01 May 2008

A Variable Partitioning Strategy for the Multirate Method in Power Systems

Jingjia Chen

Mariesa Crow
Missouri University of Science and Technology, crow@mst.edu

Recommended Citation

J. Chen and M. Crow, "A Variable Partitioning Strategy for the Multirate Method in Power Systems," *IEEE Transactions on Power Systems*, vol. 23, no. 2, pp. 259-266, Institute of Electrical and Electronics Engineers (IEEE), May 2008.
The definitive version is available at https://doi.org/10.1109/TPWRS.2008.919319

A Variable Partitioning Strategy for the Multirate Method in Power Systems

Jingjia Chen, *Student Member, IEEE*, and Mariesa L. Crow, *Senior Member, IEEE*

***Abstract*—An application of the multirate method to differential and algebraic equations (DAEs) is discussed in this paper. Based on the specific formulation of the power systems algebraic equations, an effective algebraic variable partitioning strategy is developed and implemented. The application of the partitioning strategy to an example power system provides validation of the proposed partitioning strategy in terms of speed-up and accuracy.**

***Index Terms*—Differential and algebraic equations (DAEs), dynamic security assessment, multirate method, numerical integration, power systems simulation.**

I. Introduction

TIME domain simulation is a crucial application for the dynamic security assessment of power systems. The computational complexity of time domain simulation has kept it from being widely used in online decision making. If, for example, a transient stability analysis could be run in real-time, then power system operators could implement online control to avoid cascading failures. The time domain simulation tool could assist the operator with proactive measures to limit the extent of the incident, which could significantly improve power system reliability [1].

The comprehensive dynamic simulation of a power system involves time scales that ranges from seconds to minutes to even hours; therefore, it is necessary to combine short-term and long-term analysis in a single program [2]. In recent years, considerable effort has been focused in this direction [3]–[5]. Traditional power system simulation methods focus on fixed or variable step methods which are suitable for the simulation of systems that exhibit infrequent fast decaying transients. When integrating systems of differential equations whose component dynamics persist at different time scales, it is preferable to avoid unnecessary calculations on slowly changing solution components.

For power systems, the existence of power-electronic-based devices, such as FACTS and HVDC components, and dynamic loads, such as induction motors, increases the range of the time scales. In typical transients, only a small fraction of the variables in the system exhibit fast dynamics; therefore, it is inefficient to simulate the entire power system with a small integration time step when most variables react slowly and accuracy constraints can be easily satisfied with a larger step size. Therefore, when there are sustained mid- to high-frequency dynamics in the system, conventional time domain simulation methods are inefficient.

Multirate methods were first proposed by Gear [6]. The multirate method is an integration method typically applied to ordinary differential equations (ODEs). Multirate methods are best suited for systems with widely ranging time response behavior. The basic principle of multirate methods is the integration of different variables with different step sizes which are necessary and sufficient for the accuracy required. The coupling between the slow and fast variables is realized by interpolation and extrapolation. Since there is no need to integrate the whole system with a small step size in the multirate method, considerable computational speed-up can be achieved, especially for large systems with very few fast variables.

Multirate methods have been applied to electric circuit analysis in [7]. The application of multirate methods to power system dynamics was first introduced in [8] and further developed in [9]. More recently, multirate methods have been applied to power electronics-based systems [10]. The multirate method has also been applied to mechanical systems [11], which also featured DAEs.

The multirate method works most effectively on systems that are properly partitioned into fast and slow subsystems. In [10], high speed-ups were obtained, but the power electronics-based system lent itself to easy partitioning. Several partitioning strategies have been proposed for ODE systems based on the rate of change of the state variables [9]. The local truncation error of the multirate method is discussed in [12]. Partitioning for DAE systems is more complex, since the algebraic variables do not have an inherent rate of change associated with them. Good partitioning strategies for DAE system multirate methods do not exist. A poor partitioning of the algebraic variables can adversely affect both computational speed and accuracy of the method.

In this paper, a partitioning strategy for the application of the multirate method to power system DAEs is presented. Since in power systems there are far more algebraic variables (mostly voltage angles and magnitudes) than state variables (mostly generators or other dynamics components), a proper algebraic variable partitioning is critical for significant speed-up and to maintain numerical accuracy. Based on the analysis of the specific form of the power system algebraic equations, an effective algebraic variable partitioning strategy is developed and implemented. The application of the proposed partitioning strategy

Manuscript received January 10, 2006; revised September 27, 2007. This work was supported in part by the National Science Foundation under Grant CNS-0420869 and in part by the DOE Energy Storage Program through Sandia National Laboratories under BD-0071-D. Paper no. TPWRS-00002-2006.

The authors are with the Department of Electrical and Computer Engineering, Missouri University of Science and Technology, Rolla, MO 65409 USA (e-mail: crow@mst.edu).

Digital Object Identifier 10.1109/TPWRS.2008.919319

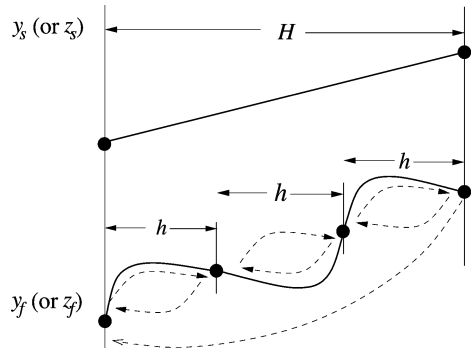

Fig. 1. Multirate implementation.

to an example power system is used to compare computational speed and efficiency.

II. Multirate Method

In the multirate method, different integration steps are used for the fast variables and slow variables. A micro step (h) is used for the fast variables and a macro step (H) is used for the slow variables. If the ratio of the macro step to the micro step is an integer m, then $H = mh$.

In a linear system, it is possible to find a matrix M which relates $y(t+ih)$ to $y(t)$ for both fast and slow variables [9]. This type of closed form relationship is not possible in a nonlinear system. Therefore, in the nonlinear case, iterative predictor-corrector methods are required to solve the nonlinear system.

At each macro step, the slow variables are first predicted for the next macro step and then interpolated to provide approximations for each micro step interval. Typically, a linear interpolation is used, but other interpolations may also be used. The choice of interpolation method affects both computational speed and numerical accuracy. Once the slow variables are interpolated at each fast interval, the fast variables can be found by numerical integration at each micro step using the interpolated slow variables as needed. The entire system is solved at each macro step. The updated values of the slow variables are compared to the predicted values. If they are within the specified tolerance, the time step is advanced to the next time interval; otherwise, the step is repeated using the updated values to provide better interpolated values. This process is illustrated in Fig. 1.

Note that the coupling from the fast subsystem to the slow subsystem is necessarily weak. From a physical standpoint, if the coupling were not weak, then the fast dynamics would be apparent in the slow subsystem; but, by definition, the slow subsystem does not have any fast dynamics—therefore, the coupling from the fast to the slow is weak. The converse, however, does not have to hold: the coupling from the slow to the fast may be either weak or strong.

III. Error and Selection of Step Ratio m

There are many classes of methods such as Adams Bashforth, Adams Moulton, BDF, and Gear's methods that fall in the family of multistep methods [6]. Any of these methods can be used as the underlying integration algorithm for a multirate method. In this paper, the well-known second-order Adams Bashforth method, also known as the trapezoidal method, is used. The trapezoidal method is given by

$$x_{n+1} = x_n + \frac{h}{2}(f(x_{n+1}, t_{n+1}) + f(x_n, t_n)). \tag{1}$$

Consider a system of differential equations that have been partitioned into fast and slow subsystems as follows:

$$\dot{x} = f(x, y, t) \tag{2}$$

$$\dot{y} = g(x, y, t) \tag{3}$$

where $x \in R^{n_f}$ denotes the vector of fast variables and $y \in R^{n_s}$ denotes the vector of slow variables. In the multirate method, the fast subsystem will be integrated with timestep h

$$x_{n+i} = x_{n+i-1} + \frac{h}{2}(f(x_{n+i}, \hat{y}_{n+i}) + f(x_{n+i-1}, \hat{y}_{n+i-1})) \tag{4}$$

for $i = 1, \ldots, m$ and $\hat{y}_{n+i}$ and $\hat{y}_{n+i-1}$ are interpolated values. The slow subsystem is integrated with timestep $H = mh$

$$y_{n+m} = y_n + \frac{mh}{2}(f(x_{n+m}, y_{n+m}) + f(x_n, y_n)). \tag{5}$$

For multistep methods, a generalized expression for the local truncation error has been developed [13]. The local truncation error for the trapezoidal method is given by

$$\varepsilon_x \approx \frac{1}{12}\frac{d^3x(t)}{dt^3}h^3. \tag{6}$$

Therefore, the error in x at time t_{n+m} is

$$\varepsilon_x \approx m\frac{1}{12}\frac{d^3x(t)}{dt^3}(h)^3 \tag{7}$$

and the error in y at time t_{n+m} is

$$\varepsilon_y \approx \frac{1}{12}\frac{d^3y(t)}{dt^3}(mh)^3. \tag{8}$$

During implementation, the ratio m should be chosen such that the level of error in both x and y are on the same order. Note that with a single rate method, the error in the slow subsystem would be much smaller than the error in the fast subsystem. The advantage of the multirate method is that much larger stepsizes in the slow subsystem can be taken for the same level of error. Therefore, if the interpolation error is neglected, then m can be chosen such that

$$m^2 = \frac{\frac{d^3x}{dt^3}}{\frac{d^3y}{dt^3}}. \tag{9}$$

This is an easily implementable method of determining m and adaptively changing it throughout the simulation. The required derivatives for the LTE can be approximated using backward's

difference methods [13]

$$\dot{x}_{n+1} \approx \nabla_{1,n+1} = \frac{x_{n+1} - x_n}{t_{n+1} - t_n} \tag{10}$$

$$\dot{x}_n \approx \nabla_{1,n} = \frac{x_n - x_{n-1}}{t_n - t_{n-1}} \tag{11}$$

$$\dot{x}_{n-1} \approx \nabla_{1,n-1} = \frac{x_{n-1} - x_{n-2}}{t_{n-1} - t_{n-2}} \tag{12}$$

$$\ddot{x}_{n+1} \approx \nabla_{2,n+1} = \frac{\nabla_{1,n+1} - \nabla_{1,n}}{t_{n+1} - t_{n-1}} \tag{13}$$

$$\ddot{x}_n \approx \nabla_{2,n} = \frac{\nabla_{1,n} - \nabla_{1,n-1}}{t_n - t_{n-2}} \tag{14}$$

$$\frac{d^3x}{dt^3} \approx \nabla_{3,n+1} = \frac{\nabla_{2,n+1} - \nabla_{2,n}}{t_{n+1} - t_{n-2}} = \frac{\frac{x_{n+1}-x_n}{t_{n+1}-tn} - \frac{x_n - x_{n-1}}{t_n - t-n-1}}{t_{n+1} - t_n} - \frac{\frac{x_n - x_{n-1}}{t_n - t_{n-1}} - \frac{x_{n-1} - x_{n-2}}{t_{n-1} - t_{n-2}}}{t_n - t_{n-2}}}{t_{n+1} - t_{n-2}}. \tag{15}$$

If the integration timestep h is fixed, then (15) can be simplified to

$$\frac{d^3x}{dt^3} \approx \frac{x_{n+1} - 3x_n + 3x_{n-1} - x_{n-2}}{6h^3}. \tag{16}$$

This approximation can be obtained from previously calculated values for x_n, x_{n-1}, and x_{n-2} without significant additional computation.

Multistep methods are fit to a polynomial of degree k and are exact if the multistep coefficients are selected correctly [13]. Therefore, if the interpolation method is also polynomial of degree k, then the interpolated values will match the integration values and the interpolation error will be zero. Therefore, if a lower order interpolation is chosen (such as a linear interpolation which is a degree 1 polynomial), the interpolation error is bounded and small. Furthermore, (9) yields a real value for m. If the ratio is chosen as the lower integer value $m_{\text{int}} \leq m$, then this is a conservative value and the interpolation errors can be safely neglected.

IV. Multirate Method Applications to DAEs

The multirate method has been widely applied to system of ODEs but has only infrequently been used for DAE systems. In this section, the multirate method is fully extended to DAEs, and the partitioning of the algebraic variables will be discussed in the next section.

Consider the two-time-scale initial value problem of nonlinear DAEs as follows:

$$\dot{x}(t) = f(x, y, z), \quad x(t_a) = x_a \tag{17}$$

$$\dot{y}(t) = g(x, y, z), \quad y(t_a) = y_a \tag{18}$$

$$0 = h(x, y, z) \tag{19}$$

where $t \in [t_a, t_b], x \in R^{n_x}, y \in R^{n_y}$ and $z \in R^{n_z}$. In this work, it is assumed that $h(x, y, z)$ is continuously differentiable with respect to z. In power systems, the variable set $[x, y]$ contains the dynamic states that correspond to generators, FACTS devices, dynamic loads, etc. The variable set z contains the network states that correspond to bus voltages and angles.

The algebraic variables can be divided into two groups associated with the fast and slow variables. Note, however, that algebraic variables are neither "fast" nor "slow" since they do not exhibit independent dynamics but are simply called fast or slow by how they are partitioned. Rather than considering algebraic equations partitions as segregating the algebraic states into "fast" and "slow" groups, consider the partitions as dictating how often the algebraic states must be updated to sufficiently propagate their information to the system. A "slow" algebraic variable is updated only when the slow states are updated—a "fast" algebraic variable is updated when the fast states are updated. The goal of the partitioning is to find the smallest set of algebraic states that require frequent updating to meet the accuracy requirements of the simulation.

The fast algebraic variables $z_x(t)$ and the slow algebraic variables $z_y(t)$ are given by

$$z(t) = \begin{pmatrix} z_x(t) \\ z_y(t) \end{pmatrix} \tag{20}$$

where $z_x \in R^{n_{z,x}}, z_y \in R^{n_{z,y}}$ and $n_{z,x} + n_{z,y} = n_z$.

Therefore, the original DAE system can be written as

$$\dot{x}(t) = f(x, y, z_x, z_y), \quad x(t_a) = x_a \tag{21}$$

$$\dot{y}(t) = g(x, y, z_x, z_y), \quad y(t_a) = y_a \tag{22}$$

$$0 = h_x(x, y, z_x, z_y) \tag{23}$$

$$0 = h_y(x, y, z_x, z_y). \tag{24}$$

This leads to the following algorithm.

Multirate Algorithm

1) Predict the slow state variables at time $t + H$. In this paper, a first-order predictor is used, but higher-order predictors may also be used

$$y^p_{t+H} = y_t + H\dot{y}_t \tag{25}$$

where the superscript p refers to the predicted value.

2) Predict the slow algebraic variables at time $t + H$

$$z^p_{y,t+H} = z_{y,t} + (z_{y,t} - z_{y,t-H}). \tag{26}$$

This is a linear extrapolation for z_y.

3) Integrate the fast components for $i = 1, 2, \ldots, m - 1$

$$\dot{x}(t + ih) = f(\hat{y}(t + ih), x(t + ih), \hat{z}_y(t + ih), z_x(t + ih)) \tag{27}$$

$$0 = h_x(\hat{y}(t + ih), x(t + ih), \hat{z}_y(t + ih), z_x(t + ih)) \tag{28}$$

with integration time step h, where $\hat{y}(t+ih)$ and $\hat{z}(t+ih)$ are the interpolated values of y and z_y at $(t+ih)$. For linear interpolation

$$\hat{y}(t + ih) = \frac{i}{m}\left(y^p_{t+H} - y_t\right) + y_t \tag{29}$$

and

$$\hat{z}_y(t + ih) = \frac{i}{m}\left(z^p_{y,t+H} - z_{y,t}\right) + z_{y,t}. \tag{30}$$

Note that since (27) and (28) are nonlinear functions, if an implicit numerical integration method is used, then the discretized equation must be solved iteratively using a Newton-type nonlinear solver.

4) Integrate both fast and slow components at the macro time step (which is the same time as the final micro step). Note that the fast subsystem will be integrated with integration time step h and the slow subsystem will be integrated with time step $mh(=H)$.
5) Compare the calculated slow value with the predicted value. If $|y^p_{t+H} - y_{t+H}| > \epsilon$, set $y^p_{t+H} = y_{t+H}$; otherwise, set $t = t + H$.
6) Go to step 1.

This algorithm is described using a fixed h and m, but in practical implementation, both can vary according to user specifications.

V. Algebraic Variable Partitioning Strategy

There are two types of variables in DAEs: state variables and algebraic variables. In power systems, the state variables are from the differential equations, which are usually generator models, controllers, or dynamic loads. The algebraic variables are usually the bus voltage magnitude and phase angles, but they may also include other system variables, depending on the models and components used.

Typically, a power system has a large number of buses, and the number of network buses far exceeds the number of generator buses; therefore, there will be a greater number of algebraic variables than state variables. Since the speed-up provided by the multirate method is directly related to the ratio of the number of fast variables to the number of slow variables, the greater the number of slow variables (including algebraic variables), the greater the speed-up.

The local truncation error (LTE) is typically used as a measure by which the state variables are partitioned into slow and fast subsets [8]. A large LTE indicates that the state variable is varying rapidly and is therefore a fast variable. Algebraic variables, however, do not have an LTE associated with them; therefore, there is no straightforward indicator which predicts whether or not they should be taken as fast, or slow, variables.

The objective of this section is to find a method to separate the algebraic variables into two groups: the "fast" algebraic variables, which require an update every micro integration step, and the "slow" algebraic variables, which require an update every macro integration step.

Power system dynamics are in the form (17)–(19). To simulate the system, the differential equations are discretized using the chosen integration method. Using the trapezoidal integration method, which is commonly adopted in power system simulation, the discretized system can be written as

$$y_{n+1} = y_n + \frac{h}{2}[f(y_{n+1}, z_{n+1}) + f(y_n, z_n)]$$
$$0 = g(y_{n+1}, z_{n+1})$$

where h is the integration time step and y_n and z_n are the converged values from the previous time step. The original DAEs can then be transformed into a set of nonlinear algebraic equations

$$F(y_{n+1}, z_{n+1}) = 0 \tag{31}$$
$$g(y_{n+1}, z_{n+1}) = 0 \tag{32}$$

where

$$F(y_{n+1}, z_{n+1}) = y_{n+1} - y_n - \frac{h}{2}[f(y_{n+1}, z_{n+1}) + f(y_n, z_n)].$$

Since these nonlinear equations are implicit in y_{n+1} and z_{n+1}, they must be solved numerically using Newton's method to iteratively solve the nonlinear algebraic equations

$$\left[J^{(k)}\right] \begin{bmatrix} y^{(k+1)}_{n+1} - y^{(k)}_{n+1} \\ z^{(k+1)}_{n+1} - z^{(k)}_{n+1} \end{bmatrix} = - \begin{bmatrix} F^{(k)} \\ g^{(k)} \end{bmatrix}$$

where $J^{(k)}$ is the Jacobian of (31) and (32) and k is the Newton iteration index. The Newton iteration stops when $\max[\|F^{(k)}\|, \|g^{(k)}\|] < \epsilon$, where ϵ is a user-defined tolerance.

To develop a partitioning scheme for the algebraic variables of DAEs, the algebraic equations are used. Consider first that not all algebraic equations converge simultaneously. This can be verified by comparing the power mismatches of different buses at each iteration. Typically: 1) some mismatches have already converged; 2) most mismatches are decaying fast enough to converge with a constant Jacobian within a few iterations; and 3) a few identifiable mismatches will converge fast only if the corresponding part (rows) of the Jacobian are updated [14]. There is no need to update the entire Jacobian at every step. This quality is frequently referred to as numerical *latency*. By applying the multirate method, only a subset of the algebraic variables are updated and, consequently, the related Jacobian elements. It can be therefore concluded that fast-changing variables will result in larger mismatch components.

Thus, it is reasonable to attempt to partition the algebraic variables by considering the bus mismatches. The proposed algebraic variable partitioning strategy is as follows.

At time t

1) Calculate the power mismatches of the second Newton iteration: $[\Delta Q_i(t), \Delta P_i(t)]$.
2) Calculate
$$\Delta P^* = \frac{1}{n}\sum_{i=1}^{n} \Delta P_i(t) \tag{33}$$
$$\Delta Q^* = \frac{1}{n}\sum_{i=1}^{n} \Delta Q_i(t). \tag{34}$$
3) Calculate
$$S^2 = \frac{1}{n}\sum_{i=1}^{n} (\Delta P_i(t) - \Delta P^*)^2 + (\Delta Q_i(t) - \Delta Q^*)^2.$$
4) If $\Delta P_i(t) > \Delta P^* + KS$ or $\Delta Q_i(t) > \Delta Q^* + KS$, then bus i is regarded as "fast," and the bus voltage and angle are added as fast algebraic variables.
5) Integrate the differential equations with the "fast" algebraic constraint(s) until $t + (m-1)h$.
6) Integrate both fast and slow steps.
7) $t = t + H$, go to step 1.

The constant K is a user-defined non-negative constant and varies by system. Initial analyses for partitioning may simply assume that $K = 0$ and refine it adaptively for improved performance. The larger K is chosen, the fewer states will be considered as "fast." A coarse "rule of thumb" is to choose K between 0.05 and 0.1.

While the formulation presented is developed using power mismatch equations, the method is also valid for current injection formulations as well since it identifies fast buses and not individual variables.

VI. Implementation Issues

The primary focus of the multirate algorithm is to develop a numerical algorithm which is superior in computational speed to traditional methods. Thus, it is imperative that all steps throughout the implementation of the algorithm be as computationally efficient as possible, without destroying the underlying convergence rate. There are several specific implementation issues that must be addressed to effectively utilize the multirate method.

A. Adaptive Partitioning

It is possible, and even desirable, to modify the fast and slow partitions adaptively throughout the simulation interval. As different states become active or decay, they should be moved up or down in the partitions. The basic adaptive partitioning strategy is to move any variables which have significantly decayed *in comparison to the other variables* to the slow partition and move any variables which exhibit continued rapid or fast behavior to the fast partition. Thus, the strategy must encompass a means by which the relative behavior of each state can be assessed within the context of the larger system response. The following simple approach can be used to adjust the states in the fast and slow partitions.

1) At the conclusion of each macro time step, calculate the following:

$$\mathrm{AVG}_f = \left(\prod_{i \in \mathrm{FAST}} \mathrm{LTE}(i) \right)^{\frac{1}{n_f}}$$

$$\mathrm{MAX}_f = \frac{\max_{i \in \mathrm{FAST}}(\mathrm{LTE}(i))}{\mathrm{AVG}_f}$$

$$\mathrm{MIN}_f = \frac{\min_{i \in \mathrm{FAST}}(\mathrm{LTE}(i))}{\mathrm{AVG}_f}$$

$$\mathrm{AVG}_s = \left(\prod_{i \in SLOW} \mathrm{LTE}(i) \right)^{\frac{1}{n_s}}$$

$$\mathrm{MAX}_s = \frac{\max_{i \in SLOW}(\mathrm{LTE}(i))}{\mathrm{AVG}_s}$$

$$\mathrm{MIN}_s = \frac{\min_{i \in SLOW}(\mathrm{LTE}(i))}{\mathrm{AVG}_s}$$

where n_f and n_s are the number of states in the FAST and SLOW partitions,respectively.

2) Move state i from SLOW to FAST if

$$\frac{\mathrm{MIN}_s}{\mathrm{MAX}_s} < m^3 \quad \text{and} \quad \frac{\mathrm{LTE}(i)}{\mathrm{AVG}_s} > m^2.$$

3) Move state i from FAST to SLOW if

$$\frac{\mathrm{MIN}_s}{\mathrm{MAX}_s} < m^3 \quad \text{and} \quad \frac{\mathrm{LTE}(i)}{\mathrm{AVG}_s} < \frac{1}{m^2}.$$

The value $\mathrm{AVG}_f(\mathrm{AVG}_s)$ is the geometric mean of the LTEs of the FAST (SLOW) partition. The values $\mathrm{MAX}_f(\mathrm{MAX}_s)$ and $\mathrm{MIN}_f(\mathrm{MIN}_s)$ are the normalized maximum and minimum LTEs in the FAST (SLOW) partition. The first conditions in steps 2 and 3 of the algorithm test for a *spread* of time-scale behavior within the group. The second condition attempts to determine if there are distinct separations in the LTEs of the states. If there is a continuum of LTEs, then it can be argued that no states should be moved up or down, but if distinct gaps exist, then states should be shuffled from one partition to another.

Lastly, a new m can be calculated from

$$m^2 = \frac{\mathrm{AVG}_f}{\mathrm{AVG}_s}$$

which is consistent with (9).

B. Sparsity

At each macro step, the entire system is solved; therefore, the Jacobian sparsity is unaffected. For the multirate method to be computationally effective, the premise is that the number of fast variables is small compared to the whole system; therefore, the fast system Jacobian will also be small and relatively full (not sparse); therefore, we did not attempt to use sparse techniques on the fast/small subsystem, only the larger full system.

C. Newton–Raphson Iterations

Fig. 1 implies that at each micro-step, Newton–Raphson iterations are performed until the fast equations are solved and then the solution is advanced to the next time step, where the process repeats. This continues until the macro-step is reached. If the slow states have not yet converged, the macro-step is repeated. Strict adherence to this approach requires numerous micro-step iterations, even in the early macro-step loops when the guess at $t + mh$ is quite coarse. Recall, however, that during the first few macro-step iterations, the slow variables may not be very accurate; thus, it is not computationally efficient to force the fast variables to within the convergence tolerance. Similarly, it is computationally inefficient to drive the slow subsystem to convergence at the macro-step if the fast system is not accurate.

A more efficient approach is to perform only **one** Newton–Raphson iteration at each micro-step and then again at the macro-step. If the fast and slow subsystems both converge with quadratic convergence (which they will if an accurate predictor is used), then both subsystems will converge simultaneously within two to three Newton–Raphson iterations. This approach eliminates the computation of $2(m-1)$ to $6(m-1)$ fast subsystem solutions and two to six slow subsystem solutions (depending on whether two or three Newtwon–Raphson iterations would normally be required) and can significantly reduce the computational effort required. The accuracy is maintained since all variables, including the fast variables, must converge within the specified tolerance throughout the whole macro-step interval.

VII. Power System Example

To illustrate the efficacy of the multirate method and to explore the impact of various partitionings, the IEEE 118-bus test system shown in Fig. 2 will be used. The system was modified to include a STATCOM at bus 34, and induction motors at buses 19, 24, and 35. The system dynamics are due to a three-phase fault at bus 15 applied shortly after 0 s and cleared at 0.05 s. The voltage at bus 34 with and without STATCOM control is shown

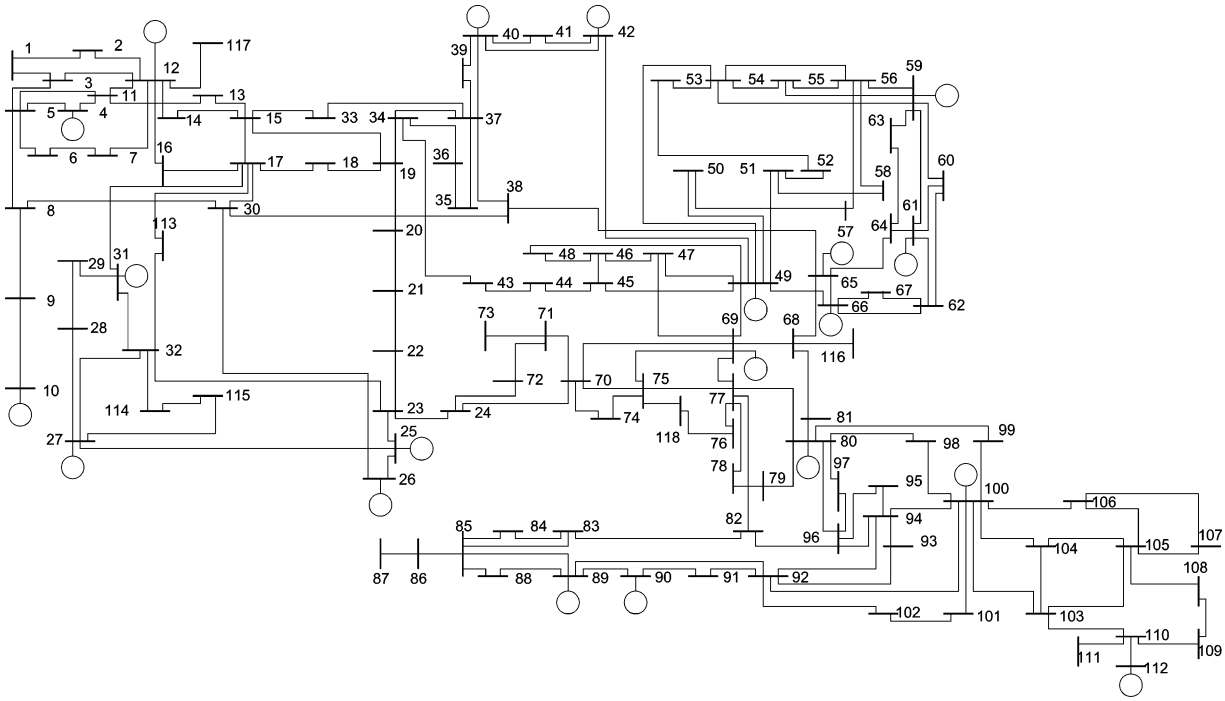

Fig. 2. IEEE 118-bus test system.

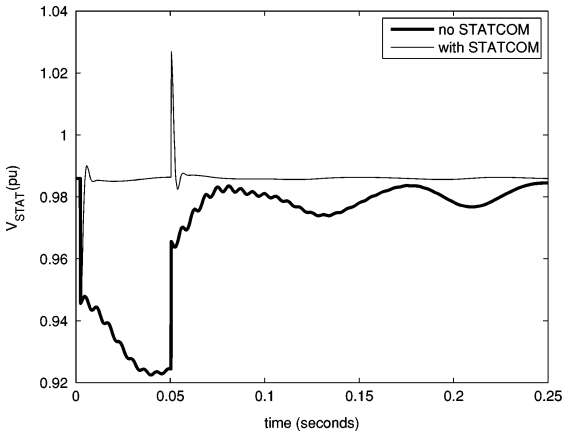

Fig. 3. Controlled versus uncontrolled voltage at bus 34.

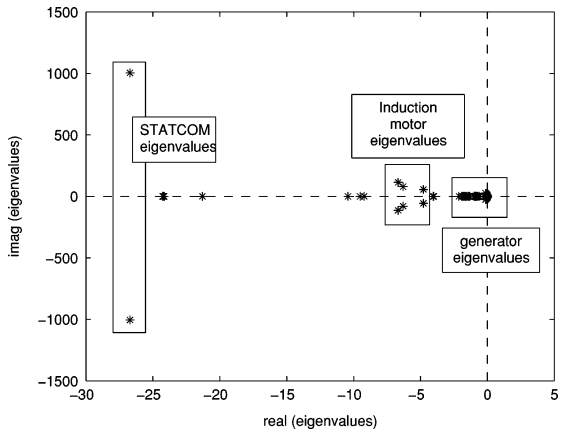

Fig. 4. System eigenvalues.

in Fig. 3. While a discussion of the control approach is not the purpose of this paper, it can be seen that the control works effectively to hold the bus voltage near the pre-fault value and rapidly damps system oscillations. The numerical results presented in this paper are all taken from the controlled STATCOM case.

The eigenvalues of the linearized system are shown in Fig. 4. The micro time step for the simulation is chosen to be $h = 0.2$ ms. From Fig. 4, the largest angular frequency is 1000 rad/s, which is equivalent to 160 cycles/s $(= 1000/2\pi)$. For numerical accuracy, each cycle requires 32 sample points to be sufficiently rendered; thus, the best micro step size is $h = (1/(32)(160)) = 0.0002$ s.

The local truncation error for the trapezoidal integration method of the STATCOM and induction motors calculated from (6) and (15) is shown in Fig. 5. From Figs. 4 and 5, it is apparent that the STATCOM is the "fastest" state—especially after the fault is applied and then removed, but the fast dynamics decay rapidly. After about 0.1 s, the dynamics of the induction motors begin to dominate with induction motors 19 and 35 having a significant effect. Induction motor 24 has the least effect. This figure also highlights the effect of "fast" but not excited state variables. Even though a state variable may be intrinsically fast, unless it is excited, its LTE will remain small, and it will be considered a "slow" state.

From (9) and Fig. 5, the ratio m is related to the ratio of the LTE of the fastest state to the slowest state. *Note that the y-axis in* Fig. 5 *is log-scale*. Taking the ratio of the various LTEs over time indicates that the LTE ratio is greater than or equal to 10; therefore, the ratio of slow timestep to fast timestep should be less than or equal to 10, or $m \leq 10$ to reflect this time response separation. Fig. 6 shows the error versus CPU time as the timestep ratio is increased. The error is calculated as

$$\varepsilon_m = \sum_{i=1}^{N} \frac{1}{x_{i,\text{mean}}} \left(\sum_{t\in[0,T]} |x_{i,m}(t) - \hat{x}_i(t)| \right) \tag{35}$$

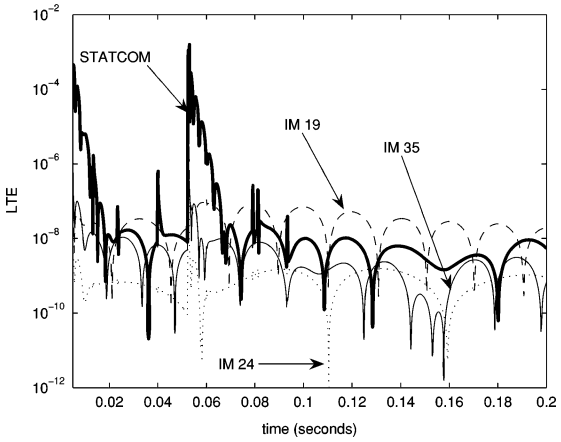

Fig. 5. Local truncation error.

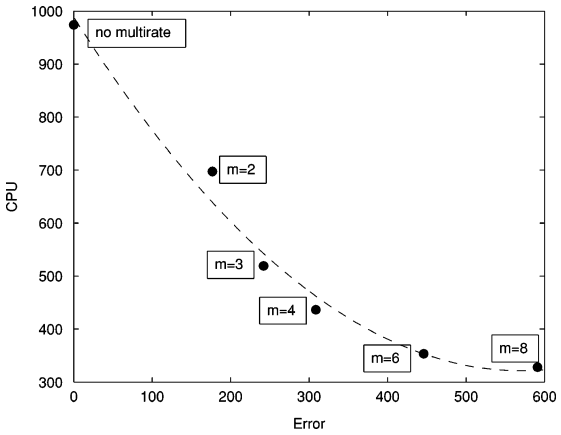

Fig. 6. Error versus CPU for different timestep ratios.

where N is the total number of dynamic and algebraic states, m is the timestep ratio, $\hat{x}$ is the fixed step numerical solution, $x_{i,m}$ is the numerical solution for ratio m, and $x_{i,\text{mean}}$ is the mean value of $x_i(t)$ over $t \in [0\ T]$. Note that the speed-up levels off as m approaches 10. This is predicted from the consideration of the local truncation errors. A comparison of error and CPU time for the multirate versus non-multirate is given in Table I. The fast grouping in the multirate case are: [STATCOM, IM1, IM3, buses 14, 15, 19, 22, 30, 33–40]. In the non-multirate case, the timestep across all states is the same. For example, consider the second row of the table. It indicates that for $m = 2$, the timestep for fast states is h, but for the remaining states, the timestep is $2h$. The combination results in a CPU of 697 and an error of 197. For the non-multirate case, *all* states use a timestep of $2h$. For this case, the CPU is 574 with an error of 581. So while the non-multirate is slightly faster (a speed-up ratio of 1.7 for non-multirate versus as speed up of 1.4 for the multirate), the multirate method has nearly three times better accuracy. This trend is continued until for a ratio of 8, the speed up difference is 5.60 (non-multirate) to 2.98 (multirate) while the multirate method has better than an order of magnitude better accuracy (12.98 to be exact). In the trade-off between speed and accuracy, the multirate method yields far better results than the non-multirate integration. Note that in this paper, the CPU measure is unitless since different processors will have different clock times.

TABLE I
CPU AND ERROR FOR NON-MULTIRATE AND MULTIRATE SIMULATION

m	non-multirate		multirate	
	CPU	error	CPU	error
1	974.0	0	974.0	0
2	574.7	581	697.3	177
3	389.3	1806	518.9	242
4	294.7	2924	436.4	308
6	200.6	5085	353.1	446
8	174.1	7578	327.9	591

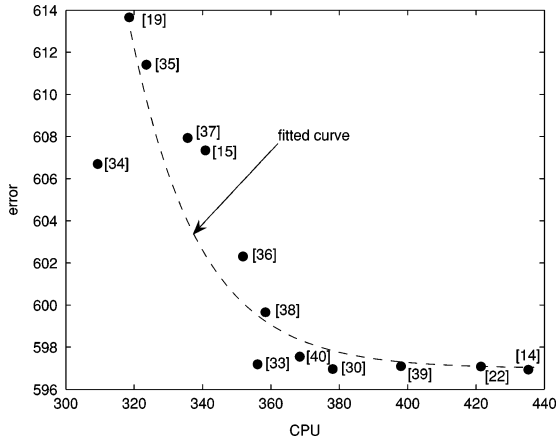

Fig. 7. CPU versus error for various fast partitionings.

The CPU results presented are intended to provide a comparison between methods and should not be considered actual times.

To illustrate the impact of various algebraic partitionings, Fig. 7 compares CPU time versus error as the algebraic states are successively added to the fast partition according to the algebraic partitioning scheme presented earlier. The numbers in the brackets ([]) indicate the number of the bus variable successively added to the fast partition. Note that as the fast buses are successively increased, the error decreases while the CPU time increases. It is not surprising that the first three buses identified are the buses adjacent to induction machines 19 and 35 and the STATCOM (at bus 34). Successive buses "fill in" the surrounding area including the tie lines to other regions of the system (buses 38, 40, 22, and 30). However, after the tie lines are included, the error tends to saturate, implying that the fast dynamics do not penetrate substantially into the remaining system.

Lastly, while the numerical error may seem quite large in absolute value, recall that it is the accumulated error over all states over the entire simulation interval. To provide a reference frame for the size of the error, consider the exact (non-multirate) waveform versus the $m = 8$ multirate waveform in Fig. 8. At full scale, the waveforms appear to be identical. It is only in the "zoomed" case at the point where the fault is cleared (a point of rapid change) that any significant error is apparent. Therefore, even for fairly large absolute errors values, the waveforms are, in fact, quite accurate.

VIII. CONCLUSION

In this paper, an application of the multirate method to DAEs for power systems is presented. The primary contributions of the paper are:

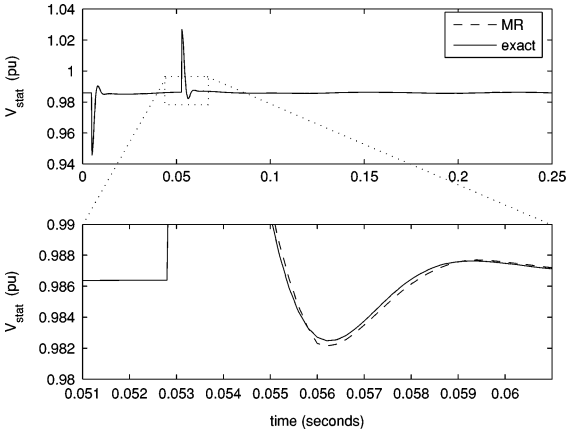

Fig. 8. Exact versus multirate ($m = 8$) waveform.

- a method of selecting the ratio m is developed;
- an algebraic variable partition strategy is developed and implemented;
- the DAE multirate method is applied to the IEEE 118-bus system with promising results.

Future work includes a rigorous error analysis for the algebraic states, including the effect of interpolation.

APPENDIX

The STATCOM Model [15]

$$\frac{1}{\omega_s}\dot{i}_d = -\frac{R_s}{L_s}i_d + \frac{w}{w_s}i_q + \frac{K}{L_s}\cos(\alpha+\theta_i)V_{dc} - \frac{V_i}{L_s}\cos(\theta_i)$$

$$\frac{1}{\omega_s}\dot{i}_q = -\frac{R_s}{L_s}i_q - \frac{w}{w_s}i_d + \frac{K}{L_s}\sin(\alpha+\theta_i)V_{dc} - \frac{V_i}{L_s}\sin(\theta_i)$$

$$\frac{C_{dc}}{\omega_s}\dot{V}_{dc} = -K\cos(\alpha+\theta_i)i_d - K\sin(\alpha+\theta_i)I_q - \frac{V_{dc}}{R_{dc}}$$

where i_d, i_q is the injected dq STATCOM currents, V_{dc} is the voltage across the dc capacitor, R_{dc} is the switching losses, and R_s, L_s is the coupling transformer resistance and inductance.

The STATCOM bus voltage is $V_i\angle\theta_i$, and the power balance equations are

$$0 = V_i(i_d\cos\theta_i + i_q\sin\theta_i) - V_i\sum_{j=1}^{n}V_jY_{ij}\cos(\theta_i-\theta_j-\phi_{ij})$$

$$0 = V_i(i_d\sin\theta_i - i_q\cos\theta_i) - V_i\sum_{j=1}^{n}V_jY_{ij}\sin(\theta_i-\theta_j-\phi_{ij}).$$

REFERENCES

[1] D. Koester, S. Ranka, and G. Fox, Power Systems Transient Stability—A Grand Computing Challenge, Northeaset Parallel Architectures Center, Tech. Rep., 1992.

[2] J. J. Sanchez-Gasca, R. D'Aquila, J. Paserba, W. Price, D. Klapper, and I.-P. Hu, "Extended-term dynamic simulation using variable time step integration," *IEEE Comput. Appl. Power*, vol. 6, no. 4, pp. 23–28, Oct. 1993.

[3] A. Kurita, H. Okubo, K. Oki, S. Agematsu, D. Klapper, N. Miller, J. Price, W. W. , J. Sanchez-Gasca, K. Wirgau, and T. Younkins, "Multiple time-scale power system dynamic simulation," *IEEE Trans. Power Syst.*, vol. 8, no. 1, pp. 216–223, Feb. 1993.

[4] J. Y. 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.

[5] F. Iavernaro, M. L. Scala, and F. Mazzia, "Boundary values methods for time-domain simulation of power system dynamic behavior," *IEEE Trans. Circuits Syst.*, vol. 45, no. 1, pp. 50–63, Jan. 1998.

[6] C. Gear, Multirate Methods for Ordinary Differential Equations, Univ. Illinois at Urbana-Champaign, Tech. Rep., 1974.

[7] M. Günther and P. Rentrop, "Partitioning and multirate strategies in latent electric circuits," *Int. Ser. Numer. Math.*, vol. 117, pp. 33–60, 1994.

[8] M. Crow and J. G. Chen, "The multirate method for simulation of power system dynamics," *IEEE Trans. Power Syst.*, vol. 9, no. 3, pp. 1684–1690, Aug. 1994.

[9] M. Crow and J. G. Chen, "The multirate simulation of FACTS devices in power system dynamics," *IEEE Trans. Power Syst.*, vol. 11, no. 1, pp. 376–382, Feb. 1996.

[10] S. D. Pekarek, O. Wasynczuk, E. Walters, J. Jatskevich, C. Lucas, N. Wu, and P. Lamm, "An efficient multirate simulation technique for power-electronic-based systems," *IEEE Trans. Power Syst.*, vol. 19, no. 1, pp. 399–409, Feb. 2004.

[11] D. Solis, "Multirate integration methods for constrained mechanical systems with interacting subsystems," Ph.D. dissertation, Univ. Iowa, Ames, 1996.

[12] J. Chen, M. Crow, B. Chowdhury, and L. Acar, "An error analysis of the multirate method for power system transient stability simulation," in *Proc. IEEE Power Eng. Soc. Power Systems Conf. Expo.* , 2004, vol. 2, no. 982–986.

[13] L. O. Chua and P. Lin, *Computer Aided Analysis of Electronic Circuits: Algorithms and Computational Techniques*. Englewood Cliffs, NJ: Prentice-Hall, 1975.

[14] A. Semlyen and F. de Leon, "Quasi-Newton power flow using partial Jacobian updates," *IEEE Trans. Power Syst.*, vol. 16, no. 3, pp. 332–339, Aug. 2001.

[15] L. Dong, M. L. Crow, Z. Yang, C. Shen, L. Zhang, and S. Atcitty, "A reconfigurable FACTS system for university laboratories," *IEEE Trans. Power Syst.*, vol. 19, no. 1, pp. 120–128, Feb. 2004.

Jingjia Chen (S'02) received the B.S. and M.S. degrees from Wuhan University, Wuhan, China, in 1999 and 2002, respectively and the Ph.D. degree in electrical engineering from the University of Missouri-Rolla in 2006.

He is currently employed by Pterra, Inc., Albany, NY. His interests are in power system modeling, simulation, and control.

Mariesa L. Crow (SM'91) received the B.S.E. degree from the University of Michigan, Ann Arbor, and the Ph.D. degree from the University of Illinois at Urbana-Champaign.

She is presently the Director of the Energy Research and Development Center and the F. Finley Distinguished Professor of Electrical Engineering at the University of Missouri-Rolla. Her research interests include developing computational methods for dynamic security assessment and the application of power electronics in bulk power systems.