A Numerical Solver Design for Extended-Term Time-Domain Simulation

Chuan Fu, *Student Member, IEEE*, James D. McCalley, *Fellow, IEEE*, and Jianzhong Tong, *Senior Member, IEEE*

***Abstract*—Numerical methods play an important role in improving efficiency for power system time-domain simulation. Motivated by the need to perform high-speed extended-term time-domain simulation (HSET-TDS) for online purposes, this paper presents design principles for numerical solvers of differential algebraic systems associated with power system time-domain simulation, focusing on DAE construction strategies, integration methods, nonlinear solvers, and linear solvers. We have implemented a design appropriate for HSET-TDS, and we have compared the proposed integration method, Hammer-Hollingsworth 4 (HH4), with Trapezoidal rule in terms of computational efficiency and accuracy, using the New England 39-bus system, an expanded 8775-bus system, and PJM 13 029-bus system.**

***Index Terms*—Cascading, extended-term, Hammer-Hollingsworth 4 (HH4), integration, numerical methods, power system dynamics, time-domain simulation.**

I. Introduction

TIME-DOMAIN simulation of power systems involve the solution of a large number of differential and algebraic equations (DAEs) [1], [2], which are constructed based on modeling of the power system networks and dynamic elements, including generators, exciters, governors and other devices. The general form of the DAEs can be described as

$$\begin{cases} \dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{y}) & (1a) \\ 0 = \mathbf{g}(\mathbf{x}, \mathbf{y}) & (1b) \end{cases} \tag{1}$$

where vector $\mathbf{x}$, $\mathbf{y}$ are differential and algebraic variables with dimension p and q respectively; (1a) and (1b) are ordinary differential equations (ODE) and algebraic equations, respectively.

Many papers and user manuals have been published on numerical methods used in software for the time-domain simulation of power systems, including [2]–[10]. Design of DAE numerical solvers requires designer decision at several levels including DAE construction strategy, integration method, nonlinear solver and linear solver. We view these decisions hierarchically as illustrated in Fig. 1, with broader, structural design decisions represented at the top. We describe each level in what follows:

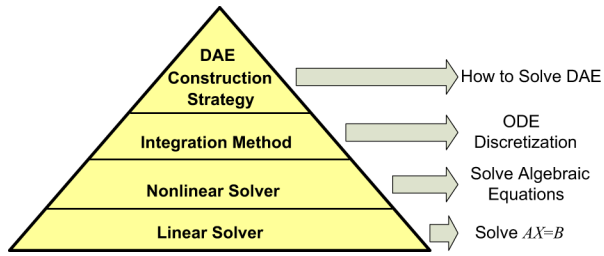

Fig. 1. Categories of numerical methods for time-domain simulation.

- DAE construction strategy: There are two basic strategies, depending on whether one solves the algebraic equations separately from the differential equations or simultaneously, with the former approach referred to as the alternating solution method (ASM), and the latter approach referred to as the direct solution method (DSM). ASM with explicit method and small integration step is used frequently in many softwares for short term time-domain simulation.
- Integration method: The integration method is used to discretize the ODE part of the DAE, resulting in algebraic equations that may or may not be solved with the network algebraic equations of (1b), depending on whether ASM or DSM is used and whether the integration method is implicit or explicit.
- Nonlinear solver: This is used to solve the nonlinear algebraic equations of the problem. If an ASM strategy is used with explicit integration, then these equations will be simply those of (1b). If an ASM strategy is used with implicit integration, then there will be two sets of nonlinear equations to solve—those of (1b) and those corresponding to the discretized ODEs. If a DSM strategy is used then there is a single set of nonlinear equations to be solved simultaneously comprised of (1b) together with the discretized ODEs. Gauss-Seidel is used in many commercial applications software such as [6] and [7] when predictor-corrector scheme is adopted. Waveform relaxation [11], [12], which is convenient and effective for parallel computing, is another Gauss-Seidel-like nonlinear solver. A Newton method is frequently chosen for DSM strategy and ASM with implicit method.
- Linear solvers: Linear solvers are necessary to solve the linear algebraic equations $AX = B$ when a DSM strategy with a Newton nonlinear solver is used and to solve the network equations if an ASM strategy is adopted. There are many open source linear solver libraries available

Manuscript received September 23, 2010; revised April 09, 2011 and September 09, 2011; accepted October 25, 2011. Date of publication January 13, 2012; date of current version October 17, 2013. This work was supported in part by the Power Systems Engineering Research Center and in part by the DOE Consortium for Electric Reliability Technology Solutions. Paper no. TPWRS-00750-2010.

C. Fu and J. D. McCalley are with the Electrical and Computer Engineering Department, Iowa State University, Ames, IA 50010 USA (e-mail: chuanfu@iastate.edu; jdm@iastate.edu).

J. Tong is with PJM, Norristown, PA 19403 USA (e-mail: tongji@pjm.com).

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.2011.2177674

such as GMM++ [13], SuperLU [14], and UMFpack [15]. Some papers reported on how to efficiently solve the linear equations to accelerate time-domain simulation by parallel computing, using the conjugate gradient method [16] and block bordered diagonal form [17].

We are motivated in this work to develop a new functionality for online applications which we refer to as high-speed extended-term time-domain simulator (HSET-TDS). Such functionality would provide extremely fast simulation of events for extended time periods of several hours, in order to capture the relatively slow degradation that is characteristic of some high-consequence scenarios (such as the 2003 Northeast U.S. Blackout [18]). Description of this functionality can be found in [19] and [20]. Although the work reported in this paper is inspired by the computational requirements of this new functionality, this work is also relevant to more general power system time-domain simulation applications.

The paper is organized as follows. Section II describes the two DAE construction strategies and provides rationale why use of DSM is attractive. Section III summarizes various implicit integration methods and identifies one, a variable-step method, as most appropriate for HSET-TDS. Section IV introduces the nonlinear algebraic solvers used in HSET-TDS. Section V describes the selected linear solver. Section VI provides simulation results on the New England 39-bus system, an expanded 8775-bus system, and PJM 13 029-bus system. Section VII concludes.

II. Construction of DAE System

For a power system DAE solution, the choice of DAE construction strategy, ASM or DSM, may affect the efficiency of the solution of the sub-DAE system describing dynamics models and algebraic equations describing networks model. We describe ASM in Section II-A and then DSM in Section II-B together with its implementation in HSET-TDS.

A. Alternating Solution Method (ASM)

References [1] and [2] described the alternating solution method [2] (or partitioned-solution approach [1]), the main idea of which is to alternatively solve for the injected currents of the dynamic elements and the voltages of the nodes within the network by assuming the last step values of node voltages as the initial iterative values. ASM, therefore, divides the DAE system into two parts, the linear equations for the network and sub-DAE system for the dynamic elements. When explicit integration is used, one iterative loop is needed for convergence of node voltages, and if the integration step is small, just one iteration may be needed. The explicit method, such as the modified Euler method with a small integration step, e.g., a half cycle, is often used for short-term time-domain simulation [3], [4]. For the extended-term application, such an approach proves computationally intensive, and attempts to reduce computation via increased integration steps risk numerical instability.

The implicit method, such as the trapezoidal rule, has been used with ASM for extended-term time-domain simulation [4]. There are two loops during the process of computing the next step value of the DAE system, the main iterative loop and the sub-iterative loop. The main loop iterates on network bus voltages using Gauss-Seidel, while the sub-loops iterate on the differential variables describing the dynamic devices. In the sub-loops, an explicit method is usually used as predictor and an implicit method is used as corrector [6], [7]. If the integration step is large, more iterations may be needed to make the differential variables converge. The advantage of ASM combined with the implicit method is that the generators can be represented by constant current sources, and admittances can be represented in the network admittance matrix. The resulting linear complex algebraic equations can be solved efficiently to obtain the bus voltages. However, since the Gauss-Seidel corrections of the main loop use sub-optimal directions, the main loop may require a larger number of iterations, with each iteration requiring a new set of sub-iterative loops (one for each generator), especially when the integration step is large. As a result, simulation based on ASM with implicit integration can be more computationally intensive compared to simulations where Newton's method is used when integration step is large.

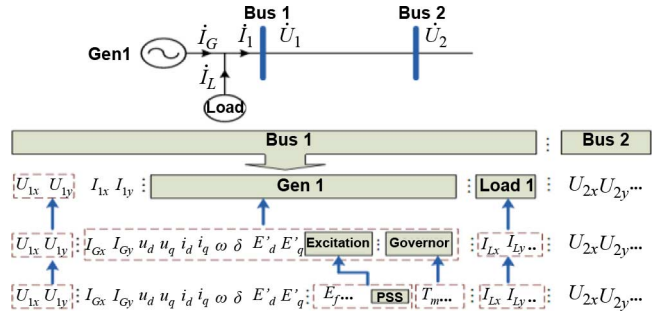

Fig. 2. Organization structure of differential and algebraic variables. (The generator shown is of 4th-order model.)

B. Direct Solution Method (DSM)

Unlike ASM, the DSM (or simultaneous-solution approach in [1]) combines the discretized ODEs and the algebraic equations describing the network. One difficulty in realizing DSM is how to dynamically organize differential and algebraic variables since power systems have different topological structures. The organizational structure shown in Fig. 2 is adopted within HSET-TDS. Each variable corresponds to a particular set of differential equations or a particular set of algebraic equations. The functions of evaluating the equation and of taking the first derivative of the equation are programmed into software for the construction of DAE system and corresponding Jacobian matrix.

Whereas ASM requires Gauss-Seidel, a non-gradient-based method, in the iteration of network bus voltages, DSM allows gradient-based (Newton) methods such that each iteration's new solution is obtained along an optimal direction, and hence the number of iterations can be reduced when a large integration step is used. When large integration steps are used, this attribute provides DSM with some degree of computational efficiency that cannot be realized by ASM.

III. Integration Methods

Integration methods can be categorized by three attributes, 1) explicit or implicit methods, 2) single-step or multi-step

methods, and 3) fixed step or variable step methods. Attributes 1) and 2) are identified from the integration formula, while attribute 3) depends on whether local truncation error is estimated and whether a variable step technique is adopted.

Criteria to judge whether an integration method is suitable for time-domain simulation is whether the integration method is able to avoid 1) numerical instability [1], [21], [22] and 2) hyper-stability [7]. Numerical instability may occur when an explicit method and a large integration step is used, and there are two ways to avoid it. The first is to use a higher order explicit method with a larger stability domain, and the second is to decrease the integration step. A-stable implicit methods do not incur numerical instability since their stability domain includes the entire left-half of the complex plane.

Hyper-stability, the numerical stabilization of an unstable system, occurs when right-half-plane eigenvalues λ of the linearized system, discretized as $h\lambda$ (h is time step), locate inside the stability domain of the integration method [7], [23]. Hyper-stability is avoided when the stability domain of an integration method used does not include the right part of the complex plane.

The trapezoidal rule has been used in many time-domain simulation software applications, including [2], [5]–[7] and the extended term version of [4]. The strength of the trapezoidal rule is that it avoids both numerical instability and hyper-stability. It can also achieve good simulation efficiency without sacrificing precision by using variable step integration [6], [7], which adjusts the integration step based on estimated local error. However, the integration step for the trapezoidal rule is limited by its precision h^2.

In this section, an integration method, Hammer-Hollingsworth 4 (HH4) [24], [25], is presented and discussed. In comparison to other Runge-Kutta methods (both explicit and implicit), HH4 possesses two attributes which can be beneficial for time-domain simulation, 1) symmetrical A-stability and 2) higher precision than that of trapezoidal rule.

A. Formula of HH4

Reference [24] introduces expressions for the HH4 integration method as follows:

$$\begin{cases} K_1 = f\left(x_n + h\left[\frac{1}{4}K_1 + \left(\frac{1}{4} - \frac{\sqrt{3}}{6}\right)K_2\right]\right) \\ K_2 = f\left(x_n + h\left[\left(\frac{1}{4} + \frac{\sqrt{3}}{6}\right)K_1 + \frac{1}{4}K_2\right]\right) \\ x_{n+1} = x_n + \frac{h}{2}\left[K_1 + K_2\right] \end{cases} \tag{2}$$

where K_1 and K_2 are scalars. The formula (2) corresponds to the ODE $\dot{x} = f(x), x(t_0) = x_0$. If we apply the formula (2) into the DAE system (1), the HH4 integration method can be expressed as a two-stage formula (3) and (4):

$$\begin{cases} \mathbf{K_1} = \mathbf{f}\left(\left(\mathbf{x}_n + h\left[\frac{1}{4}\mathbf{K_1} + \left(\frac{1}{4} - \frac{\sqrt{3}}{6}\right)\mathbf{K_2}\right]\right), \mathbf{y_{K_1}}\right) \\ 0 = \mathbf{g}\left(\left(\mathbf{x}_n + h\left[\frac{1}{4}\mathbf{K_1} + \left(\frac{1}{4} - \frac{\sqrt{3}}{6}\right)\mathbf{K_2}\right]\right), \mathbf{y_{K_1}}\right) \\ \mathbf{K_2} = \mathbf{f}\left(\left(\mathbf{x_n} + h\left[\left(\frac{1}{4} + \frac{\sqrt{3}}{6}\right)\mathbf{K_1} + \frac{1}{4}\mathbf{K_2}\right]\right), \mathbf{y_{K_2}}\right) \\ 0 = \mathbf{g}\left(\left(\mathbf{x_n} + h\left[\left(\frac{1}{4} + \frac{\sqrt{3}}{6}\right)\mathbf{K_1} + \frac{1}{4}\mathbf{K_2}\right]\right), \mathbf{y_{K_2}}\right) \end{cases} \tag{3}$$

$$\begin{cases} \mathbf{x_{n+1}} = \mathbf{x_n} + \frac{h}{2}\left[\mathbf{K_1} + \mathbf{K_2}\right] \\ 0 = \mathbf{g}(\mathbf{x}_{n+1}, \mathbf{y}_{n+1}) \end{cases} \tag{4}$$

where $\mathbf{K}_1$, $\mathbf{y}_{\mathrm{K1}}$, $\mathbf{K}_2$, and $\mathbf{y}_{\mathrm{K2}}$ are vectors with dimension p, q, p, and q, respectively; $\mathbf{y}_{\mathrm{K1}}$ and $\mathbf{y}_{\mathrm{K2}}$ are the algebraic variables corresponding to the first two and second two equations, respectively, of (3). The Jacobian matrix of (3) is shown in (5) at the bottom of the page, where $\mathbf{E}$ and $\mathbf{O}$ are the unit matrix and zero matrix, respectively.

In order to decrease the number of nonzero items of Jacobian matrix, we introduce the following transformation:

$$\begin{cases} \xi = x_n + h\left[\frac{1}{4}K_1 + \left(\frac{1}{4} - \frac{\sqrt{3}}{6}\right)K_2\right] \\ \eta = x_n + h\left[\left(\frac{1}{4} + \frac{\sqrt{3}}{6}\right)K_1 + \frac{1}{4}K_2\right] \end{cases} \tag{6}$$

where η and ξ are scalars. The expression (6) can be transformed into (7)

$$\begin{cases} hK_1 = \left(2\sqrt{3} - 3\right)(\eta - x_n) + 3(\xi - x_n) \\ hK_2 = 3(\eta - x_n) - \left(2\sqrt{3} + 3\right)(\xi - x_n). \end{cases} \tag{7}$$

Thus the HH4 formula (2) can be transformed into (8)

$$\begin{cases} (2\sqrt{3} - 3)(\eta - x_n) + 3(\xi - x_n) = hf(\xi) \\ 3(\eta - x_n) - (2\sqrt{3} + 3)(\xi - x_n) = hf(\eta) \\ x_{n+1} = x_n + \sqrt{3}(\eta - \xi). \end{cases} \tag{8}$$

For the multidimensional DAE system (1), the solution of $\mathbf{x}_{\mathrm{n+1}}$ by the HH4 integration method can be described as two stages, 1) solution of vector $\boldsymbol{\xi}$, $\mathbf{y}_{\boldsymbol{\xi}}$, $\boldsymbol{\eta}$, and $\mathbf{y}_{\boldsymbol{\eta}}$, and 2) solution of $\mathbf{x}_{\mathrm{n+1}}$, $\mathbf{y}_{\mathrm{n+1}}$. For the first stage, we need to solve the following nonlinear algebraic equations:

$$\begin{cases} (2\sqrt{3} - 3)(\boldsymbol{\eta} - \mathbf{x_n}) + 3(\boldsymbol{\xi} - \mathbf{x_n}) = h\mathbf{f}(\boldsymbol{\xi}, \mathbf{y}_{\boldsymbol{\xi}}) \\ 0 = \mathbf{g}(\boldsymbol{\xi}, \mathbf{y}_{\boldsymbol{\xi}}) \\ 3(\boldsymbol{\eta} - \mathbf{x_n}) - (2\sqrt{3} + 3)(\boldsymbol{\xi} - \mathbf{x_n}) = h\mathbf{f}(\boldsymbol{\eta}, \mathbf{y}_{\boldsymbol{\eta}}) \\ 0 = \mathbf{g}(\boldsymbol{\eta}, \mathbf{y}_{\boldsymbol{\eta}}). \end{cases} \tag{9}$$

$$\begin{bmatrix} \mathbf{K_1} & \mathbf{y_{K_1}} & \mathbf{K_2} & \mathbf{y_{K_2}} \\ -\mathbf{E}_{p\times p} + \frac{h}{4}\left[\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\Big|_{\mathbf{x}=\mathbf{x}_n + h[1/4\mathbf{K_1} + (1/4 - \sqrt{3}/6)\mathbf{K_2}]}\right]_{p\times p} & \left[\frac{\partial \mathbf{f}}{\partial \mathbf{y}}\Big|_{\mathbf{y}=\mathbf{y_{K_1}}}\right]_{p\times q} & h\left(\frac{1}{4} - \frac{\sqrt{3}}{6}\right)\left[\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\Big|_{\mathbf{x}=(\mathbf{x_n} + h[(1/4 + \sqrt{3}/6)\mathbf{K_1} + 1/4\mathbf{K_2}])}\right]_{p\times p} & \mathbf{O}_{p\times q} \\ \frac{h}{4}\left[\frac{\partial \mathbf{g}}{\partial \mathbf{x}}\Big|_{\mathbf{x}=\mathbf{x}_n + h[1/4\mathbf{K_1} + (1/4 - \sqrt{3}/6)\mathbf{K_2}]}\right]_{q\times p} & \left[\frac{\partial \mathbf{g}}{\partial \mathbf{y}}\Big|_{\mathbf{y}=\mathbf{y_{K_1}}}\right]_{q\times q} & h\left(\frac{1}{4} - \frac{\sqrt{3}}{6}\right)\left[\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\Big|_{\mathbf{x}=\mathbf{x}_n + h[1/4\mathbf{K_1} + (1/4 - \sqrt{3}/6)\mathbf{K_2}]}\right]_{p\times p} & \mathbf{O}_{q\times q} \\ h\left(\frac{1}{4} - \frac{\sqrt{3}}{6}\right)\left[\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\Big|_{\mathbf{x}=\mathbf{x}_n + h[1/4\mathbf{K_1} + (1/4 - \sqrt{3}/6)\mathbf{K_2}]}\right]_{p\times p} & \mathbf{O}_{p\times q} & -\mathbf{E}_{p\times p} + \frac{h}{4}\left[\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\Big|_{\mathbf{x}=(\mathbf{x_n} + h[(1/4 + \sqrt{3}/6)\mathbf{K_1} + 1/4\mathbf{K_2}])}\right]_{p\times p} & \left[\frac{\partial \mathbf{f}}{\partial \mathbf{y}}\Big|_{\mathbf{y}=\mathbf{y_{K_2}}}\right]_{p\times q} \\ h\left(\frac{1}{4} - \frac{\sqrt{3}}{6}\right)\left[\frac{\partial \mathbf{g}}{\partial \mathbf{x}}\Big|_{\mathbf{x}=\mathbf{x}_n + h[1/4\mathbf{K_1} + (1/4 - \sqrt{3}/6)\mathbf{K_2}]}\right]_{q\times p} & \mathbf{O}_{q\times q} & \frac{h}{4}\left[\frac{\partial \mathbf{g}}{\partial \mathbf{x}}\Big|_{\mathbf{x}=(\mathbf{x_n} + h[(1/4 + \sqrt{3}/6)\mathbf{K_1} + 1/4\mathbf{K_2}])}\right]_{q\times p} & \left[\frac{\partial \mathbf{g}}{\partial \mathbf{y}}\Big|_{\mathbf{y}=\mathbf{y_{K_2}}}\right]_{q\times q} \end{bmatrix} \tag{5}$$

For the second stage we need to solve

$$\begin{cases} \mathbf{x}_{n+1} = \mathbf{x}_n + \sqrt{3}(\boldsymbol{\eta} - \boldsymbol{\xi}) \\ 0 = \mathbf{g}(\mathbf{x}_{n+1}, \mathbf{y}_{n+1}) \end{cases} \tag{10}$$

where $\boldsymbol{\eta}$ and $\mathbf{y}_{\boldsymbol{\eta}}$ are used as initial values of $\mathbf{x}_{n+1}$ and $\mathbf{y}_{n+1}$ for the Newton method. The Jacobian matrix of (9) is expressed as (11). It can be seen that the number of nonzero items in (11) are decreased compared with (10). See (11) at the bottom of the page.

The "cost" of applying HH4 in time-domain simulation is that the nonlinear algebraic equations that must be solved at each integration step are twice the dimension as those that must be solved using the trapezoidal rule, an attribute that can be observed in (9) and (11). However, when using the Newton method to solve these equations, the Jacobian (11) is highly sparse, and therefore the "cost" can be minimized by utilizing a sparse linear solver library.

B. Variable Integration Step Control

Step control techniques and error estimation are used in many time-domain simulation applications [6], [7]. Due to the fact that there is significant computation during one HH4 integration step, integration step control is necessary to enlarge the integration step as much as possible. In HSET-TDS, the local error is estimated by comparing the simulation result with that from another integration method (usually the explicit method is chosen for its high efficiency). The error estimation is evaluated by

$$\left|\tilde{E}(x_{n+1})\right| = c_e \left|x_{n+1} - \tilde{x}_{n+1}\right| \tag{12}$$

where $\mathbf{x}_{n+1}$ is the next step value calculated by one integration method, $\tilde{x}_{n+1}$ is the value calculated by another integration method, and c_e is the error coefficient obtained as the difference between the Taylor expansion of $\mathbf{x}_{n+1}$ and $\tilde{x}_{n+1}$. The 4th-order explicit Runge-Kutta method or the 4th-order explicit Adams [24] method with variable step can be used for error estimation in applying HH4. When there is a switching event in a simulation process such as a fault shown in Fig. 3, the integration step is adjusted to be zero after the occurrence of the fault, and thus the estimated local error at the initiation of the switching event is zero. After the switching event, the logic resumes the variable integration step control beginning with a small integration step (0.001 s).

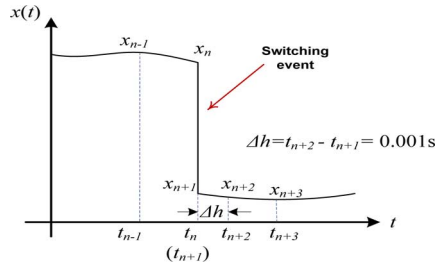

Fig. 3. Switching event in the simulation.

HSET-TDS adopts double thresholds to control the integration step, an upper threshold, and a lower threshold. If the error norm exceeds the upper threshold, the current integration step is decreased; if the error norm is smaller than the lower threshold, the next integration step is increased. Selection of the upper and lower bounds is done to achieve the right tradeoff on accuracy and speed. Raising the upper error bound or raising the lower error bound decreases accuracy while increasing computational speed. Although these values may need tuning to achieve optimal accuracy-speed tradeoffs for a particular system, one general guideline is that the lower error bound should not be smaller than the precision used in the nonlinear solver of the time-domain simulation; doing otherwise can unnecessarily increase integration step size change. This double threshold approach, compared to a single threshold approach, decreases the frequency at which the time step is changed and therefore improves simulation efficiency by reducing recalculations following a change of time step.

C. Discussion of HH4

The stability domain of HH4 is the left-half complex plane [23], and thus the method of HH4 is symmetrically A-stable, avoiding both numerical instability and hyper-stability. Additionally, HH4 guarantees precision of h^4 with the local error (the difference between the next step value produced by the integration method and the exact next step value) of $O(h^5)$ which can be proven using the Taylor series expansion. The high precision of HH4 enhances simulation efficiency by allowing increased integration step while maintaining precision. For example, if the

$$\begin{bmatrix} \boldsymbol{\xi} & \mathbf{y}_{\boldsymbol{\xi}} & \boldsymbol{\eta} & \mathbf{y}_{\boldsymbol{\eta}} \\ -3\mathbf{E}_{p\times p} + h\left[\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\Big|_{\mathbf{x}=\boldsymbol{\eta}}\right]_{p\times p} & \left[\frac{\partial \mathbf{f}}{\partial \mathbf{y}}\Big|_{\mathbf{y}=\mathbf{y}_{\mathbf{K}_1}}\right]_{p\times q} & (3-2\sqrt{3})\mathbf{E}_{p\times p} & \mathbf{O}_{p\times q} \\ \left[\frac{\partial \mathbf{g}}{\partial \mathbf{x}}\Big|_{\mathbf{x}=\boldsymbol{\eta}}\right]_{q\times p} & \left[\frac{\partial \mathbf{g}}{\partial \mathbf{y}}\Big|_{\mathbf{y}=\mathbf{y}_{\boldsymbol{\eta}}}\right]_{q\times q} & \mathbf{O}_{q\times p} & \mathbf{O}_{q\times q} \\ (3+2\sqrt{3})\mathbf{E}_{p\times p} & \mathbf{O}_{p\times q} & -3\mathbf{E}_{p\times p} + h\left[\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\Big|_{\mathbf{x}=\boldsymbol{\eta}}\right]_{p\times p} & h\left[\frac{\partial \mathbf{f}}{\partial \mathbf{y}}\Big|_{\mathbf{y}=\mathbf{y}_{\boldsymbol{\eta}}}\right]_{p\times q} \\ \mathbf{O}_{q\times p} & \mathbf{O}_{q\times q} & \left[\frac{\partial \mathbf{g}}{\partial \mathbf{x}}\Big|_{\mathbf{x}=\boldsymbol{\eta}}\right]_{q\times p} & \left[\frac{\partial \mathbf{g}}{\partial \mathbf{y}}\Big|_{\mathbf{y}=\mathbf{y}_{\mathbf{K}_2}}\right]_{q\times q} \end{bmatrix} \tag{11}$$

numerical precision of 10^{-4} is required, the maximum integration step for trapezoidal rule is 0.01 because of it 2nd-order precision. In contrast, HH4 may use a time step of 0.1 due to its 4th-order precision (h^4).

References [6] and [26] discussed and mentioned the problem of numerical oscillations, caused by the DAE Jacobian having an eigenvalue with a small real part and a large imaginary part. There are two ways of eliminating the numerical oscillation: 1) using a small integration step, or 2) applying the integration method whose stability domain covers the imaginary axis, such as the trapezoidal rule with damping [26]. Due to the attribute of the high precision, HH4 may avoid numerical oscillations when using an integration step for which the trapezoidal rule would not.

IV. Nonlinear Algebraic Equation Solver

The two most common nonlinear algebraic equation solvers used for power system time-domain simulation are Gauss-Seidel and Newton-Raphson (NR). We have used Newton-Raphson in HSET-TDS and describe its implementation in this section.

A main attribute of the NR method is that the solution process does not require many iterations compared with the Gauss-Seidel method because of the optimum choice in selecting the direction of variable change in each iteration. Each NR iteration involves a solution of simultaneous linear algebraic equations, the part of the NR process that requires the most computational time.

There are two problems associated with the efficiency of the NR method. First, the linear algebraic equations solved in each NR iteration are usually of large dimensions but are of high sparsity requiring a sparse linear solver to avoid excessive computation time. Additionally, when the integration step becomes large, failure of the NR method may occur because of inappropriate starting points. Reference [6] introduced a modified NR method where a deceleration factor is introduced. Use of a deceleration factor provides that failures due to cycling may be avoided; however, the cost is that the selected direction of variable change in each iteration is no longer optimal, and thus more iterations are needed.

A strategy used in HSET-TDS to address NR failures is step control, and the number of NR iterations is the criterion used in determining whether to adjust the integration step. For the nonlinear algebraic equations $G(X) = 0$, the modified NR method is described as follows:

$$\begin{cases} \mathbf{X}^{(n+1)} = \mathbf{X}^{(n)} - \boldsymbol{\alpha} \cdot \Delta\mathbf{X} \\ [\mathbf{J}]\Delta\mathbf{X} = [\mathbf{b}] \end{cases} \tag{13}$$

where $[J] = [\partial G/\partial X]\big|_{x=x^{(n)}}$ denotes the Jacobian matrix; α is the deceleration factor; and $[b] = [G(X^{(n)})]$. Initially the deceleration factor is 1.0, and it is decreased by a small value when the max norm of ΔX exceeds a threshold. If the number of iterations exceed a set value, the NR method is considered failed, and then the integration step is decreased to obtain an improved starting point. The NR iterations will continue until ΔX falls below a tolerance.

Another strategy in HSET-TDS to enhance efficiency of the NR method, which is called the Very DisHonest Newton (VDHN) method [7], [27], is to keep the Jacobian matrix constant over many integration steps. The Jacobian matrix is not updated until the number of VDHN iterations exceed a specified limit (5 is used in HSET-TDS). When the Jacobian matrix needs to be updated, VDHN is switched to a standard Newton. The advantage of VDHN is that the time dedicated to solving linear equations is decreased since the Jacobian factorization necessary to solve the linear equations is avoided for some iterations. In HSET-TDS, VDHN is implemented in coordination with SuperLU, a sparse linear solver discussed in the next section.

V. Sequential Library of SuperLU Solver

The process of solving linear algebraic equations is necessary in time-domain simulation when the NR method is used within a DSM strategy and also for the solution of networks equations when an ASM strategy is utilized. In both cases, solution of linear equations comprises a significant portion of the simulation time, and so use of an efficient linear solver is essential. Furthermore, because the Jacobian matrix of the DAE system is highly sparse, it is important that the linear solver incorporates sparsity techniques. In this section, SuperLU [14], [28], [29], [30], an open source library developed at the University of California, Berkeley, is introduced. We have deployed the sequential version within HSET-TDS (there is also a parallelized version).

Reference [28] introduces the sparse Gaussian elimination algorithm adopted in SuperLU, which can be summarized in two steps:

a) Compute a triangular factorization $P_r D_r A D_c P_c = LU$, where D_r and D_c are diagonal matrices to equilibrate the linear system, which means to scale the A-matrix to make the matrix norm smaller with respect to the eigenvalues. P_r and P_c are permutation matrices to reorder the rows and columns of A. L is a unit lower triangular matrix $(L_{ii} = 1)$ and U is an upper triangular matrix.
b) Solve $AX = B$ by evaluating

$$X = A^{-1}B = \left(D_r^{-1}P_r^{-1}LUP_c^{-1}D_c^{-1}\right)^{-1}$$
$$B = D_c\left(P_c\left(U^{-1}\left(L^{-1}\left(P_r\left(D_r B\right)\right)\right)\right)\right).$$

SuperLU supplies two main driver routines to call SuperLU library, 1) simple driver dgssv(), and 2) expert driver dgssvx(). Both of these driver routines implement the following operations: factorization, triangular solving, estimating condition number, equilibrating and refining the solution. In the expert driver routine, more options are provided to control SuperLU in terms of solving the linear equations. One useful option is the configuration of how to factorize the matrix A, which includes 1) normal factorization, 2) factorization with same pattern, 3) factorization with same pattern and same row, and 4) no factorization.

Option 2, factorization with same pattern, allows SuperLU to repeatedly use the diagonal matrices (D_r and D_c) and permutation matrices (P_r and P_c) generated from an initial normal factorization; thus, the efficiency of factorization can be enhanced. Option 3, factorization with same pattern and same row, can be more efficient than that of option 2, because option 3 directly

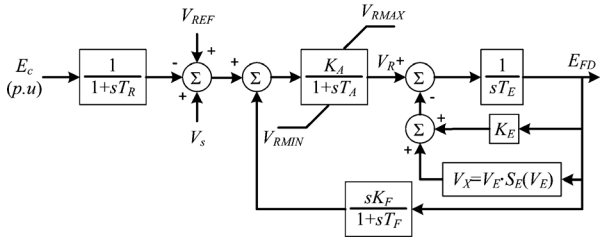

Fig. 4. Excitation model used in the cases.

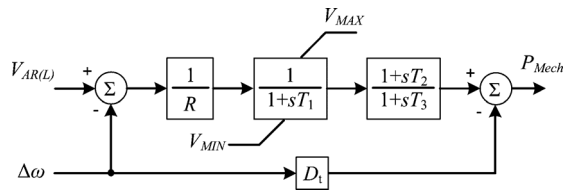

Fig. 5. Governor model used in the cases.

updates the matrices L and U generated from the previous factorization if the A-matrix elements do not change much. During extended periods of time between network switching events, when the DAE structure remains unchanged, the factorization with same pattern and same row is an attractive option. Option 4, no factorization, is used when the matrix A can be utilized repeatedly after a first factorization and so is selected when implementing the nonlinear solver VDHN.

VI. Simulation Results

In this section, the New England 39-bus system, an expanded New England 8775-bus system, and a modified PJM 13 029-bus system are simulated within HSET-TDS to compare the integration methods of Trapezoidal rule and HH4 with variable step control. The simulation is conducted on a Dell, OPTIPLEX 755, Intel Corel 2 Duo CPU E8400 @ 3.00 GHz 2.99 GHz, 3.25 GB of RAM. The computation environment is Visual C++ 2008 in Win7. The simulation results are validated using PSS\E version 30.3.2 [4].

The generator model, excitation model and governor model used in the simulation cases is GENROU [4] (a six-order model), IEEET1 [4] as shown in Fig. 4, and TGOV1 [4] as shown in Fig. 5. In the cases discussed in this section, the smallest time constant in the generator model is 0.02 [31], which is the subtransient time constant T_d" and T_q" for a steam unit. The smallest time constant in the excitation and governor models are 0.06 and 0.05, respectively. The following cases A and B focus on the short-term time-domain simulation, and case C focuses on the extended-term time-domain simulation.

A. Extended-Term Cascading Sequence on Expanded New England 8775-Bus System

The New England system with 10 generators and 39 buses is shown in Fig. 6(a). In order to test a large scale system, we constructed an 8775-bus test system based on the New England 39-bus system. The construction scheme is as follows:

1) Assume there are 15×15 (225) subsystems, each of which is identical to the original New England 39-bus system;

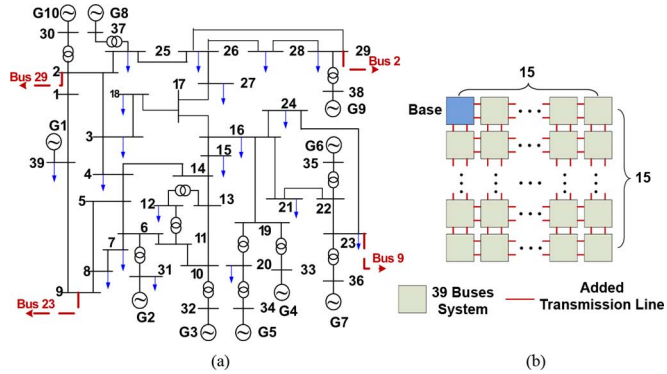

Fig. 6. Expanded system from New England 39-bus system (15×15). (a) Original 39-bus system. (b) Expanding mesh.

TABLE I
Events List of Cascading on 39-Bus System

Event	Start Time(s)	Events List
1	0.00	Start simulation
2	30.00	Bus fault on bus 17
3	30.10	The fault on bus 17 is removed
4	600.00	Fault line 17-18 (end of 17)
5	600.08	Line 17-18 is outaged
6	800.00	Fault line 26-25 (end of 26)
7	800.08	Line 26-25 is outaged
8	1000.00	Fault line 26-28 (end of 26)
9	1000.12	Line 26-28 is outaged
10	1100.00	Load on bus 26 is increased by 10%
11	1200.00	Terminate simulation

2) Buses 2, 9, 23, and 29 in each 39-bus subsystem are connected to adjacent subsystems as shown in Fig. 5(b); the impedances of connecting branches are small in magnitude;
3) The generators within each subsystem are modeled as previously described for the original 39-bus system.

To summarize the implementation HSET-TDS, it utilizes the direct solution method to construct and solve the DAE system. The DAE system includes 24 750 ODEs and 42 750 algebraic equations. The initial values of the DAE system are computed by the power flow program in HSET-TDS, which is based on the Newton-Raphson method. The variable step technique is used to compare the integration method of Trapezoidal rule and HH4. Explicit Adams 2 [6] and variable step explicit Adams 4 [24] are used for error estimation. VDHN is used for solving algebraic equations, and SuperLU is switched to the option of "no factorization" if the same Jacobian matrix can be used for Newton iteration. The option of "same-pattern-same-row" in SuperLU is used when the Jacobian matrix is updated.

Table I describes a cascading sequences of bus fault, line fault, line outage and increased load. The voltage of bus 37 is monitored. The system is first simulated in PSS/E with fixed integration step of 10^{-4} to provide a reference to validate results from HSET-TDS. The function of "PRNT" in PSSPLT of PSSE is used to obtain the simulation points (with precision of 10^{-4}) from PSSE. A 10 second part of the results from PSSPLT of PSS/E is shown in Figs. 7 and 8.[1] The corresponding curve simulated from HSET-TDS is shown in Fig. 8. If we define delta is the difference of the simulation result at one time point from

[1]Please check online version for colored figures of simulation results.

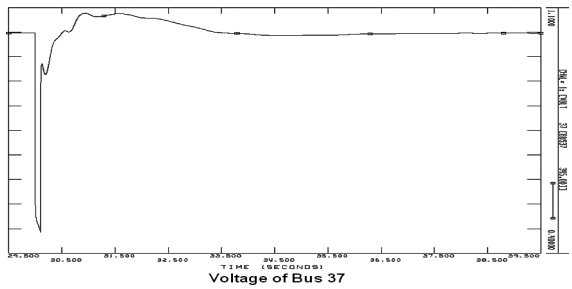

Fig. 7. Part of simulation results (29.5–39.5 s) of 8775-bus system from PSSPLT of PSS/E with integration step of 10^{-4}.

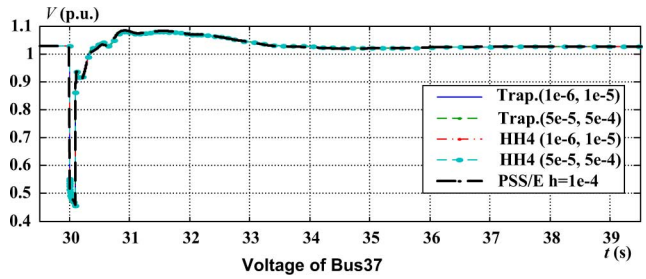

Fig. 8. Part of simulation results (29.5–39.5 s) of 8775-bus system from HSET-TDS.

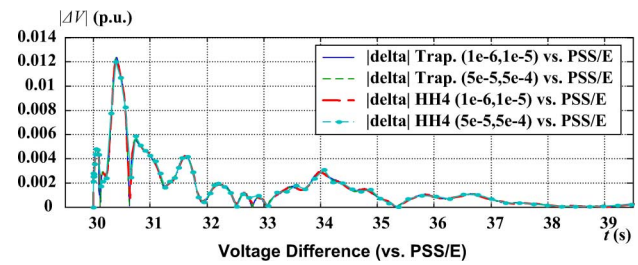

Fig. 9. Simulation results difference between HSET-TDS and PSS/E.

HSET-TDS and that from PSS/E, the abstract delta value versus time can be shown in Fig. 9. It can be seen that there is maximum delta value of 0.012 between the results of HSET-TDS and PSS/E. This error may be due to the different solvers inside the software or error of models. It also can be seen that for different error bounds, the results simulated by Trapezoidal rule and HH4 have the same trend, which makes the comparison feasible.

Fig. 11 plots the integration step and the number of VDHN iterations for the most (1e-6, 1e-5) and least (5e-5, 5e-4) precise simulations using both trapezoidal and HH4 integration. Although the number of VDHN iterations for HH4 are larger than those for trapezoidal in almost every integration step, the integration step for HH4 is generally much larger than that for trapezoidal. This is the reason why Table II shows that the total number of integration steps for HH4 is much smaller than that for trapezoidal and therefore HH4 uses less wall-clock time than trapezoidal for the same upper and lower error bounds and the same algebraic solver precision.

The simulation results from HSET-TDS are summarized in Table II and plotted in Fig. 10. Key information in Table II includes upper and lower error bounds, the maximum step size following the fault and number of simulation steps, and the simulation (wall-clock) time. Comparison of times for trapezoidal and HH4 integration for simulations having the same error bounds indicates HH4 outperforms trapezoidal by a factor of between 721/319=2.3 to 1811/471=3.8 where the larger improvement occurs for higher-precision simulations.

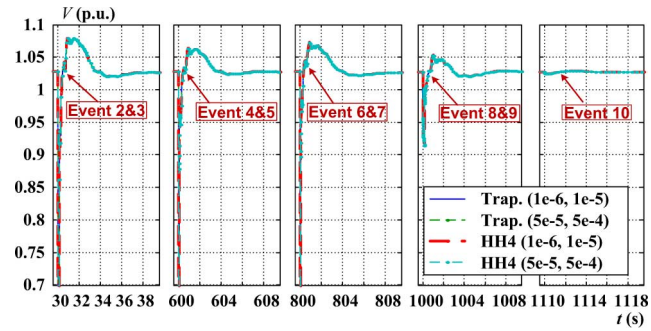

Fig. 10. Simulation results of 8775-bus system with variable integration step for 20 min in HSET-TDS.

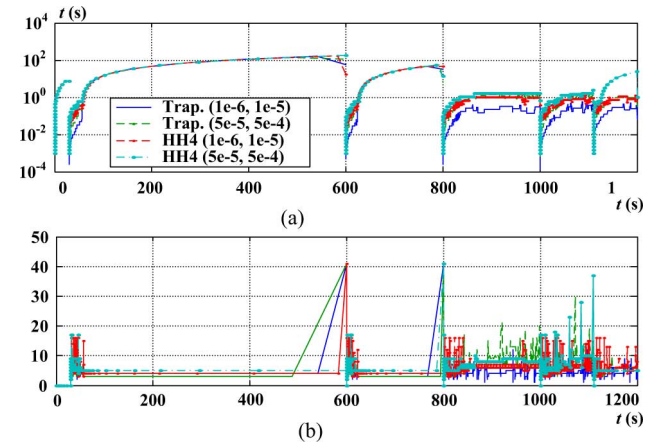

Fig. 11. Integration step and iteration times of VDHN for Trap. and HH4. (a) Integration step versus time. (b) Number of VDHN Iterations.

TABLE II
SIMULATION RESULTS OF 8775-BUS SYSTEM IN HSET-TDS

Integration Methods					Algebraic Solver		Linear Solver Library	Sim. Time (s)
Methods	Upper Error Bound	Lower Error Bound	Maximum Step (s)	Num. of Integration Steps	Alg. Solver	Precision		
Trap.	5×10^{-4}	5×10^{-5}	142.915	3872	VDHN	10^{-6}	SuperLU	721.468
Trap.	10^{-4}	10^{-5}	143.813	6393	VDHN	10^{-6}	SuperLU	926.086
Trap.	5×10^{-5}	5×10^{-6}	156.54	8070	VDHN	10^{-6}	SuperLU	1184.582
Trap.	10^{-5}	10^{-6}	161.79	14191	VDHN	10^{-6}	SuperLU	1811.488
HH4	5×10^{-4}	5×10^{-5}	153.574	819	VDHN	10^{-6}	SuperLU	318.988
HH4	10^{-4}	10^{-5}	172.771	964	VDHN	10^{-6}	SuperLU	384.868
HH4	5×10^{-5}	5×10^{-6}	173.997	1090	VDHN	10^{-6}	SuperLU	418.236
HH4	10^{-5}	10^{-6}	175.128	1380	VDHN	10^{-6}	SuperLU	471.292

B. Extended-Term of Cascading Sequence on Modified PJM System

A model with 13 029 buses, 431 generators, and 17 873 branches representing the PJM system was used to test HSET-TDS and compare the integration methods of trapezoidal rule and HH4. The dynamic models in the original dynamic file were replaced by models of GENROU, IEEET1, and TGOV1, with appropriate parameters. The cascading

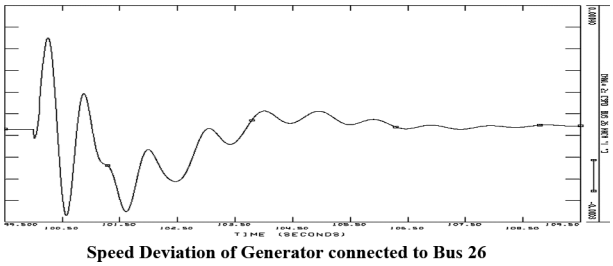

Fig. 12. Part of simulation results of PJM 13 029-bus system from PSSPLT of PSS/E with integration step of 10^{-4}.

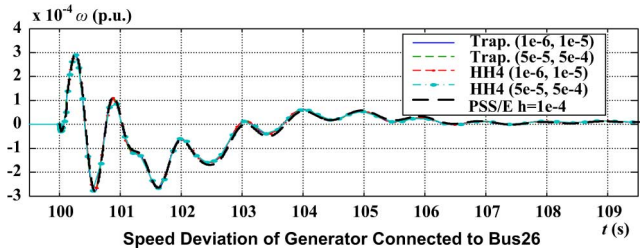

Fig. 13. Part of simulation results of PJM 13 029-bus system from HSET-TDS.

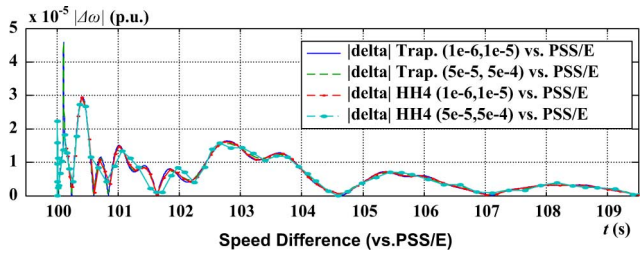

Fig. 14. Simulation results difference between HSET-TDS and PSS/E.

TABLE III
EVENTS LIST OF CASCADING ON 39-BUS SYSTEM

Event	Start Time(s)	Events List
1	0.00	Start simulation
2	100.00	Bus fault on bus 37"BRANCHBU"
3	100.10	The fault on bus 37 is removed
4	500.00	Bus fault on bus 79 "ESSEX"
5	500.08	The fault on bus 79 is removed
6	800.00	Fault line 204"SWATERFR"-171"NEWP PS" (end of 204)
7	800.08	Line 204-171 is outaged
8	840.00	Fault line 176"PENHOTAP"-23"BELLEVIL" (end of 176)
9	840.10	Line 176-23 is outaged
10	1100.00	Load on bus 125"LEONIA" is increased by 10%
11	1130.00	Load on bus 156"NBERGEN" is increased by 10%
12	1200.00	Terminate simulation

sequences of bus fault, line fault, line outage, and increased load are shown in Table III. The generator connected to bus 26 named "BERGEN" was monitored. A part of simulation results from PSS/E with fixed integration step of 10^{-4} are shown in Figs. 12 and 13, and the corresponding result from HSET-TDS are shown in Fig. 13. Fig. 14 shows the difference between the results from HSET-TDS and PSS\E. It can be seen that there is largest difference of 4.5×10^{-5} which is for the trapezoidal rule when upper and lower bounds are 5×10^{-4} and 5×10^{-5}.

We simulated the modified PJM system in HSET-TDS using trapezoidal and HH4 integration with variable step techniques. The DAE system includes 4741 ODEs and 41 837 algebraic equations. The configuration of the numerical solver (integration method, nonlinear solver, and linear solver) is the same as described in Section VI-A. The simulation results are summarized in Table IV and plotted in Fig. 15. Comparison of times for trapezoidal and HH4 integration for simulations having the same error bounds indicates HH4 outperforms trapezoidal by a factor of between $287/146 = 1.97$ to $862/262 = 3.29$, where the larger improvement generally occurs for higher-precision simulations, an effect also observed for the expanded New England system.

TABLE IV
SIMULATION RESULTS OF PJM 13 029-BUS SYSTEM IN HSET-TDS

Integration Methods					Algebraic Solver		Linear Solver Library	Sim. Time (s)
Methods	Upper Error Bound	Lower Error Bound	Maximum Step (s)	Num. of Integration Steps	Alg. Solver	Precision		
Trap.	5×10^{-4}	5×10^{-5}	102.383	2212	VDHN	10^{-6}	SuperLU	286.67
Trap.	10^{-4}	10^{-5}	103.545	3762	VDHN	10^{-6}	SuperLU	378.813
Trap.	5×10^{-5}	5×10^{-6}	104.383	4823	VDHN	10^{-6}	SuperLU	459.112
Trap.	10^{-5}	10^{-6}	116.752	9211	VDHN	10^{-6}	SuperLU	862.828
HH4	5×10^{-4}	5×10^{-5}	109.932	573	VDHN	10^{-6}	SuperLU	146.624
HH4	10^{-4}	10^{-5}	115.385	627	VDHN	10^{-6}	SuperLU	196.852
HH4	5×10^{-5}	5×10^{-6}	116.392	691	VDHN	10^{-6}	SuperLU	211.084
HH4	10^{-5}	10^{-6}	122.998	852	VDHN	10^{-6}	SuperLU	261.721

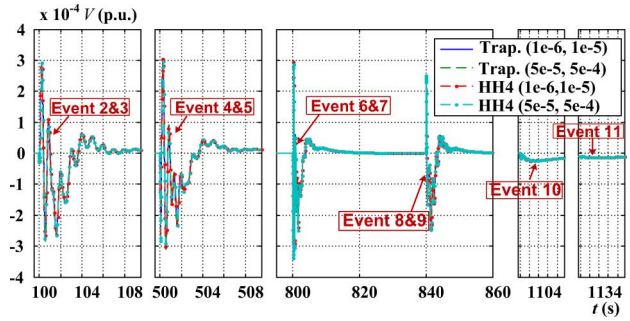

Fig. 15. Simulation results of PJM 13 029-bus system with variable integration step in HSET-TDS.

C. Extended-Term Simulation of Cascading Sequence on 39-Bus New England System

A sequence of events based on the 39-bus New England system shown in Fig. 6(a) is simulated in HSET-TDS to further test the efficiency of HH4 while illustrating the type of application for which its strengths are most salient, extended-term unfolding of cascading scenarios. The dynamic parameters used in the simulation are the same as those used in the simulations described in Section VI-A. The load and generation of the networks have been modified. The DAE system includes 110 ODEs and 190 algebraic equations.

As shown in Table V, a sequence of fault/line outage events follows an initiating fault/line outage event, a scenario which could occur as lines' load more heavily as a result of earlier outages. In addition, a corrective action (capacitor insertion) is included to prevent the system from becoming unstable. The voltage of bus 26 is monitored, and the simulation curve is shown in Fig. 17. The responses without the corrective action of inserting shunt capacitors (events 7 and 10) are also shown in Fig. 17.

The simulation results from HSET-TDS are shown in Table VI. Fig. 18 plots the integration step and number of

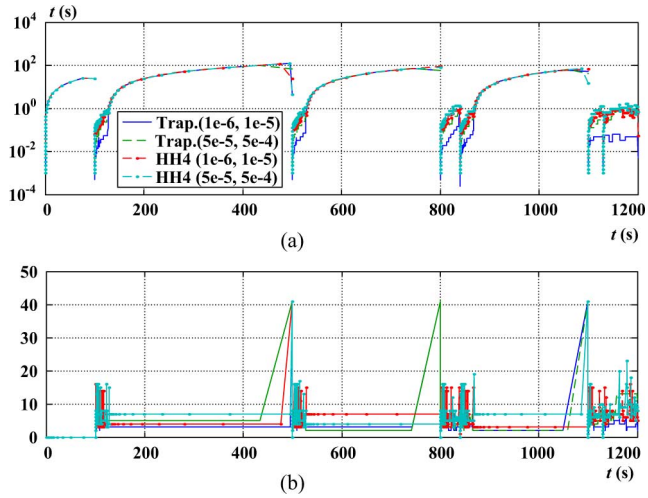

Fig. 16. Integration step and iteration times of VDHN for Trap. and HH4. (a) Integration step versus time. b) Number of VDHN iterations.

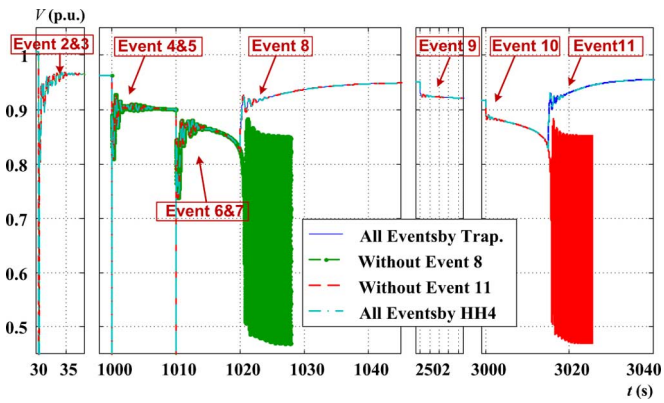

Fig. 17. Simulation results of cascading for 1 h in HSET-TDS.

TABLE V
EVENTS LIST OF CASCADING ON 39-BUS SYSTEM

Event	Start Time(s)	Events List
1	0.00	Start simulation
2	30.5	Fault line 25-26 (end of 25)
3	30.58	Line 25-26 is outaged
4	1000.0	Fault line 26-29 (end of 26)
5	1000.04	Line 26-29 is outaged
6	1010.0	Fault line 17-18 (end of 17)
7	1010.05	Line 17-18 is outaged
8	1020.0	Shunt Capacitor is inserted at bus 26
9	2500.0	Load on bus 26 is increased by 25%
10	3000.0	Load on bus 26 is increased by 25%
11	3015.0	Shunt Capacitor is inserted at bus 26
12	3600.0	Terminate simulation

algebraic solver's iterations for Trapezoidal rule and HH4. When one transient process is finished and the system stabilizes, the Jacobian matrix of nonlinear system may become ill-conditioned because some elements become very small in magnitude. This may increase the number of VDHN iterations. In address this, the method of Newton with SuperLU of "same-pattern-same-row" is used for the whole simulation. Fig. 18(b) shows that the iteration times of Newton is much smaller than that of VDHN shown in Figs. 11(b) and 16(b).

Additionally, when the system reaches the steady state, the integration step can be as large as the maximum integration step

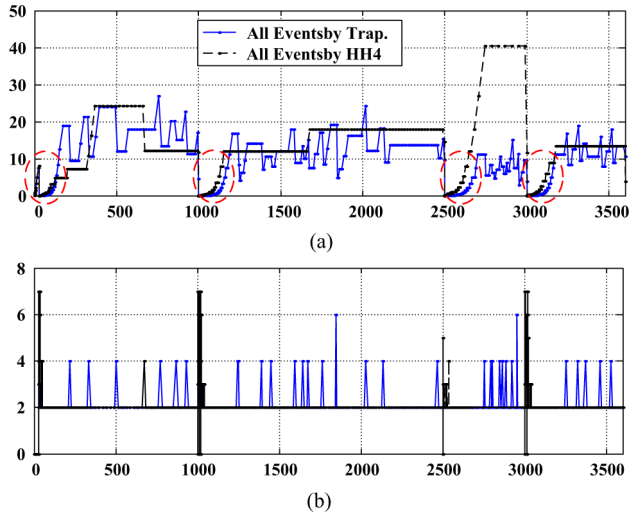

Fig. 18. Integration step and number of Newton iterations for Trapezoid rule and HH4 in simulation of cascading. (a) Integration step versus time. (b) Number of Newton iterations.

TABLE VI
SIMULATION RESULTS OF ALL SWITCHING EVENTS FOR 3600 s WITH VARIABLE INTEGRATION STEP

Integration Methods					Algebraic Solver		Linear Solver Library	Sim. Time (s)
Methods	Upper Error Bound	Lower Error Bound	Maximum Step (s)	Num. of Integration Steps	Alg. Solver	Precision		
Trap.	10^{-4}	10^{-5}	35.9533	12356	Newton	10^{-6}	SuperLU	10.192
HH4	10^{-4}	10^{-5}	40.4475	1273	Newton	10^{-6}	SuperLU	2.256

shown in Table VI, and less computation is needed. Most of the simulation time is spent on the transient process, where many iterations of the nonlinear solver are needed. The curves covered by circles with red dashed lines in Fig. 18(a) show that HH4 is able to enlarge the integration step when there is a transient process where much computation is needed. Due to decreased number of algebraic solver's iterations, larger integration step, and decreased number of integration steps, HH4 is significantly more efficient than trapezoidal as shown in Table VI.

VII. CONCLUSION

High-speed extended-term (HSET) time-domain simulator (TDS) has been developed within a direct solution method strategy using different integration methods, and various nonlinear solvers and linear solvers. We have employed HH4, VDHN, and SuperLU to accelerate computational speed. Results of simulation tests support the assertion that the integration method HH4 can be used to accelerate time-domain simulation especially for extended-term, because it can enlarge the integration step while still maintaining acceptable numerical precision.

REFERENCES

[1] B. Stott, "Power system dynamic response calculations," *Proc. IEEE*, vol. 67, pp. 219–240, Feb. 1979.
[2] H. W. Dommel and N. Sato, "Fast transient stability solutions," *IEEE Trans. Power App. Syst.*, vol. PAS-91, no. 4, pp. 1643–1650, 1972.

[3] F. De Mello, J. Feltes, and T. Laskowski, "Simulating fast and slow dynamic effects in power systems," *IEEE Comput. Appl. Power*, vol. 5, no. 3, pp. 33–38, Jul. 1992.
[4] Siemens, Power Transmission and Distribution, PSS/E TM 30.2 Online Documentation, 2005.
[5] EPRI EL 4610, Extended Transient Midterm Stability Program, 1987.
[6] J. Sanchez-Gasca, R. D'Aquila, and W. Price, "Variable time step, implicit integration for extended-term power system dynamic simulation," in *Proc. IEEE Power Industry Computer Application Conf. 1995*, May 7–12, 1995, pp. 183–189.
[7] J. Astic, A. Bihain, and M. Jerosolimski, "The mixed Adams-BDF variable step size algorithm to simulate transient and long term phenomena in power systems," *IEEE Trans. Power Syst.*, vol. 9, no. 2, pp. 929–935, May 1994.
[8] O. Fillatre, C. Evrard, and D. Paschini, "A powerful tool for dynamic simulation of unbalanced phenomena," in *Proc. 4th Int. Conf. Advances in Power System Control, Operation and Management, 1997 (APSCOM-97), Conf. Publ. No. 450*, Nov. 11–14, 1997, vol. 2, pp. 526–531.
[9] Power Factory/ DIgSILENT. [Online]. Available: http://www.digsilent.de/Software.
[10] General Electric International, Inc., PSLF Version 14.2 User's Manual, 2004.
[11] A. Zecevic and N. Gacic, "A partitioning algorithm for the parallel solution of differential-algebraic equations by waveform relaxation," *IEEE Trans. Circuits Syst. I, Fundam. Theory Appl.*, vol. 46, no. 4, pp. 421–434, Apr. 1999.
[12] E. Lelarasmee, A. Ruehli, and A. Sangiovanni-Vincentelli, "The waveform relaxation method for time-domain analysis of large-scale integrated circuits," *IEEE Trans. Comput.-Aided Des.*, vol. 1, pp. 131–145, Jul. 1982.
[13] [Online]. Available: http://home.gna.org/getfem/gmm_intro.html.
[14] [Online]. Available: http://www.cs.berkeley.edu/~demmel/SuperLU.html.
[15] [Online]. Available: http://www.cise.ufl.edu/research/sparse/umfpack.
[16] I. Decker, D. Falcao, and E. Kaszkurewicz, "Conjugate gradient methods for power system dynamic simulation on parallel computers," *IEEE Trans. Power Syst.*, vol. 11, no. 3, pp. 1218–1227, Aug. 1996.
[17] W. Xue, J. Shu, and Y. Wu, "Parallel algorithm and implementation for realtime dynamic simulation of power system," in *Proc. Int. Conf.n Parallel Processing, 2005 (ICPP 2005)*, Jun. 14–17, 2005, pp. 137–144.
[18] U.S.-Canada Power System Outage Task Force, Final Report on the Blackout in the United States and Canada: Causes and Recommendations, 2004.
[19] Q. Chen and J. McCalley, "Operational defense of power system cascading sequences," in *Proc. 2011 IEEE PES General Meeting*.
[20] S. Khaitan, C. Fu, and J. McCalley, "Fast parallelized algorithms for on-line extended-term dynamic cascading analysis," in *Proc. IEEE PES Power Systems Conf. Expo.*, Seattle, WA, Mar. 15–18, 2009.
[21] G. Gross and A. Bergen, "An efficient algorithm for simulation on transients in large power systems," *IEEE Trans. Circuits Syst.*, vol. 23, no. 12, pp. 791–799, Dec. 1976.
[22] D. Yang and V. Ajjarapu, "A decoupled time-domain simulation method via invariant subspace partition for power system analysis," *IEEE Trans. Power Syst.*, vol. 21, no. 1, pp. 11–18, Feb. 2006.
[23] E. Hairer and G. Wanner, *Solving Ordinary Differential Equations II: Stiff and Differential-Algebraic Problems*. New York: Springer-Verlag, 1996.
[24] E. Hairer, S. Norsett, and G. Wanner, *Solving Ordinary Differential Equations I: Nonstiff Problems*. New York: Springer-Verlag, 1987.
[25] P. Hammer and J. Hollingsworth, "Trapezoidal methods of approximating solutions of differential equations," *MATC*, vol. 9, pp. 92–96, 1959.
[26] F. L. Alvarado, R. H. Lasseter, and J. J. Sanchez, "Testing of trapezoidal integration with damping for the solution of power transient problems," *IEEE Trans. Power App. Syst.*, vol. PAS-102, no. 12, pp. 3783–3790, Dec. 1983.
[27] J. S. Chai, N. Zhu, and A. Bose, "Parallel Newton type methods for power system stability analysis using local and shared memory multiprocessors," *IEEE Trans. Power Syst.*, vol. 6, no. 4, pp. 1539–1545, Nov. 1991.
[28] J. W. Demmel, J. R. Gilbert, and X. S. Li, SuperLU Users' Guide, Nov. 2007. [Online]. Available: http://www.cs.berkeley.edu~demmel/SuperLU.html.
[29] J. W. Demmel, S. C. Eisenstat, and J. R. Gilbert, "A supernodal approach to sparse partial pivoting," *SIAM J. Matrix Anal. Appl.*, vol. 20–3, pp. 720–755, 1999.
[30] X. Li and J. Demmel, "Making sparse Gaussian elimination scalable by static pivoting," in *Proc. High Performance Networking and Computing Confe.*, Orlando, FL, Nov. 7–13, 1998.
[31] P. Kundur, *Power System Stability and Control*. New York: McGraw-Hill, 1994.

Chuan Fu (S'07) received the B.S. degree and M.S. degree of science from Tianjin University, Tianjin, China, in 2003 and 2006, respectively. He is currently pursuing the Ph.D. from the Department of Electrical and Computer Engineering, Iowa State University, Ames.

James D. McCalley (F'04) received the B.S., M.S., and Ph.D. degrees in electrical engineering from Georgia Institute of Technology, Atlanta, in 1982, 1986, and 1992, respectively.

He was employed with Pacific Gas & Electric Company, San Francisco, CA, from 1985 to 1990 as a Transmission Planning Engineer. He is now a Professor of Electrical and Computer Engineering at Iowa State University, Ames, where he has been employed since 1992.

Dr. McCalley is a registered professional engineer in California.

Jianzhong Tong (SM'98) received the Ph.D. degree in electrical engineering from Zhejiang University, Hangzhou, China, in 1987

He is currently a Senior Strategist at PJM Interconnection, Norristown, PA. He was a post doctoral fellow at Cornell University, Ithaca, NY, from 1990 to 1993, an Assistant Professor at Zhejiang University in 1988, an R&D engineer at Siemens Power Systems Controls from 1993 to 1996, and a principal engineer at OATI in 1997. His research interests include power system stability, voltage/Var controls, smart grid of control center, etc. He has led the building of online voltage stability and transient stability tools in the PJM control room for real-time operations.