time intervals of h until making up the desired simulation period. When a switch is detected with any controller during the simulation, the timing can be identified to decrease the step size around the switching point if the resulting error exceeds a threshold based on defect control [26]. Therefore, switches with controllers do not have a significant influence on the time performance of the DT method.

III. PI CONTROL-BASED STEP STRATEGY

A. A typical step change algorithm

In power system simulation, achieving high speed is crucial as online screening requires faster-than-real-time simulation to provide fault information in time, enabling strategies to be implemented promptly. Additionally, simulating a large power system, both in the time domain and for cascading failures, can be exceedingly time-consuming, with no promising solutions yet for dealing with such high-dimensional problems. Therefore, balancing solver accuracy and speed is always a key issue in power system research. Adopting a proper step control algorithm for the DT method could be significant for speeding up simulations. One famous step strategy for RK methods addresses step issues by estimating the error in every step and controlling the step size with a given tolerance value. When a K^{th} order solution with the DT method is used, the absolute local truncation error of $[\mathbf{x}]_i$ can be estimated using the last term $[\mathbf{X}]_i(K)h^K$. Note that for a power system model, state variables have different physical meanings and units. Therefore, errors are compared across different variables, and they need to be normalized. Unlike the existing work [11], [21], the error index with order K is estimated as follows. Suppose that the approximate solution converges and satisfies the root criteria [27], i.e., there exist a radius \bar{r} and \bar{K} determined by h such

that $(\|\mathbf{X}(K)\|_\infty h^K)^{1/K} \leq \bar{r} < 1$ for $\forall K \geq \bar{K}$. Equivalently, if $\|\mathbf{x}\|_\infty$ is bounded, there exist radius r and \hat{K} such that

$$r(k) := \left(\max_{1 \leq i \leq m_1} \left| \frac{[\mathbf{X}]_i(K) h^K}{\|\mathbf{x}\|_i + \eta_i} \right| \right)^{1/K} \leq r \quad (6)$$

for $\forall K \geq \hat{K}$. Here, η_i is a small constant introduced to scale the i^{th} component of \mathbf{x} . This is to avoid the singularity caused by $[\mathbf{x}]_i = 0$. It also helps adjust the trade-off between absolute and normalized errors. Note that the left-hand side of (6) represents the proposed comprehensive error index at order K . Once this error at a particular order is determined, the radius r can be approximated using the left-hand side of (6). Then, truncation errors related to higher orders can be bounded using geometric sequences $\sum_{l=K+1}^{\infty} r^l$ based on the value of r , as per our assumption. Consequently, the local truncation error can be estimated as follows:

$$\begin{aligned} \mathcal{E}(K, h) &:= \sum_{l=K+1}^{\infty} \left(\max_{1 \leq i \leq m_1} \left| \frac{[\mathbf{X}]_i(l) h^l}{\|\mathbf{x}\|_i + \eta_i} \right| \right) \\ &\approx \sum_{l=K+1}^{\infty} r^l = \frac{r^{K+1}}{1-r}. \end{aligned} \quad (7)$$

In comparison to truncation error estimation based on a single term, (7) gives a more conservative error estimation considering higher order terms. By setting proper tolerance Tol for per step size error, a control strategy [28] can be

$$\begin{aligned} e_n &= \frac{\mathcal{E}(K, h_n)}{h_n}, \\ \theta_{n+1} &= \min \left\{ \gamma \left(\frac{Tol}{e_n} \right)^{K-1}, \theta_{\max} \right\}, \\ h_{n+1} &= \theta_{n+1} h_n, \end{aligned} \quad (8)$$

where n represents the step stage, from now we simplify t_n as subscript n , θ is a dynamic parameter adjusted at each step to determine the next step size, balancing the need for accuracy (keeping e_n small) and efficiency (allowing for larger step sizes h_n when possible). The scheme introduced above provides a simple method for determining a step size that is as large as possible without exceeding the error tolerance, i.e., ensuring that $e_n < Tol$. To reduce the likelihood of a rapid increase in error, a safety factor $\gamma < 1$ can be used. This approach has been shown to be quite effective for general problems and can even expand the stability region [28]. However, it is prone to step oscillations when the system becomes stiff, making the scheme unstable. We will show later that the algorithm above can be considered an integral controller, and that a PI controller typically outperforms it.

B. A PI Control Based Step Strategy

In this subsection, a DT scheme based on PI control is applied to accelerate time domain simulations. For simplicity, ignore the practical setting of (8), i.e. the maximum gain constraint θ_{\max} , and take its logarithm to obtain:

$$\log h_{n+1} - \log h_n = \frac{1}{K} (\log \gamma^K Tol - \log e_n), \quad (9)$$

which represents a discrete integral loop. The left-hand side is the difference form of the output derivative with t replaced by n , while the right-hand side is the difference between the input $\log \gamma^K Tol$ and the feedback variable $\log e_n$. From a control perspective, a PI controller can perform better than a single I controller. Therefore, we consider a more general and flexible framework in this paper, and the stability and robustness of this approach have been proven in [25], [29], adding a proportional loop to (9), yields:

$$\log h_{n+1} - \log h_n = K_I d_n + K_P (d_n - d_{n-1}), \quad (10)$$

where $d_n = (\log \gamma^K Tol - \log e_n)$ and K_I, K_P are parameters for the integral loop and proportional loop. Rearranging (10), the explicit control strategy can be

$$h_{n+1} = h_n \left(\frac{\gamma^K Tol}{e_n} \right)^{K_I} \left(\frac{e_{n-1}}{e_n} \right)^{K_P}, \quad (11)$$

It turns out that the typical step change algorithm is a specific case of the PI algorithm with $K_I + K_P = 1/K$ and $K_P = 0$.

C. Stability of the PI Controller and Selection of Parameters

The stability of the PI controller is analyzed in this section. Consider the Dahlquist test equation $\dot{x} = \lambda x$, $\lambda \in \mathbb{C}$, a standard system for analyzing numerical stability. By adopting the DT method and PI-controlled variable step strategy, the corresponding discrete equations become:

$$x_{n+1} = D(z)x_n \quad (12)$$

$$e_n = |E(z)x_n| \quad (13)$$

$$h_{n+1} = h_n \left(\frac{Tol}{e_n} \right)^{K_I} \left(\frac{e_{n-1}}{e_n} \right)^{K_P} \quad (14)$$

where $z := \lambda h_n$, $D(z) = \sum_{p=0}^K z^p/p!$, and $E(z) = z^K/K!$. Notice that different from the definition of e_n in (8), the error is analyzed in a general form, i.e., the absolute error, which does not influence the stability analysis. Difference equations (12) and (13) discretize $\dot{x} = \lambda x$ based on the DT method considering K^{th} order accuracy. In (13), the local truncation error is estimated by the last term in the power series, and (14) is rearranged from (10). For the purpose of convergence analysis, let $\alpha = K_I + K_P$, $\beta = K_P$, $\varphi_n = \log |x_n|$, and $\chi_n = \log h_n$, the system above (12)-(14) becomes:

$$\begin{aligned} \varphi_{n+1} &= \log |D(\lambda e^{x_n})| + \varphi_n \\ \chi_{n+1} &= \chi_n - \alpha \log |E(\lambda e^{x_n})| - \alpha \varphi_n \\ &\quad + \beta \log |E(\lambda e^{x_{n-1}})| + \beta \varphi_{n-1} + \gamma_1 \end{aligned}$$

where γ_1 denotes some constant, the stability of the system can be analyzed from the radius of the Jacobian matrix, i.e., the transformed matrix of the map: $(\varphi_n, \chi_n, \varphi_{n-1}, \chi_{n-1}) \xrightarrow{\text{DT with PI}} (\varphi_{n+1}, \chi_{n+1}, \varphi_n, \chi_n)$:

$$\frac{\partial(\varphi_{n+1}, \chi_{n+1}, \varphi_n, \chi_n)}{\partial(\varphi_n, \chi_n, \varphi_{n-1}, \chi_{n-1})} = \begin{pmatrix} 1 & u & 0 & 0 \\ -\alpha & 1 - \alpha v & \beta & \beta v \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \quad (15)$$

where u and v are defined by

$$u = \Re \left(\frac{dD(z)}{dz} \cdot \frac{z}{D(z)} \right), v = \Re \left(\frac{dE(z)}{dz} \cdot \frac{z}{E(z)} \right)$$

where $\Re(\cdot)$ means the real part of a variable. The Jacobian matrix can be evaluated at the equilibrium point of the mapping, i.e., at (φ_n, χ_n) such that $|D(\lambda e^{X_n})| = 1$, and φ_n can be determined from the second equation. By choosing proper values for α and β , the PI control algorithm (14) expands the stability boundary and increases the simulation speed. Some cases for RK methods are demonstrated in [29]. It is known that the DT method shares the same stability domain with the RK method. Thus, the PI control strategy can also expand the stability region of the DT method. However, how to select the proper parameters K_P and K_I of the PI controller, is an interesting problem. Different from the RK method, the DT method can easily estimate its error by plugging $E(z) = z^N/N!$ into (10) to obtain the linear difference equation if h_n is small:

$$\log h_{n+1} - (K\alpha - 1) \log h_n - N\beta \log h_{n-1} = \gamma_2 \quad (16)$$

where

$$\gamma_2 = (\alpha - \beta) \log \left(\frac{K! Tol}{\lambda^K} \right) \quad (17)$$

The corresponding characteristic equation is

$$\lambda^2 - (K\alpha - 1)\lambda - K\beta = 0. \quad (18)$$

It turns out that a typical variable step algorithm is a special case of the proposed PI control algorithm when $\alpha = 1/K$ and $\beta = 0$. If e_n in (8) for each step is estimated using the last term, a slight modification to (18) is required

$$\lambda^2 - (K\alpha - 1 - \alpha)\lambda - (K - 1)\beta = 0. \quad (19)$$

The selection of α and β , or in other words K_I and K_P of the PI controller, should make that the modulus of solutions of (18) or (19) < 1 to accelerate the simulation. Also, the radius of the Jacobian matrix in (15) needs to be < 1 to expand the stability domain. The setting in [30] is followed, which is verified by both control theory and computational case studies. Thus, let $K_I = 0.3/K$ and $K_P = 0.4/K$. K_I and K_P can be calculated once the order of the DT method is determined.

IV. OPTIMAL ORDER SELECTION

A. The Complexity of Power System Simulation Using DT

Previous studies [31], [32] focused on selecting the optimal order for the Taylor series, while in this work, we propose a dynamic order selection approach based on the PI and typical step change algorithms. The complexity of the DT method is quantified as the number of multiplication operations performed in each time step. For a classical power system model without controllers, the computational complexity is expressed as:

$$C(K) = C_{ca}(K) = N_{ca1}K^2 + N_{ca2}K + N_{ca3}, \quad (20)$$

where $N_{ca1} = 2N_g$, $N_{ca2} = 2N_g + 4N_g^2 + N_xN_z + N_x$, and $N_{ca3} = N_x$, N_g denotes the number of generators, $N_x = m_1$ is the dimension of the original system (1), and $N_z = m_1 + m_2$ is the dimension of the augment variable \mathbf{z} . The computation of

complexity can be divided into three steps. First, we count the multiplication operations involved in calculating the augment variables $Y(k)$ up to K in (4). Second, we calculate the multiplication operations required for computing $X(k)$ up to order K . Last, we count the multiplication operations needed for calculating from $X(k)$ to \mathbf{x}_{n+1} .

The same procedure can apply to other power system models. For instance, for a detailed model given by (39a) to (40), a complexity function is given by:

$$C(K) = C_{de}(K) = N_{de1}K^2 + N_{de2}K + N_{de3}, \quad (21)$$

where $N_{de1} = 17N_g/2$, $N_{de2} = 4N_g^2 + 27N_g/2 + N_xN_z + N_x$, and $N_{de3} = N_x$. With a formula like (20) or (21) in mind, we can design a special algorithm for optimal order switching based on different power system models. The process of computing complexity is based on the steps outlined in lines 5 to 10 of Algorithm 1.

B. Algorithm of the Adaptive Order Strategy

From now we use K_{n+1} to represent the order of the DT method used in the step $n + 1$. To select the optimal order, we formulate an optimization problem based on the complexity of the simulation model. Specifically, the goal is to maximize the step size h_{n+1} while minimizing the complexity $C(K_{n+1})$, without significantly affecting the accuracy of the simulation. We model the problem as follows:

$$\max \frac{h_{n+1}}{C(K_{n+1})} \text{ s.t. } (8) \text{ or } (11). \quad (22)$$

Here, note that to strike a balance between stability and speed, as we discussed in the previous section, the PI controller parameters, namely K_I and K_P , are dependent on the order of the DT algorithm, and the error estimation is also linked to the choice of order. Given the fact that this optimization problem cannot be solved explicitly, we propose a heuristic approach to separate this problem into three situations:

Situation 1. If the step size has increased or the order has decreased in the previous step, which basically indicates that the error is lower compared to the tolerance, then the order is likely to decrease further in this step to accelerate the simulation. The corresponding condition is:

$$h_{n-1} \leq h_n \text{ or } K_n \leq K_{n-1}, \quad (23)$$

if the (23) holds, the problem is transformed into checking whether the inequality

$$\frac{h_{de}}{C(K_{de})} > \mu_{de} \frac{h_{es}}{C(K_n)} \quad (24)$$

holds, where

$$K_{de} := K_n - \delta K, \quad (25)$$

$$r_{de} := r(K_{de}, h_n) = \left(\max_{1 \leq i \leq m_1} \left| \frac{[\mathbf{X}]_i(K_{de})h_n^{K_{de}}}{|[\mathbf{x}]_i| + \eta_i} \right| \right)^{1/K_{de}}, \quad (26)$$

$$e_{de} := \frac{\mathcal{E}(K_{de}, h_n)}{h_n} = \frac{r_{de}^{K_{de}+1}}{(1 - r_{de})h_n}, \quad (27)$$

$$\theta_{de} := \min \left\{ \gamma \left(\frac{Tol}{e_{de}} \right)^{1/K_{de}}, \theta_{\max} \right\}, \quad (28)$$

$$h_{de} := h_{n+1, K_{de}} = \max \{ \theta_{de} h_n, h_{\min} \}. \quad (29)$$

μ_{de} controls the switching threshold, δK is the order intended to decrease in the next step, and h_{de} is the step size candidate obtained from (8) with order $K_n - \delta K$. If $\mu_{de} > 1$ and the operation point (h_{de}, K_{de}) performs better than (h_n, K_n) , the order and step are switched which means a reduced order can be used if the same accuracy can be guaranteed to speed up the simulation. The step size h_{es} is estimated by (8) and controlled by a PI controller (11). If (24) does not hold, h_{n+1} is estimated using the PI controller and $K_{n+1} = K_n$ is fixed. θ_{\max} and h_{\min} are the limits for θ_{de} and h_{de} , respectively.

Situation 2. If the step size has decreased or the order has increased in the previous step, which indicates that the error is larger compared with tolerance, then the order tends to increase in this step to mitigate the error. The condition is

$$h_n \leq h_{n-1} \text{ or } K_{n-1} \leq K_n. \quad (30)$$

Its satisfaction transforms the problem to checking whether this inequality holds:

$$\frac{h_{in}}{C(K_{in})} > \mu_{in} \frac{h_{es}}{C(K_n)} \quad (31)$$

where

$$K_{in} := K_n + \Delta K, \quad (32)$$

$$\rho_{in} := \min$$

$$\left\{ \max_{1 \leq i \leq m_1} \left| \frac{[\mathbf{X}]_i(K_n - 1)}{[\mathbf{X}]_i(K_n)} \right|, \max_{\substack{1 \leq i \leq m_1 \\ 0 \leq j \leq 1}} \left| \frac{[\mathbf{X}]_i(K_n - (j + 2))}{[\mathbf{X}]_i(K_n - j)} \right|^{\frac{1}{2}} \right\}, \quad (33)$$

$$e_{in} := \frac{e_n}{\rho_{in}} = \frac{\mathcal{E}(K_n, h_n)}{(\rho_{in} h_n)^{\Delta K}}, \quad (34)$$

$$\theta_{in} := \min \left\{ \gamma \left(\frac{Tol}{e_{in}} \right)^{K_{in}^{-1}}, \theta_{\max} \right\}, \quad (35)$$

$$h_{in} := h_{n+1, K_{in}} = \min \{ \theta_{in} h_n, h_{\max} \}, \quad (36)$$

μ_{in} is the threshold for switching orders. Different from **situation 1**, the higher order truncation error is implicit as the coefficients' vector $\mathbf{X}(K_n + \Delta K)$ is unknown for a K_n^{th} order DT method. So a convergence rate factor ρ_{in} approximates the local truncation error candidate e_{in} that corresponds to the order K_{in} , and the ratios in (33) involving different orders help approximate how much the solution has changed between orders. By considering different ratios, potential issues caused by even or odd functions can be avoided. Furthermore, taking the minimum of these ratios ensures that the convergence rate factor is conservatively estimated, providing a safety margin for the increase in order. ΔK represents the increased order in the n^{th} step, and h_{in} represents the candidate step size predicted by (8) with order K_{in} . If (31) holds, the operation point switches from (h_n, K_n) to (h_{in}, K_{in}) since a higher order DT method enables a larger time step at step $n + 1$. Set $K_{n+1} = K_n$ and $h_{n+1} = h_{es}$ by (11) if (31) doesn't hold.

Situation 3: If conditions for **situation 1** and **situation 2** are both satisfied, then the order is determined by optimizing the objective function with respect to all three values. For simplicity, assume that $\mu_{in} = \mu_{de} = 1$. If the conditions (23) and (30) hold, the problem is equivalent to solving

$$(h_{n+1}, K_{n+1}) \in \arg \max_{(h, K) \in \mathcal{I}} \frac{h}{C(K)}, \quad (37)$$

such that $\mathcal{I} = \{(h_{de}, K_{de}), (h_{es}, K_n), (h_{in}, K_{in})\}$, note that all elements in \mathcal{I} are defined above in **situation 1** and **situation 2**, i.e., (11), (25) to (29) and (32) to (36). This is the general formulation of the previous situation, where specific situations are taken into consideration when certain conditions are met. Otherwise, the operational point is determined by comparing the three objective functions, i.e., $h/C(K)$. The algorithm for this VSOO-DT approach is outlined below.

Algorithm 2 The Variable-Step-Optimal-Order DT Method

Require: $t_0, t_e, h_0, K_0, \mathbf{x}_0, Tol, h_{\min}, h_{\max}, K_{\min}, K_{\max}, \mu_{de}, \mu_{in}, \delta K, \Delta K, \eta, \gamma, \theta_{\max}$.

- 1: initialization $t = t_0, n = 0, e_0 = 0, \mathbf{X}(0) = \mathbf{x}_0, \mathbf{Y}(0) = \mathbf{G}(\mathbf{X}(0)), \mathbf{Z}(0) = (\mathbf{X}^\top(0), \mathbf{Y}^\top(0))^\top$
- 2: **while** $t + h_n < t_e$ **do**
- 3: **while** $k < K_n$ **do**
- 4: $\mathbf{Y}(k) \leftarrow \mathbf{G}(\mathbf{X}(k), \mathbf{X}(k-1) \cdots \mathbf{X}(0))$
- 5: $\mathbf{Z}(k) \leftarrow (\mathbf{X}(k), \mathbf{Y}(k))$
- 6: $\mathbf{X}(k+1) \leftarrow (k+1)^{-1} \mathbf{JZ}(k)$
- 7: $k \leftarrow k + 1$
- 8: **end while**
- 9: $\mathbf{x}(t_{n+1}) \leftarrow \sum_{k=0}^{K_n} \mathbf{X}(k) h^k$
- 10: **if** $h_{es} \leq h_{\min}$ **then**
- 11: $h_{es} \leftarrow h_{\min}$
- 12: **else if** $h_{es} \geq h_{\max}$ **then**
- 13: $h_{es} \leftarrow h_{\max}$
- 14: **end if**
- 15: calculating step h_{es} using PI control strategy (11)
- 16: **if** (23) and (30) hold and $K_{\min} \leq K_{de} \leq K_{in} \leq K_{\max}$ **then**
- 17: $(h_{n+1}, K_{n+1}) \leftarrow \arg \max_{(h, K) \in \mathcal{I}} \frac{h}{C(K)}$
- 18: **else if** (23), (24) hold and $K_{\min} \leq K_{de}$ **then**
- 19: $(h_{n+1}, K_{n+1}) \leftarrow (h_{de}, K_{de})$ by (25) – (29)
- 20: **else if** (30), (31) hold and $K_{\min} \leq K_{de}$ **then**
- 21: $(h_{n+1}, K_{n+1}) \leftarrow (h_{in}, K_{in})$ by (32) – (36)
- 22: **else**
- 23: $(h_{n+1}, K_{n+1}) \leftarrow (h_{es}, K_n)$ by (11)
- 24: **end if**
- 25: $\mathbf{X}(0) \leftarrow \mathbf{x}_{n+1}$
- 26: $t_{n+1} \leftarrow t + h_n, t \leftarrow t + h_n, n \leftarrow n + 1, k \leftarrow 0$
- 27: **end while**
- 28: **return** $\mathbf{x}(t), t_0 \leq t \leq t_e$

C. Selecting Parameters

With the basic motivations and principles of the PI control strategy and the optimization strategy explained, the selection of each parameter with the strategies is suggested below:

Initial Values: Begin with a small initial step size (h_0) and order (K_0) to have a safe start. Notably, the proposed methods'

robustness allows these values to adapt during the simulation, reducing sensitivity to initial settings.

Error Tolerance (Tol): Select the *Tol* value based on specific simulation requirements, typically ranging from 10^{-25} to 10^{-2} . Higher *Tol* values are primarily used to assess performance but are generally unnecessary. Typically, a range of 10^{-8} to 10^{-2} is sufficient.

Additional Parameters: Parameters such as minimum and maximum step sizes (h_{\min} and h_{\max}) are problem-specific and designed for practical purposes, like achieving desired solution resolutions. Minimum and maximum orders (K_{\min} and K_{\max}) are flexible depending on the problem. Note that high orders are rarely needed unless exceptional accuracy is required.

Thresholds for Order Switching (μ_{de} and μ_{in}): These thresholds relate to the order-switching process within the VSOO-DT method. Values exceeding 1 impose stricter switching criteria, ensuring superior performance in the new state. Notably, μ_{in} should exhibit a normalizedly higher value compared to μ_{de} since K_{in}^{th} order coefficients are estimated by ρ_{in} but K_{de}^{th} order coefficients could be computed accurately.

Order Adjustment Parameters (δK and ΔK): These parameters can be adjusted as needed. This paper lets $\delta K = \Delta K = 1$ to achieve a smooth transition between orders. Alternatively, order switching can occur after several steps with larger δK and ΔK values.

Scaling Factor for Error Estimation (η_i): η_i scales the estimated error. It helps to avoid singularities when certain variables approach zero. Setting $\eta_i = 0$ results in purely normalized error estimation, while $|\eta_i| > 0$ allows a balance between absolute and normalized errors. This point will become clear if setting $\eta_i = -||\mathbf{x}_i|| + 1$. Generally, the (6) can capture the hybrid absolute-normalized error, and one can adjust the ratio between absolute and normalized errors, a small $|\eta_i|$ will lead to a domination of the normalized error which is usually meaningful and reliable for simulating a large-scale power system having state variables with different physical meaning.

Safety Factors and Limits (γ and θ_{max}): γ acts as a safety factor and can be set to 1 in the proposed VSOO-DT method due to the presence of a P controller. θ_{max} limits the maximum step size increment to prevent excessive overshooting, e.g., $\theta_{max} = 2$ means that the largest h_{n+1} will not exceed $2h_n$. In power system simulations, double the step size would be enough to respond to the dynamics of systems without influencing the resolution of solutions.

Values and ranges of these parameters are suggested in Table I for power system simulations. It is worth noting that most of the parameters are adaptable to meet any specific requirements.

V. NUMERICAL RESULTS

In this section, the efficiency, robustness, and validity of the proposed algorithm are evaluated using both classical and detailed models of the IEEE 9-bus system, IEEE 39-bus system, and Polish 2383-bus system [19]. The case study focuses on three stages: (1) verification of the proposed approach with different scenarios in different systems, (2) validation of the robustness with different tolerances and comparison of the proposed approach with different fixed orders using

TABLE I
SETTING OF PARAMETERS

Parameters	Meanings	Values
h_0	initial step-size	0.001
K_0	initial order	8
Tol	error tolerance	$[10^{-25}, 10^{-2}]$
h_{\min}	minimum step-size	depends
h_{\max}	maximum step-size	depends
K_{\min}	minimum order	≥ 4
K_{\max}	maximum order	≤ 45
μ_{de}	switching factor for decreasing order	$[1, 2.5]$
μ_{in}	switching factor for increasing order	$[1, 2.5]$
δK	increased order	≥ 1
ΔK	decreased order	≥ 1
η_i	scaling factor for component $[\mathbf{x}]_i$	$[10^{-19}, 0.1]$
γ	safety factor	$[0.85, 1]$
θ_{max}	maximum incremental gain	$[1.25, 2]$

variable time step algorithm, (3) comparison of the efficiency of the proposed approach with other numerical algorithms. To compare the error of each method, the result from the RK4 method with a tiny time step size $h = 1 \times 10^{-4}$ s is considered the benchmark.

A. Screening of Comprehensive Scenarios

The proposed VSOO-DT method is tested on three systems: IEEE 9-bus, IEEE 39-bus, and Polish 2383-bus systems. Simulations are conducted using Matlab on a laptop computer with an Intel CoreTM i7-6600U CPU and 8 GB RAM. For the IEEE 9-bus system, simulations are conducted with and without the tripping of the branch between buses 5 and 7 after grounding. The parameters used in the simulations are provided in Table II.

TABLE II
DEFAULT SETTING OF PARAMETERS FOR IEEE 9-BUS SYSTEM

h_0	K_0	Tol	h_{\min}	h_{\max}	K_{\min}	K_{\max}
0.001 s	4	10^{-5}	10^{-4} s	0.2 s	4	45
μ_{de}	μ_{in}	δK	ΔK	η	γ	θ_{max}
1	1.02	1	1	10^{-19}	1	2

The simulation results of several variables, including relative rotor angles, absolute error, adaptive step size, and the optimal order during the simulation, are displayed in Fig. 1a-1h. The sequence of error in Fig. 1b is calculated by the infinity norm of the difference between the benchmark $\mathbf{x}_{bh}(t_n)$ and \mathbf{x}_n for every step, i.e., given a time t_n , the error is calculated:

$$\text{Error}(t_n) = \|\mathbf{x}_{bh}(t_n) - \mathbf{x}_n\|_{\infty}. \quad (38)$$

From Fig. 1b, the error is finally stabilized around 4×10^{-6} which means the accuracy of the proposed method can be guaranteed. The step size and order are time-varying and controlled by the PI controller and the optimal order algorithm. By increasing the order during the simulation, a relatively large time step size can be applied without violating the numerical stability as well as with promising accuracy. Note that the step size finally is approaching the upper limit $h_{\max} = 0.2$ s and

the order is also stabilized around 15. For the unstable case, the step size can be stabilized around 0.04 s and the order oscillates around 15, since compared with the stable one, the system is stiffer, thus requiring a more minor time step.

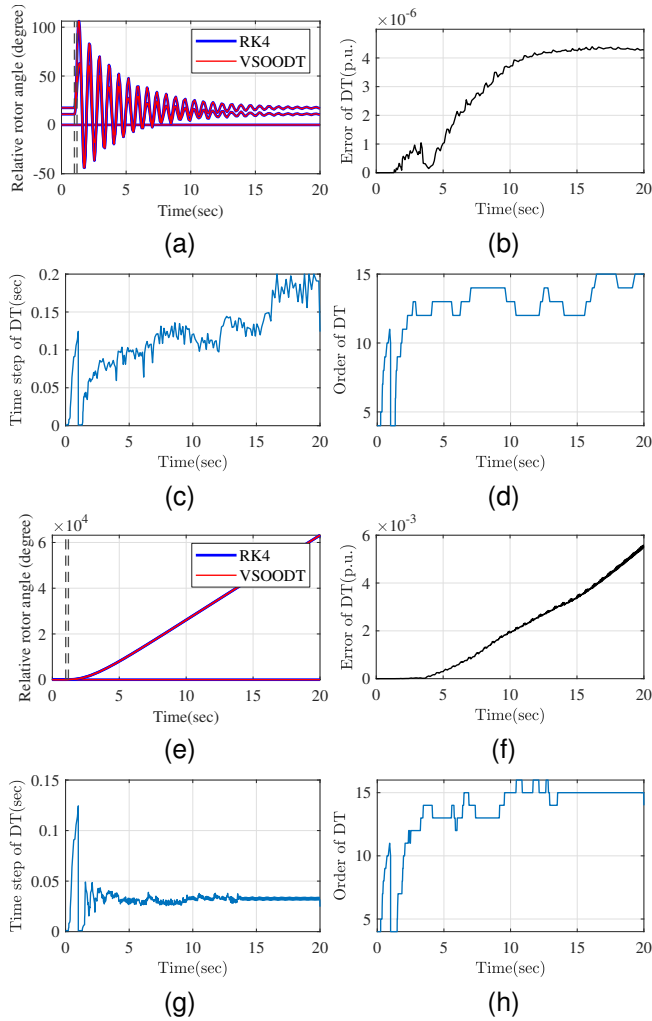


Fig. 1. Simulation results for the IEEE 9-bus system. (a) Relative rotor angles in the stable case. (b) maximum errors in the stable case. (c) Step size during the simulation in the stable case. (d) Optimal order of DT during the simulation in the stable case. (e) Relative rotor angles in the unstable case. (f) maximum errors in the unstable case. (g) Step size during the simulation in the unstable case. (h) Optimal order of DT during the simulation in the unstable case.

For the IEEE 39-bus system, a three-phase bus grounding fault is applied to bus 9 at 1 s and cleared at 1.2 s, resulting in potential rotor angle instability. An unstable case is considered where a three-phase bus grounding is applied to bus 39 at 1 s and cleared at 1.5 s by tripping the generator at bus 39. The same parameters as in Table II are used. Simulation results are displayed in Fig. 2a-2h. In this case, initially **situation 2** is the dominant mode, and the step size increases rapidly with the increasing order. The upper limit of the step size is approached around 10s, and the order also stops to increase as there is no further room to increase the step size for acceleration. **Situation 1 or 3** is activated, and such a high-order model does not need to be maintained. Nonetheless, errors can still be controlled. For the unstable case, the step size is increased and oscillates around 0.1 s, and the order is adjusted to increase. Otherwise, the time step size will start to decrease to influence the time

performance.

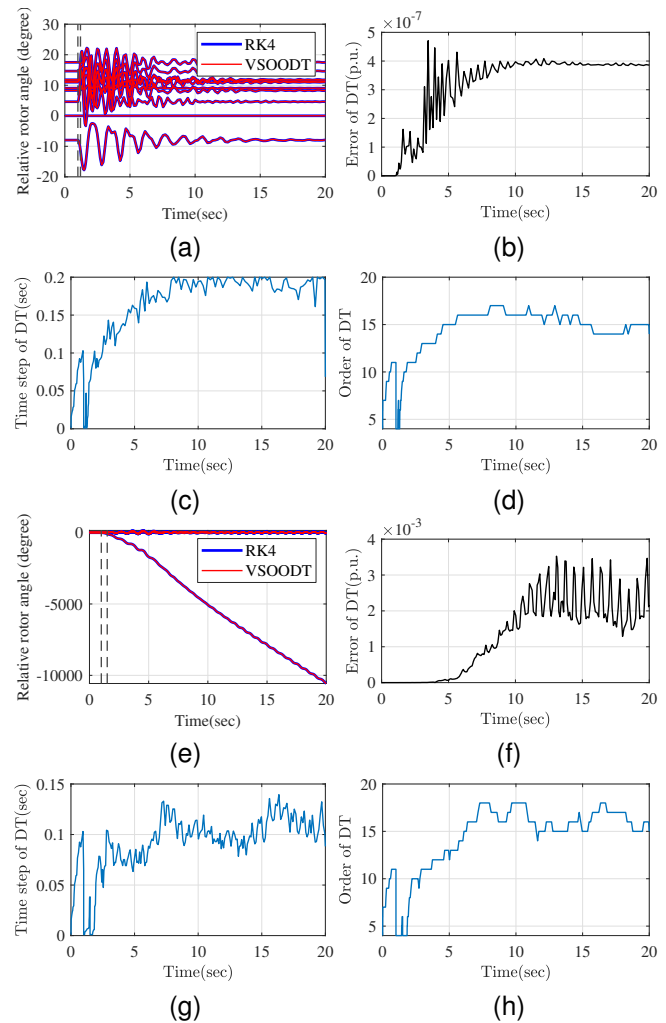


Fig. 2. Simulation results for the IEEE 39-bus system. (a) Relative rotor angles in the stable case. (b) maximum errors in the stable case. (c) Step size during the simulation in the stable case. (d) Optimal order of DT during the simulation in the stable case. (e) Relative rotor angles in the unstable case. (f) maximum errors in the unstable case. (g) Step size during the simulation in the unstable case. (h) Optimal order of DT during the simulation in the unstable case.

Lastly, the Polish 2383-bus system is considered, where a three-phase bus grounding is applied to bus 9 at 1 s and the branch between bus 6 and bus 9 is tripped at 1.2 s, leading to a stable system. An unstable case arises from tripping the branch between bus 6 and bus 9 at 1.5 s. The parameters are shown in Table II for the above two cases except for $K_0 = 8$ and $h_{max} = 0.12s$. Simulation results for $Tol = 10^{-15}$ are displayed in Fig. 3. In the stable case, the step size approaches the upper limit at around 3 s with the order increasing up to 32. The system gradually converges to an equilibrium point with the order ultimately converging to 29. Note that the order will keep decreasing if the simulation continues beyond 20 s since a lower-order model can support the h_n approach to the upper limit. The unstable case has the step size approach the upper limit before one rotor angle diverges after 5 s, and reduces the step size rapidly due to the faster dynamics than the stable case. The order with the unstable case is lower, and increasing the order does not significantly improve the

simulation speed. Overall, the proposed VSOO-DT method demonstrates its robustness in different cases with the desirable numerical stability and enlarged step sizes.

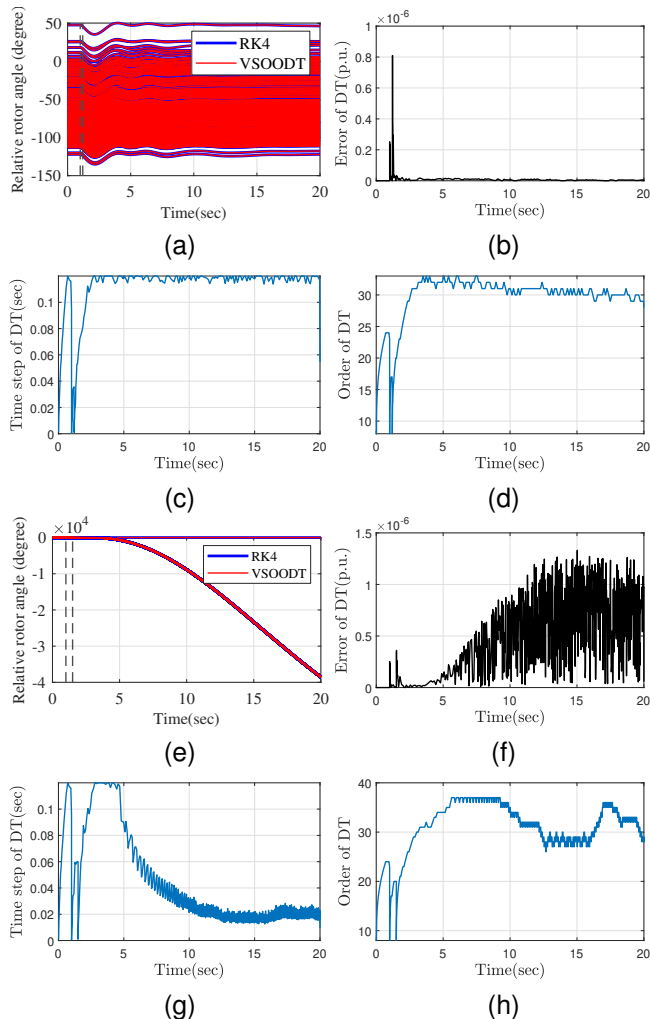


Fig. 3. Simulation results for the Polish 2383-bus system. (a) Relative rotor angles in the stable case. (b) maximum errors in the stable case. (c) Step size during the simulation in the stable case. (d) Optimal order of DT during the simulation in the stable case. (e) Relative rotor angles in the unstable case. (f) maximum errors in the unstable case. (g) Step size during the simulation in the unstable case. (h) Optimal order of DT during the simulation in the unstable case.

B. Comparing with the Variable-Step DT Method with a Fixed Order

To emphasize the importance of selecting the optimal order, the proposed VSOO-DT method is compared with the proposed VS-DT method using different fixed orders. The parameter settings follow the last subsection and those stable cases are considered. System dynamics within 20 s are simulated, and the corresponding computational times are compared under different tolerances (Tol), as shown in Table III.

When $Tol = 10^{-15}$, an increase of the order leads to a longer execution time for the 9-bus and 39-bus systems since the increased complexity by using a higher order offsets the benefit from using a larger time step size. For these cases, the optimal order is 15. For the 2383-bus system, the optimal order is 25. Note that the execution time decreases when the

order changes from 15 to 25 but increases after the order of 25. This highlights the benefit of a higher-order solution for simulating a large-scale system, allowing for a larger step size to accelerate the simulation. However, the benefit may decrease and vanish beyond a certain order.

Similar situations occur in the 9-bus and 39-bus systems when Tol is increased to 10^{-20} . 25 is the ideal order with minimum execution time. However, for the 2383-bus system, the minimum execution time is achieved at order 35. When Tol approaches 10^{-25} , the ideal order for the 9-bus system is 45, but for the 39-bus system, it is 35. This is because the lower dimension of the 9-bus system allows for employing a higher order VS-DT method, with a larger time step size, which outperforms the use of a lower order model with a smaller step size. For large-scale systems, the optimal order of the VS-DT method remains 35 when $Tol = 10^{-25}$, indicating that simply increasing the order does not always help reduce computational time. While the VS-DT method provides promising results across different models and tolerances, the determination of a universally ideal order remains unsolved. However, the results of the VSOO-DT method outperform the VS-DT method under all tolerances because of the additional optimal order control strategy. These findings affirm the robustness and efficiency of the proposed optimal order strategy in comparison to the VS-DT method.

TABLE III
COMPUTATIONAL TIME WITH DIFFERENT FIXED ORDER VS-DT METHODS

Order K	Tolerance Tol	CPU Time(s) 9-bus	CPU Time(s) 39-bus	CPU Time(s) 2383-bus
15	10^{-15}	0.2887	0.3356	18.9236
	10^{-20}	0.7169	0.6177	45.3932
	10^{-25}	1.5774	1.5614	110.2162
25	10^{-15}	0.3336	0.3629	10.4229
	10^{-20}	0.5263	0.4411	20.9032
	10^{-25}	0.8235	0.7110	30.4770
35	10^{-15}	0.4468	0.5695	14.4394
	10^{-20}	0.5981	0.5836	17.7867
	10^{-25}	0.7329	0.6255	22.4509
45	10^{-15}	0.5680	0.7359	24.5646
	10^{-20}	0.6138	0.8080	25.6463
	10^{-25}	0.7029	0.7811	26.6313
Optimal Order (Start from $K_0 = 8$)	10^{-15}	0.3712	0.3291	10.1469
	10^{-20}	0.6047	0.4976	14.5964
	10^{-25}	0.8858	0.5557	21.0387

To further compare the numerical stability and speed of the DT, VS-DT, and VSOODT methods, consider a three-phase grounding fault applied to bus 3 and cleared after 0.2s for the 2383-bus system. Parameters in Table II are utilized without setting h_{max} . A large step size $h_0 = 0.1$ s is employed and the error is shown in Fig. 4a. The errors of the VS-DT and VSOO-DT methods can be maintained at less than 10^{-2} but the DT method is diverged. It can be concluded that both VS-DT and VSOO-DT methods improve the numerical stability compared with the DT method. Furthermore, results of the execution time are shown in Fig. 4b. While the VS-DT method is numerically stable under a large step size, its order $K = 4$ still limits the computational performance. The proposed VSOO-DT method achieves the same level of accuracy with a much less execution

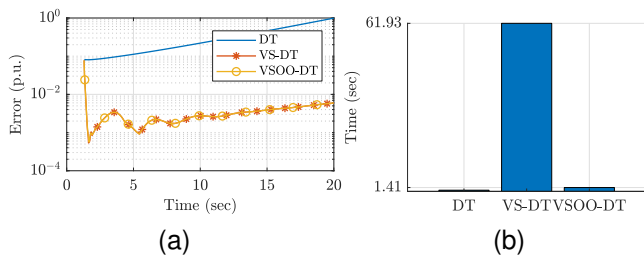


Fig. 4. Stability comparison for the DT, VS-DT, and VSOO-DT methods for the 2383-bus system. (a)Errors comparison. (b)Execution time comparison.

time (over 40 times speed up) because of the proposed optimal order strategy. In summary, the failure of the original DT method due to the improper step size and order can be partially solved by the VS-DT method and fully addressed by the VSOO-DT method.

C. Comparison with Classical Numerical Method

The proposed method is compared with the RK4 method which is applied in the power system simulation on three systems under the same error level. For the 9-bus system and 39-bus system, previous stable cases are still considered. The results are shown below in Table IV, it can be observed the time performances of the VSOO-DT method are much better than those of the RK4 method under all different levels of errors. A more comprehensive case involving a large disturbance for

TABLE IV
COMPUTATIONAL TIMES WITH THE RK4 AND VSOO-DT METHODS

System	Error	CPU Time(s) (RK4)	CPU Time(s) (VSOO-DT)
9-bus	10^{-3}	0.2303	0.0812
	10^{-4}	0.4377	0.0933
	10^{-5}	0.5586	0.1078
39-bus	10^{-3}	0.1469	0.0922
	10^{-4}	0.3041	0.1410
	10^{-5}	0.4038	0.1564

the 2383-bus system is studied. A three-phase grounding fault is applied to bus 1681 and then is cleared after 0.2 s. The simulation covers the system's dynamics over 20 seconds, and the resulting errors and computational times are compared across various numerical methods, including the RK4 method, the Modified Euler (ME) method, and the Trapezoidal Rule (TR) method. Note that the TR method refers to the optimized and superior MATLAB built-in solver (ode23t), which also incorporates the variable step strategy. Table V provides the comparison of the execution times and mean-max errors, here "mean" errors represent average errors for maximum errors along the time. Fig 5 displays the maximum errors during the post-fault stage for different methods. The proposed VS-DT and VSOO-DT methods exhibit a significant advantage in terms of both efficiency and accuracy when compared to RK4, ME, and TR methods. In particular, to compare with the conventional DT method, both DT and VSOO-DT methods start from identical conditions, i.e., $K = K_0 = 4$, $h = h_0 = 0.008s$ and $\mu_{in} = 1.5$, and other parameters are shown in Table II. The results show that the VSOO-DT method is faster than the DT

and VS-DT methods and it has the same level of high accuracy compared to the VS-DT method. The proposed VSOO-DT method effectively adjusts the order and step size on the fly of the simulation, affirming the efficiency and importance of the optimal order strategy. Moreover, the performance of the DT highly depends on the selection of the initial step size and order, which requires extra experience. This limitation has been verified in the previous stability study. In cases where the settings are not ideal, the execution time can become prolonged, or the accuracy may not be promising.

In summary, the proposed VSOO-DT method not only attains high accuracy like the VS-DT method but also has the fastest computational speed among all other methods even when the initial settings are not ideal.

TABLE V
PERFORMANCES OF DIFFERENT METHODS ON THE 2383-BUS SYSTEM

Method	CPU time (s)	Mean-Max Error(p.u.)
VSOO-DT ($Tol = 10^{-5}$, $K_0 = 4$)	1.8943	8.79×10^{-8}
VS-DT ($Tol = 10^{-5}$, $K = 8$)	3.3317	3.11×10^{-8}
DT ($h = 0.008$ s, $K = 4$)	3.7092	1.63×10^{-5}
RK4 ($h = 0.005$ s)	4.3180	2.49×10^{-6}
RK4 ($h = 0.01$ s)	2.2454	3.80×10^{-5}
ME ($h = 0.001$ s)	11.7445	2.15×10^{-4}
ME ($h = 0.005$ s)	2.4866	0.0054
TR ($tol = 10^{-4}$)	146.8446	0.017
TR ($tol = 10^{-2}$)	44.8635	0.033

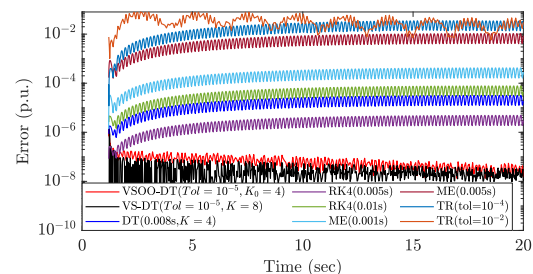


Fig. 5. Simulation errors of the 2383-bus system using different methods

While the TR and ME methods cannot simultaneously match the accuracy and speed of the VSOO-DT method, other traditional methods like the RK4 method require an extremely small time step size, like 0.001 s, to achieve a comparable error level to the VSOO-DT method, but it significantly slows down the computational speed. To provide a more comprehensive comparison, a MATLAB built-in variable step solver (ode45), i.e., the RK45 method is also considered, which is known for its success and robustness. By adjusting the tolerances for both VSOO-DT and RK45 methods, and the step sizes for the RK4 method, execution times across various levels of mean-max errors are compared in Fig 6. The results clearly demonstrate that the VSOO-DT method outperforms the RK4 and RK45 methods under different error requirements. Its impressive efficiency and scalability are further indicated by the nearly flat slope observed in the curve of the VSOO-DT method. Slopes of these characteristic curves represent the increased computational effort for increasing the accuracy. The flat slope of the VSOO-DT method verifies its adaptability through flexible order and step adjustments. To demonstrate the

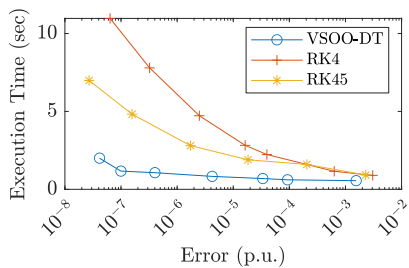


Fig. 6. Comparison of execution times for different errors by three methods

robustness and efficiency of the proposed VSOO-DT method, a dynamic N-1 stability assessment is studied. All three-phase grounding faults in the Polish 2383-bus system are considered, and each fault lasts 0.2 seconds. The default simulation time is 20 s. Simulations would be terminated if the maximum relative rotor angle exceeds 1000 degrees, indicating instability of the system. The comparison between the DT, VS-DT, VSOO-DT, RK45 (tolerance set at 10^{-8}), and ME (with a step size of 0.001s) methods is shown in Fig 7. The N-1 stability assessment includes all 2898 contingencies which can effectively verify the scalability and robustness of the proposed methods.

The initial orders of the DT, VS-DT, and VSOO-DT methods are 8 for a fair comparison. The step size of the DT is 0.01 s to prevent the numerical collapse. Despite the DT's capacity for excessive accuracy, it is unnecessarily time-consuming. This aligns with the observation that the DT requires tiny step sizes to maintain numerical stability, resulting in time wastage. Note that all methods based on the variable step strategy achieve faster speeds. However, a fixed optimal order and step size for the DT method may exist, but there is no general way to find it. And this is the motivation to design the proposed VSOO-DT method. Notably, the VSOO-DT method achieves a remarkable 10-fold speedup compared to the ME and a 2-fold speedup compared to the RK45 method, while the latter is the fastest conventional numerical method with promising numerical stability. Importantly, the execution times of the VSOO-DT method exhibit even less variation than those of the RK45 method, all while maintaining the same accuracy level (lower than 10^{-7}). This N-1 study validates the importance of generalizing the SAS method with the variable step size and order strategies in large-scale simulations. Based on the VSOO-DT method, commercial software tools can be developed with improved accuracy and numerical stability of simulation for efficient dynamic stability assessment.

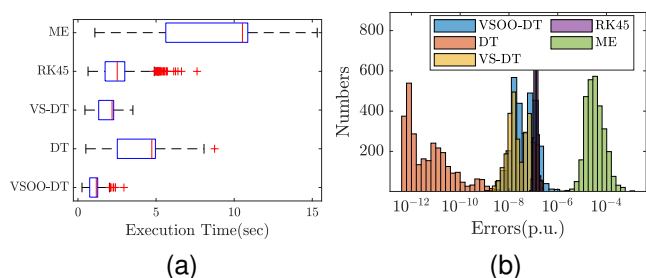


Fig. 7. Distribution of N-1 dynamic screenings. (a) Execution times distributions. (b) Mean-Max error distributions.

VI. CONCLUSION

This paper addresses a critical gap in power system dynamic simulations, where conventional methods often rely on the fixed step size and order, lacking a systematic approach to optimize these parameters on the fly of the simulation. While the DT method has demonstrated the potential for increased step sizes using SAS, the theoretical studies of the optimal relationship between the step size and order remain understudied.

Two innovative strategies are proposed to solve this challenge: the VS-DT and the VSOO-DT methods. The VS-DT introduces an automatic PI-controlled variable step strategy, leveraging a fixed order SAS to achieve high-speed performance without compromising accuracy, even for large-scale power systems. Building upon this, the VSOO-DT method optimally varies the order of the SAS, efficiently determining the optimal order for each step through a formulated optimization problem based on the complexity of the system model.

The key contributions of this paper are summarized. First, the VS-DT and VSOO-DT methods provide a dynamic framework for simulations, offering an optimal balance between numerical stability and simulation speed. The automatic adjustment of step size and the determination of the optimal SAS order are achieved on the fly of the simulation, ensuring efficiency and accuracy. Additionally, the proposed strategies are theoretically analyzed for numerical stability, offering insights into the selection of parameters.

Furthermore, case studies demonstrate the effectiveness of the proposed methods. Compared to existing approaches, the VS-DT and VSOO-DT methods exhibit robust and efficient simulations by adapting step size and optimizing the order as needed. Importantly, these strategies present a general framework applicable to the DT method and other semi-analytical or numerical methods with adjustable orders. This generality opens avenues for broader applications in power system simulations.

In summary, the proposed strategies represent a fundamental change in dynamic simulations, adapting the step size and optimizing the order for improved performance in different situations. The framework presented goes beyond the DT method, laying the foundation for progress in a wide range of simulation methods.

APPENDIX

Consider the detailed power system dynamic model depicted below, which includes several controllers:

$$T_{sv}\dot{p}_{sv} = -p_{sv} + p_{ref} - \frac{1}{R}s_m, \quad (39a)$$

$$T_{ch}\dot{p}_m = -p_m + p_{sv}, \quad (39b)$$

$$T_\omega\dot{v}_1 = -(K_\omega + K_p p_e + K_v v_t + v_1), \quad (39c)$$

$$T_e\dot{e}_{fd} = v_r - K_e e_{fd}, \quad (39d)$$

$$T_f\dot{v}_f = -v_f + K_f e_{fd}, \quad (39e)$$

$$T_r\dot{v}_{ts} = -v_{ts} + v_t, \quad (39f)$$

$$T_a\dot{v}_r = \begin{cases} -v_r + K_a(v_{ref} + v_s - v_{ts} - v_f), \\ \text{if } v_{rmin} < v_r < v_{rmax} \\ 0, \text{ if } v_{rmin} = v_r, \dot{v}_r < 0 \text{ or } v_{rmax} = v_r, \dot{v}_r > 0, \end{cases} \quad (39g)$$

where v_s is defined as $K_\omega + K_p p_e + K_v v_t + v_1$ and (39a) and (39b) represent the governor and turbine models, respectively. The PSS IEEE Type I model is given in (39c), while the simplified IEEE Type I exciter model is represented by (39d) through (39g). Furthermore, for the synchronous machine, a detailed 6th order model is considered:

$$\begin{aligned} \dot{\delta} &= \omega_s s_m, \\ 2H\dot{s}_m &= p_m - v_d i_d - v_q i_q - D s_m, \\ T'_{q0} \dot{e}'_d &= -\frac{x_q - x''_q}{x'_q - x''_q} e'_d + \frac{x_q - x'_q}{x'_q - x''_q} e''_d, \\ T'_{d0} \dot{e}'_q &= -\frac{x_d - x''_d}{x'_d - x''_d} e'_q + \frac{x_d - x'_d}{x'_d - x''_d} e''_q + e_{fd}, \\ T''_{q0} \dot{e}''_d &= e'_d - e''_d + (x'_q - x''_q) i_q, \\ T''_{d0} \dot{e}''_q &= e'_q - e''_q - (x'_d - x''_d) i_d, \end{aligned} \quad (40)$$

where $i_{xy} = \mathbf{Y}_r v_{xy}$ and

$$\begin{aligned} \begin{bmatrix} i_d \\ i_q \end{bmatrix} &= \begin{bmatrix} r_a & -x''_q \\ x''_d & r_a \end{bmatrix}^{-1} \left(\begin{bmatrix} e'_d \\ e'_q \end{bmatrix} - \begin{bmatrix} v_d \\ v_q \end{bmatrix} \right), \\ \begin{bmatrix} i_x \\ i_y \end{bmatrix} &= \mathbf{R} \begin{bmatrix} i_d \\ i_q \end{bmatrix}, \quad \begin{bmatrix} v_x \\ v_y \end{bmatrix} = \mathbf{R} \begin{bmatrix} v_d \\ v_q \end{bmatrix}, \end{aligned} \quad (41)$$

and

$$\mathbf{R} = \begin{bmatrix} \sin \delta & \cos \delta \\ -\cos \delta & \sin \delta \end{bmatrix}. \quad (42)$$

For a more in-depth understanding of (39) and (40), please refer to [19].

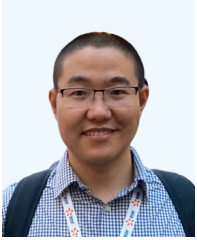
REFERENCES

- [1] P. Varaiya, F. Wu, and R.-L. Chen, "Direct methods for transient stability analysis of power systems: Recent results," *Proceedings of the IEEE*, vol. 73, no. 12, pp. 1703–1715, 1985.
- [2] H.-D. Chiang, F. Wu, and P. Varaiya, "A bcu method for direct analysis of power system transient stability," *IEEE Transactions on Power Systems*, vol. 9, no. 3, pp. 1194–1208, 1994.
- [3] H.-D. Chiang, *Direct methods for stability analysis of electric power systems: theoretical foundation, BCU methodologies, and applications*. John Wiley & Sons, 2011.
- [4] M. Anghel, F. Milano, and A. Papachristodoulou, "Algorithmic construction of lyapunov functions for power system stability analysis," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 60, no. 9, pp. 2533–2546, 2013.
- [5] F. Milano, "An open source power system analysis toolbox," *IEEE Trans. Power Syst.*, vol. 20, no. 3, pp. 1199–1206, 2005.
- [6] S. K. Khaitan, J. D. McCalley, and Q. Chen, "Multifrontal solver for online power system time-domain simulation," *IEEE Transactions on Power Systems*, vol. 23, no. 4, pp. 1727–1737, 2008.
- [7] D. Fabozzi, A. S. Chieh, B. Haut, and T. Van Cutsem, "Accelerated and localized newton schemes for faster dynamic simulation of large power systems," *IEEE Transactions on Power Systems*, vol. 28, no. 4, pp. 4936–4947, 2013.
- [8] P. Aristidou, D. Fabozzi, and T. Van Cutsem, "Dynamic simulation of large-scale power systems using a parallel schur-complement-based decomposition method," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 10, pp. 2561–2570, 2014.
- [9] J. Stiasny, S. Chatzivasileiadis, and B. Zhang, "Solving differential-algebraic equations in power systems dynamics with neural networks and spatial decomposition," *arXiv preprint arXiv:2303.10256*, 2023.
- [10] F. Milano, *Power system modelling and scripting*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010.
- [11] J. R. Dormand and P. J. Prince, "A family of embedded runge-kutta formulae," *J. Comput. Appl. Math.*, vol. 6, no. 1, pp. 19–26, 1980.
- [12] G. Tzounas, I. Dassios, and F. Milano, "Small-signal stability analysis of numerical integration methods," 2022.
- [13] —, "Small-signal stability analysis of implicit integration methods for power systems with delays," *Electric Power Systems Research*, vol. 211, p. 108266, 2022.

- [14] G. Tzounas and G. Hug, "Unified numerical stability and accuracy analysis of the partitioned-solution approach," *arXiv preprint arXiv:2304.05955*, 2023.
- [15] G. Gurralla, A. Dimitrovski, S. Pannala, S. Simunovic, and M. Starke, "Parareal in time for fast power system dynamic simulations," *IEEE Transactions on Power Systems*, vol. 31, no. 3, pp. 1820–1830, 2016.
- [16] S. Jin, Z. Huang, R. Diao, D. Wu, and Y. Chen, "Comparative implementation of high performance computing for power system dynamic simulations," *IEEE Trans. on Smart Grid*, vol. 8, no. 3, pp. 1387–1395, 2017.
- [17] T. Xiao, Y. Chen, S. Huang, T. He, and H. Guan, "Feasibility study of neural ode and dae modules for power system dynamic component modeling," *IEEE Transactions on Power Systems*, vol. 38, no. 3, pp. 2666–2678, 2023.
- [18] B. Wang, N. Duan, and K. Sun, "A time–power series-based semi-analytical approach for power system simulation," *IEEE Trans. Power Syst.*, vol. 34, no. 2, pp. 841–851, 2018.
- [19] Y. Liu, K. Sun, R. Yao, and B. Wang, "Power system time domain simulation using a differential transformation method," *IEEE Trans. Power Syst.*, vol. 34, no. 5, pp. 3739–3748, 2019.
- [20] Y. Liu and K. Sun, "Solving power system differential algebraic equations using differential transformation," *IEEE Trans. Power Syst.*, vol. 35, no. 3, pp. 2289–2299, 2019.
- [21] R. Yu, W. Gu, H. Lu, S. Yao, S. Zhang, S. Lu, S. Ding, and E. Luo, "Non-iterative calculation of quasi-dynamic energy flow in the heat and electricity integrated energy systems," *IEEE Trans. Power Syst.*, 2022.
- [22] N. Duan and K. Sun, "Power system simulation using the multistage adomian decomposition method," *IEEE Trans. Power Syst.*, vol. 32, no. 1, pp. 430–441, 2016.
- [23] R. Yao, Y. Liu, K. Sun, F. Qiu, and J. Wang, "Efficient and robust dynamic simulation of power systems with holomorphic embedding," *IEEE Trans. Power Syst.*, vol. 35, no. 2, pp. 938–949, 2019.
- [24] B. Park, K. Sun, A. Dimitrovski, Y. Liu, and S. Simunovic, "Examination of semi-analytical solution methods in the coarse operator of parareal algorithm for power system simulation," *IEEE Trans. Power Syst.*, vol. 36, no. 6, pp. 5068–5080, 2021.
- [25] K. Gustafsson, M. Lundh, and G. Söderlind, "A pi stepsize control for the numerical solution of ordinary differential equations," *BIT Numer. Math.*, vol. 28, pp. 270–287, 1988.
- [26] W. Enright, "Continuous numerical methods for odes with defect control," *Journal of Computational and Applied Mathematics*, vol. 125, no. 1, pp. 159–170, 2000, numerical Analysis 2000. Vol. VI: Ordinary Differential Equations and Integral Equations.
- [27] S. Lang, *Complex Analysis*, ser. Graduate Texts in Mathematics. Springer New York, 2003. [Online]. Available: <https://books.google.com/books?id=WZX4GepxPRgC>
- [28] E. Hairer, S. P. Norsett, and G. Wanner, *Solving ordinary differential equations I. Nonstiff Problems*, 2nd ed. Berlin: Springer, 1993.
- [29] G. Wanner and E. Hairer, *Solving ordinary differential equations II*. Springer Berlin Heidelberg New York, 1996, vol. 375.
- [30] K. Gustafsson, "Control theoretic techniques for stepsize selection in explicit runge-kutta methods," *ACM Trans. Math. Softw. (TOMS)*, vol. 17, no. 4, pp. 533–554, 1991.
- [31] Å. Jorba and M. Zou, "A software package for the numerical integration of ODEs by means of high-order Taylor methods," *Exp. Math.*, vol. 14, no. 1, pp. 99 – 117, 2005.
- [32] R. Barrio, F. Blesa, and M. Lara, "Vsvo formulation of the taylor method for the numerical solution of odes," *Comput. Math. with Appl.*, vol. 50, no. 1-2, pp. 93–111, 2005.

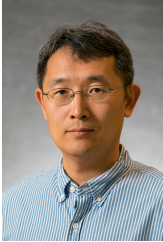


Kaiyang Huang (Student Membe, IEEE) received the B.S. degree in electrical engineering from North China Electric Power University, China, in 2020. He is currently pursuing the Ph.D. degree at the Department of Electrical Engineering and Computer Science, University of Tennessee, Knoxville, USA. His research interests include power system simulation, transient stability analysis, and dynamics.

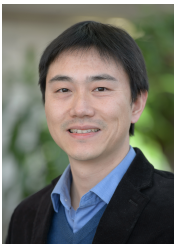


Yang Liu (Member, IEEE) received the B.S. degree in energy and power engineering from Xi'an Jiaotong University, China, in 2013, the M.S. degree in power engineering from Tsinghua University, China, in 2016, and the Ph.D. degree in electrical engineering from the University of Tennessee, Knoxville, USA, in 2021. He was a Post-Doctoral Research Associate with the University of Tennessee, in 2022. Currently, he is a Post-Doctoral Appointee with the Argonne National Laboratory. His research interests include stability, dynamics, and control of power grids and

other complex systems.



Kai Sun (Fellow, IEEE) is a professor at the Department of EECS, University of Tennessee, Knoxville, USA. He received the B.S. degree in automation in 1999 and the Ph.D. degree in control science and engineering in 2004 both from Tsinghua University, Beijing, China. From 2007 to 2012, he was a project manager in grid operations and planning with the Electric Power Research Institute (EPRI), Palo Alto, CA.



Feng Qiu (Senior Member, IEEE) received his Ph.D. from the School of Industrial and Systems Engineering at the Georgia Institute of Technology in 2013. He is a principal computational scientist and a section leader with the Energy Systems Division at Argonne National Laboratory, Argonne, IL, USA. His current research interests include power system modeling and optimization, electricity markets, power grid resilience, machine learning and data analytics.