Algorithms for the Accounting of Multiple Switching Events in Digital Simulation of Power-Electronic Systems

M. O. Faruque, *Student Member, IEEE*, Venkata Dinavahi, *Member, IEEE*, and Wilsun Xu, *Senior Member, IEEE*

***Abstract*—Digital simulation of power systems containing power electronics apparatus is challenging due to the need to account multiple switching events within one simulation time-step. This paper describes a family of algorithms, with varying levels of computational complexity, for accounting such switching events in digital simulations. The proposed algorithms are applicable for both off-line and real-time simulations. A comparative study on their performance such as harmonic content, errors in fundamental component and simulation time requirement is presented. A Pulse Width Modulated (PWM) Voltage Source Converter (VSC) based D-STATCOM system is used as a case study for simulations. Simulation results indicate excellent performance (accuracy and efficiency) in comparison with a fixed time-step algorithm using a small step-size.**

***Index Terms*—Digital control, discrete event simulation, extrapolation, interpolation, power electronics.**

I. Introduction

REAL-TIME digital simulation of power systems is becoming increasingly commonplace for testing of new FACTS and HVDC controllers and protection systems [1]–[3] before they are commissioned. When a digital controller for a power-electronic apparatus such as a unified power-flow controller (UPFC) needs to be tested, it is interfaced with a real-time simulator modeling the UPFC and the surrounding power system into which the UPFC is embedded. The outputs of the digital controller are predominantly discrete switching signals which are used to control the switches in the UPFC. The real-time simulator, solving the system differential equations at a certain discrete time step $\triangle t$, is therefore required to account multiple *inter-step* switching events which arrive at its input between two calculation cycles. The number of switching events within any given time-step of the simulator depends on two factors: 1) the switching frequency of the power-electronic system (i.e., its digital controller sampling rate); and 2) the complexity of the power-electronic system (i.e., the number of switches in the system). The higher the switching frequency or the larger the system, the higher is the number of inter-step switching events. Furthermore, the timing of an incoming switching event is not known a priori (since it is controlled by processes external to the simulator) and it seldom coincides exactly with $\triangle t$ thereby creating a switching delay which produces erroneous simulation results.

Various techniques are available in literature to deal with single inter-step switching events in digital simulation under both off-line and real-time conditions. State-space methods [4], [5] have been effectively used to simulate thyristor based circuits. Fixed time-step methods based on nodal analysis have been used for traditional power system transient simulation [6], [7] as well as power-electronic circuit simulation [8], [9] under both off-line and real-time conditions. The advantage of such methods is that the admittance matrix of the simulated network remains unchanged. Combined with a fixed time-step nodal method, linear interpolation has proved to be a very effective technique in estimating the circuit variables at the switching instant. Some of the pioneering research in this area has been reported in [10]–[14]. PSCAD/EMTDC uses a double interpolation technique to allow correction for switching delays [15], [16]. Details of this technique is given in Section III.A.1. Real-time simulator ARENE uses linear interpolation to bring the solution to the switching point, however, the interpolated values are used as the value of next time-step so that equal spacing between data points are maintained. An extrapolation is then used to bring back the solution to the original time-grid [15]. In [17] a fixed time-step technique is described in which states are linearly interpolated assuming instantaneous switching. Then a method for calculating the initial conditions following a discontinuity or switching is presented. The algorithm is applied using both nodal and state-space approaches. An interpolation method using instantaneous solution is also described in [18]. HYPERSIM [19] is a parallel processor based real-time simulator which implements a backward-forward interpolation technique to simulate switching phenomenon in power-electronic simulation. In [20] a structure changing concept is discussed, which can be used to simulate the changes at arbitrary time instants. Variable time-step nodal methods using linear interpolation to account for single interstep switching event have also been reported in [21], [22]. A method based on re-initialization through interpolation and extrapolation for treatment of discontinuities in simulating power-electronic circuits has been proposed in [23].

This paper presents algorithms designed to take into account multiple switching events in digital simulation of power-electronic systems employing forced-commutated devices. A full comparison of the algorithms is presented under both open-loop

Manuscript received December 4, 2003; revised June 15, 2004. This work was supported in part by the Natural Sciences and Engineering Research Council (NSERC) of Canada and in part by the University of Alberta. Paper no. TPWRD-00613-2003.

The authors are with the University of Alberta, Edmonton, AB T6G 2V4, Canada (e-mail: faruque@ece.ualberta.ca; dinavahi@ece.ualberta.ca; wxu@ece.ualberta.ca).

Digital Object Identifier 10.1109/TPWRD.2004.834672

0885-8977/$20.00 © 2005 IEEE

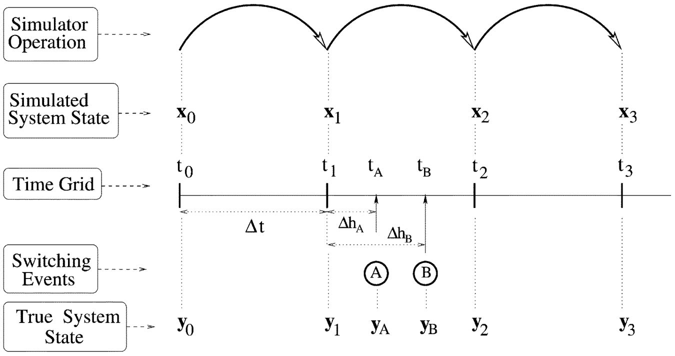

Fig. 1. Multiple interstep switching events in digital simulation.

and closed-loop control operation. The proposed algorithms are not limited to fixed time-step schemes but also employ variable time-step schemes along with linear interpolation/extrapolation to meet simulation accuracy and efficiency constraints. Section II briefly describes the fixed time-step approach in digital simulation and Section III presents the algorithms. Section IV describes the test system, a six-pulse PWM D-STATCOM system, and its digital controller. Section V provides results showing full comparison of the performance of the proposed algorithms. Section VI presents the conclusions.

II. Multiple Switching Events in Fixed Time-Step Digital Simulation

When a real-time digital simulator modeling a power-electronic system is interfaced with a digital controller for that system, the discrete firing pulses coming from the controller are rarely in synchronism with the time-step chosen for the simulator. Unlike a physical system, the digital simulator cannot respond instantaneously to a switching event; it can respond only after it has completed its current task (calculating states for $\triangle t$ thereby creating a switching delay. Furthermore, since a switching event does not always come in at the same time instant the switching delay introduced is not constant; the delay can range from a fraction of a time-step up to a full time-step. Fig. 1 illustrates the fixed time-step simulation approach with two switching events occurring within one time-step. $\mathbf{x}_i, (i = 0, 1, 2, \ldots)$ represent the states of the system computed by the simulator at every time step $\triangle t$ and $\mathbf{y}_i, (i = 0, 1, 2, \ldots)$ represent the true states of the physical system.

In the context of the paper, *Time-Step* refers to the discrete-time interval with which the system differential equations are numerically integrated by the simulator; the term *step-size* is used interchangeably with *time-step*. In contrast, the term *Time-Grid* is chosen to represent the interval at which output data points are generated by the simulator. In Fig. 1 the simulation time-grid is the same as the time-step $\triangle t$. The switching events Ⓐ and Ⓑ which occur at times t_A and t_B respectively are accounted at time t_2 when the real-time simulator has already calculated and emitted the incorrect state $\mathbf{x}_2$. Real-time operation does not permit recalling and changing state $\mathbf{x_2}$. The physical system, on the other hand, would respond to the events with states $\mathbf{y}_A$ and $\mathbf{y}_B$ at times t_A and t_B respectively and $\mathbf{y}_2$ would be the true state of the system at time t_2.

In a fixed time-step algorithm, at the beginning of every time-step the simulator looks for the switching event and its timing information. With this knowledge it updates the power-electronic model in the system. Depending on the type of switches, the circuit variables $\mathcal{V}$, for example voltages at the point of common coupling, may be functions of the firing signals $\mathcal{S}$ and/or the network state $\mathbf{x}$

$$\mathcal{V} = \mathbf{f}(\mathcal{S}, \mathbf{x}). \tag{1}$$

Then, the simulator proceeds to obtain the numerical solution of the network state equations for the given time-step. The state equations for a linear time-invariant network can be written in general as

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \tag{2}$$

subject to the initial conditions $\mathbf{x}(t_0) = \mathbf{x}_0$ and $\mathbf{u}(t_0) = \mathbf{u}_0$, where $\mathbf{x} \in \mathbb{R}^N$ is the network state vector and $\mathbf{u} \in \mathbb{R}^M$ is the input vector. $\mathbf{A}$ and $\mathbf{B}$ are constant matrices dependent on the network constants such as R, L, and C. With a time-step of integration $\triangle t$, the solution at time t can be expressed in terms of the solution at time $(t - \triangle t)$

$$\mathbf{x}(t) = \mathbf{x}(t - \triangle t) + \int_{t-\triangle t}^{t} [\mathbf{A}\mathbf{x}(\varsigma) + \mathbf{B}\mathbf{u}(\varsigma)]\, d\varsigma \tag{3}$$

where ς is a variable of integration. Numerical solution of (3) can be obtained by approximating the integral in (3) using a numerical integration algorithm. Let the subscript $(n-1)$ denote quantities at time $(t - \triangle t)$ and n quantities at time t. Then

$$\mathbf{x}_n = \mathbf{x}_{n-1} + \mathbf{g}(\triangle t, \mathbf{A}, \mathbf{B}, \mathbf{x}_n, \mathbf{x}_{n-1}, \ldots, \mathbf{u}_n, \mathbf{u}_{n-1}, \ldots) \tag{4}$$

(4) forms the discrete time solution of (2). The number of history terms of $\mathbf{x}$ and $\mathbf{u}$ in the integral approximation will depend on the type of the chosen numerical method. A difference equation can be obtained by using the Trapezoidal Rule (assuming the integrand to be piecewise linear) on (3)

$$\mathbf{x}_n = \mathbf{x}_{n-1} + \mathbf{A}\frac{\triangle t}{2}(\mathbf{x}_n + \mathbf{x}_{n-1}) + \mathbf{B}\frac{\triangle t}{2}(\mathbf{u}_n + \mathbf{u}_{n-1}) \tag{5}$$

which gives us

$$\mathbf{x}_n = \boldsymbol{\alpha}\mathbf{x}_{n-1} + \boldsymbol{\beta}(\mathbf{u}_{n-1} + \mathbf{u}_n) \tag{6}$$

where

$$\boldsymbol{\alpha} = \left(\mathbf{I} - \mathbf{A}\frac{\triangle t}{2}\right)^{-1}\left(\mathbf{I} + \mathbf{A}\frac{\triangle t}{2}\right) \tag{7}$$

and

$$\boldsymbol{\beta} = \left(\mathbf{I} - \mathbf{A}\frac{\triangle t}{2}\right)^{-1}\left(\mathbf{B}\frac{\triangle t}{2}\right). \tag{8}$$

Recursive solution of (6) starting with $n = 0$ will yield the desired approximate solution of (3). Equation (6) can be rewritten as

$$\mathbf{x}_n = \boldsymbol{\beta}\mathbf{u}_n + \mathbf{J} \tag{9}$$

where

$$\mathbf{J} = \boldsymbol{\alpha}\mathbf{x}_{n-1} + \boldsymbol{\beta}\mathbf{u}_{n-1} \tag{10}$$

represents the *history* of the network since $\mathbf{J}$ depends on the state and input at the previous time-step. This formulation is identical to that of Dommel's method [6] using trapezoidal rule. $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ can be interpreted as the discrete time equivalent network parameters dependent not only on the original network parameters $\mathbf{A}$ and $\mathbf{B}$ but also on the time-step $\triangle t$. Therefore, a change in the time-step $\triangle t$ would necessitate a recalculation of $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$.

III. Algorithms for the Accounting of Multiple Switching Events

As shown in Fig. 1, two switching events Ⓐ and Ⓑ arrive in the time interval $(t_2 - t_1)$ while the simulator computes and emits state $\mathbf{x}_2$ at time t_2. Although, real-time operation does not allow changing state $\mathbf{x}_2$, it does allow the simulator to take certain *corrective* action before proceeding to calculate the next state $\mathbf{x}_3$ at time t_3 provided that the simulator has the timing information of the switching events relative to the simulation time-grid. The earliest time at which this information can be made available to the simulator is at the instant an event occurs (i.e., t_A and t_B, and the latest is at the end of time t_2. The algorithms described in this section can be broadly classified into two categories).

1) *Fixed Time-Grid Algorithms*: In these algorithms the output data of the simulation is obtained at equal intervals of time thereby fixing the time-grid. If $\triangle t$ is varied internally, then a re-synchronization is performed to get the simulation back on the original time-grid. Therefore, these algorithms can have a fixed or a variable simulation time-step. Algorithms I through VII in Fig. 2 belong to this category. For the sake of clarity these algorithms are illustrated with only two switching events.
2) *Variable Time-Grid Algorithms*: In these algorithms, the real-time simulator is interrupted in its normal operation at the instant a switching event occurs, effectively varying the external simulation time-grid. However, the internal time-step $\triangle t$ of the simulation is always fixed. Algorithm VIII belongs to this category.

At the beginning of each time-step, the simulator takes one of the two operational paths based on the information it receives about any switching events:

1) Normal Operation described in Section II is executed in case of no switching event has been detected in the previous time-step.
2) In case of one or more switching events in the previous time step, *Post-event Correction Operation* is executed. Depending on the algorithm, correction involves linear interpolation, linear extrapolation or a change of time-step. The correction variables include the state variables of the system and the inputs (voltage or current sources). After correction to the point of switching, the system is updated before proceeding to the next time-step. During the updates, inductor currents and capacitor voltages are kept unchanged. The following algorithms explain more details of this mode of operation.

A. Fixed Time-Grid Algorithms

Once the exact instant of switching event is detected, these algorithms implement post-event corrections. For all the algorithms, the interpolated variables include the state variables, $\mathbf{x}$ (inductor currents and capacitor voltages) and the inputs $\mathbf{u}$ (AC system voltages). These variables are interpolated to the point of switching and then the power-electronic converter model (1) is updated in accordance with the switching changes. The *updating* procedure depends on the model used to simulate the switches. Using the switching function models, the switch status are flipped and the voltage levels are decided.

1) Algorithm I: This algorithm is similar to the Double interpolation technique that has been used by PSCAD/EMTDC [15].

After calculating state $\mathbf{x}_2$, when the simulator has acknowledged that two events have arrived in the previous simulation interval at times t_A and t_B, the following steps can be taken to *correct* the subsequent state calculation:

1) Based on states $\mathbf{x}_1$ and $\mathbf{x}_2$, state $\mathbf{x}_A^{(1)}$ at time t_A is linearly interpolated as

$$\mathbf{x}_A^{(1)} = \mathbf{x}_1 + \frac{(t_A - t_1)}{\triangle t}(\mathbf{x}_2 - \mathbf{x}_1). \tag{11}$$

2) Once the interpolation is performed, the power-electronic model (1) is updated according to switching event Ⓐ. This will yield $\mathbf{x}_A^{(2)}$ which is different from $\mathbf{x}_A^{(1)}$.
3) State $\mathbf{x}_{A+\triangle t}$ at time $(t_A + \triangle t)$ is calculated using (6).
4) Now, based on $\mathbf{x}_A^{(2)}$ and $\mathbf{x}_{A+\triangle t}$, state $\mathbf{x}_B^{(1)}$ is interpolated as

$$\mathbf{x}_B^{(1)} = \mathbf{x}_A^{(2)} + \frac{(t_B - t_A)}{\triangle t}\left(\mathbf{x}_{A+\triangle t} - \mathbf{x}_A^{(2)}\right). \tag{12}$$

5) The power-electronic model (1) is again updated using switching event Ⓑ and state $\mathbf{x}_B^{(2)}$ is calculated in accordance with the switching changes at t_B.
6) State $\mathbf{x}_{B+\triangle t}$ at time $(t_B + \triangle t)$ is calculated using (6).
7) The corrected state $\hat{\mathbf{x}}_2$ at time t_2 is now interpolated based on states $\mathbf{x}_B^{(2)}$ and $\mathbf{x}_{B+\triangle t}$ as

$$\hat{\mathbf{x}}_2 = \mathbf{x}_B^{(2)} + \frac{(t_2 - t_B)}{\triangle t}\left(\mathbf{x}_{B+\triangle t} - \mathbf{x}_B^{(2)}\right). \tag{13}$$

This step puts the simulation on the original time-grid.
8) Finally, state $\mathbf{x}_3$ is calculated from $\hat{\mathbf{x}}_2$ using (6).

There are 3 interpolations and 3 full-state calculations and two model updates involved in going from time t_2 to time t_3, with no change in $\triangle t$, after the simulator receives the information about the switching events. Step 7 is solely an internal calculation since it is not possible to change $\mathbf{x}_2$ in real-time. This step may be used in off-line simulations to produce more accurate results.

2) Algorithm II: After Step 6 in Algorithm I:

7. Using $\mathbf{x}_B^{(2)}$ and $\mathbf{x}_{B+\triangle t}$, state $\mathbf{x}_3$ is linearly extrapolated.

For this step, states $\mathbf{x}_B^{(2)}$ and $\mathbf{x}_{B+\triangle t}$ have been chosen, in particular, because they are fully corrected states. Note that state

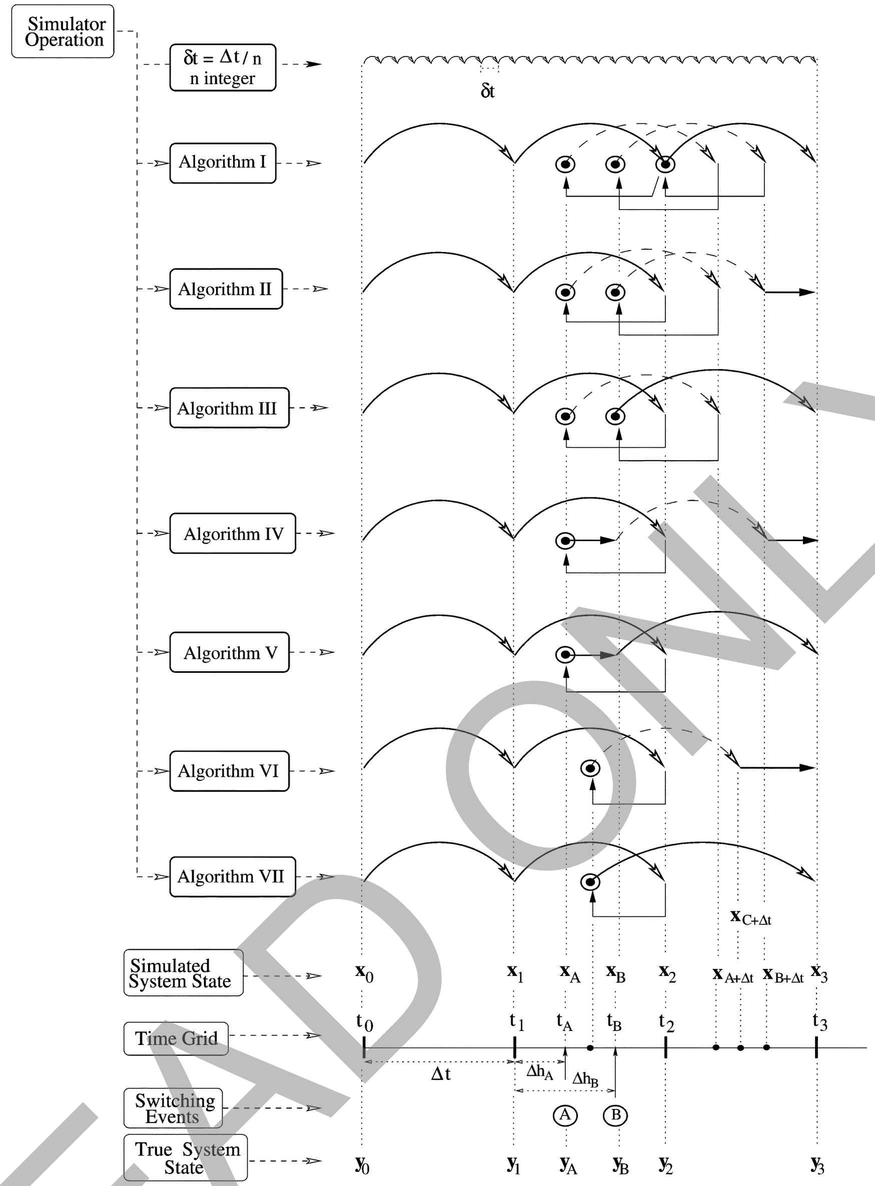

Fig. 2. Digital simulation algorithms for multiple switching events based on a fixed time-grid. ⌒ ⌒: Full-state calculation, ⊙: Linear Interpolation, ⟶: Linear Extrapolation.

$\mathbf{x}_2$ is left uncorrected, however, for off-line simulator state $\mathbf{x}_2$ can be corrected to have more accurate results. Thus, in this algorithm there are 2 interpolations, 2 full-state calculations and 1 extrapolation with a fixed $\triangle t$.

3) Algorithm III: After Step 5 in Algorithm I (i.e., once $\mathbf{x}_B^{(2)}$ has been determined)

6. The step-size of the simulation is changed from $\triangle t$ to $\{\triangle t + (t_2 - t_B)\}$ and $[\boldsymbol{\alpha\beta}]_{\{\triangle t + (t_2 - t_B)\}}$ is calculated.
7. State $\mathbf{x}_3$ is calculated using (6).

Therefore, there are 2 interpolations, 2 full-state calculations and 1 $[\boldsymbol{\alpha\beta}]$ recalculation in this algorithm.

4) Algorithm IV: After Step 2 in Algorithm I, i.e., after $\mathbf{x}_A^+$ has been obtained, this algorithm uses the following steps:

3. $\mathbf{x}_\mathbf{B}^{(1)}$ is extrapolated based on $\mathbf{x}_1$ and $\mathbf{x}_A^{(2)}$

$$\mathbf{x}_B^{(1)} = \mathbf{x}_1 + \frac{(t_B - t_1)}{(t_A - t_1)} \left(\mathbf{x}_A^{(2)} - \mathbf{x}_1 \right). \tag{14}$$

4. The power-electronic model (1) is updated to obtain $\mathbf{x}_\mathbf{B}^{(\mathbf{2})}$ using switching event Ⓑ.
5. $\mathbf{x}_{B+\triangle t}$ at time $t_B + \triangle t$ is calculated using (6).
6. Based on $\mathbf{x}_B^{(2)}$ and $\mathbf{x}_{B+\triangle t}$ state $\mathbf{x}_3$ is extrapolated as

$$\mathbf{x}_3 = \mathbf{x}_B^{(2)} + \frac{(t_3 - t_B)}{\triangle t} \left(\mathbf{x}_{B+\triangle t} - \mathbf{x}_B^{(2)} \right). \tag{15}$$

This algorithm is a variation of Algorithm II in which 1 interpolation and 1 full-state calculation has been substituted by one additional extrapolation with $\triangle t$ remaining fixed.

5) Algorithm V: After Step 4 in Algorithm IV (i.e., after obtaining $\mathbf{x}_B^{(2)}$ by extrapolation from $\mathbf{x}_1$ and $\mathbf{x}_A^{(2)}$ and updating switching at t_B), this algorithm uses the following steps:

5. The step-size of the simulation is changed from $\triangle t$ to $\{\triangle t + (t_2 - t_B)\}$ and $[\boldsymbol{\alpha\beta}]_{\{\triangle t + (t_2 - t_B)\}}$ is calculated.
6. $\mathbf{x}_3$ is calculated using (6).

This algorithm requires 1 interpolation, 1 full state calculation, 1 extrapolation, and 1 $[\boldsymbol{\alpha\beta}]$ recalculation. It is a variation of Algorithm III where 1 interpolation and 1 full-state calculation has been replaced by 1 extrapolation.

6) Algorithm VI: Once the simulator obtains the information of the two switching events then:

1) The system state at $t_C = (t_A + t_B)/2$, the average location of the two events at t_A and t_B, is interpolated based on the states $\mathbf{x}_1$ and $\mathbf{x}_2$

$$\mathbf{x}_C = \mathbf{x}_1 + \frac{(t_C - t_1)}{\triangle t}(\mathbf{x}_2 - \mathbf{x}_1). \tag{16}$$

The power-electronic model (1) is updated using both switching events Ⓐ and Ⓑ.

2) State $\mathbf{x}_{C+\triangle t}$ is calculated using (6).

3) $\mathbf{x}_3$ is now extrapolated using $\mathbf{x}_{C+\triangle t}$ and $\mathbf{x}_C$

$$\mathbf{x}_3 = \mathbf{x}_C + \frac{(t_3 - t_C)}{\triangle t}(\mathbf{x}_{C+\triangle t} - \mathbf{x}_C). \tag{17}$$

Thus, in this algorithm we have 1 interpolation, 1 full state calculation and 1 extrapolation with a fixed $\triangle t$.

7) Algorithm VII: After Step 1 in Algorithm VI, this algorithm uses the following steps:

2. The step-size of the simulation is changed from $\triangle t$ to $\{\triangle t + (t_2 - t_C)\}$ and $[\boldsymbol{\alpha\beta}]_{\{\triangle t+(t_2-t_C)\}}$ is calculated.
3. State $\mathbf{x}_3$ is calculated using (6).

Therefore, we have 1 interpolation, 1 full state calculation and 1 $[\boldsymbol{\alpha\beta}]$ recalculation.

If the simulation time-step is changed internally as in algorithms III, V, and VII, the recalculation of $[\boldsymbol{\alpha\beta}]$ may become a significant computational bottleneck if the network is large. An efficient way to circumvent on-line calculation of $[\boldsymbol{\alpha\beta}]$ for a large network is to store pre-computed values for several sub-multiples of $\triangle t$ ($\delta t = \triangle t/n, n$ integer), in a look-up table and apply them when required. For example, if $n = 10, [\boldsymbol{\alpha\beta}]$ can be computed off-line 10 times for 10 values of time $(\triangle t + k\delta t)k = 1, 2, \ldots 10$. Such an approach may not be precise as to the location of a switching event, however, it is sufficiently accurate compared with $[\boldsymbol{\alpha\beta}]$ calculation based on a single large time-step $\triangle t$. This pre-calculation of $[\boldsymbol{\alpha\beta}]$ is feasible if the network condition prior to the switching events is known. During a transient, this condition is generally unpredictable. Another approach to accommodate a large network could be to interface the power-electronic circuit simulation using algorithms III, V, or VII, with a fixed step-size network simulation, as reported in [4].

The above algorithms are indeed capable of correcting three switchings in one time-step. In such a case, the first two switchings are corrected initially, followed by the correction of the third switching event. Further extension for accounting more than three events is straight forward. The cost for having more switching events in one time-step is an increase in complexity of the algorithms and they become extremely cumbersome. An efficient trade-off in handling multiple switching events would be to reduce the time-step $\triangle t$ such that at most two or three events occur within one $\triangle t$.

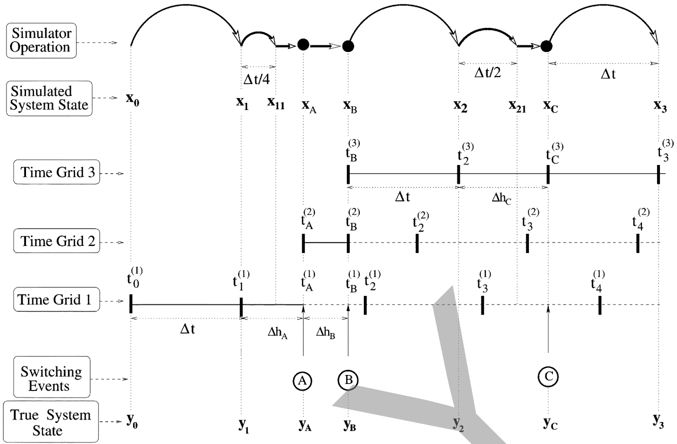

Fig. 3. Variable time-grid Algorithm VIII for two switching events. ⌒: Full-state calculation, ●: Switching update, ⟶: Linear extrapolation.

B. Variable Time-Grid Algorithms

1) Algorithm VIII: This algorithm is event-driven and varies its external time-grid whenever it is interrupted by a switching event. However, the internal time-step of the simulation is fixed. Whenever a switching event is encountered, the simulator is stopped at that instant and the time difference $(\triangle h)$ between the initial point of that particular time step and the switching instant is calculated. Based on the value of $\triangle h$, one of the following four cases is executed.

- **Case I** If $\triangle h < \triangle t/4$, an extrapolation using the two previously known states is performed to find the state at the point of switching.
- **Case II** If $\triangle h \geq \triangle t/4$ but $\leq \triangle t/2$, Backward Euler's integration is adopted for $\triangle t/4$ time-step and then an extrapolation is performed to the point of switching.
- **Case III** If $\triangle h \geq \triangle t/2$ but $\leq 3\triangle t/4$, Trapezoidal integration is adopted for $\triangle t/2$ time-step and then an extrapolation is performed to the point of switching.
- **Case IV** If $\triangle h \geq 3\triangle t/4$, Trapezoidal integration is adopted for $\triangle t/2$ time-step, Backward Euler's integration is adopted for $\triangle t/4$ time-step and then an extrapolation is performed to the point of switching.

Once the simulation reaches to the switching point then, the switching change is updated and the simulation proceeds with original time-step but moves to a new time-grid. In case of two switching events, similar procedure is adopted and the simulation follows any of the four ways to reach the switching instant. For each switching event the external time-grid is changed even though the internal simulation time step remains unchanged. The only additional requirement for this algorithm is to save $[\boldsymbol{\alpha\beta}]$ for a time-step of $\triangle t/2$.

Fig. 3 illustrates this algorithm for both single and two switching events. The simulation starts at the time $t_0^{(1)}$ on Time-Grid 1. State $\mathbf{x}_1$ at time $t_1^{(1)}$ is computed from state $\mathbf{x}_0$ with a fixed time-step $\triangle t$ using (6). At time $t_1^{(1)}$ the simulator begins calculating the state for time $t_2^{(1)}$, however, it is interrupted by a switching event Ⓐ at time $t_A^{(1)}$. The time difference between $t_1^{(1)}$ and $t_A^{(1)}$ is less than $\triangle t/2$ but greater than $\triangle t/4$. Therefore, Case II is followed where Backward

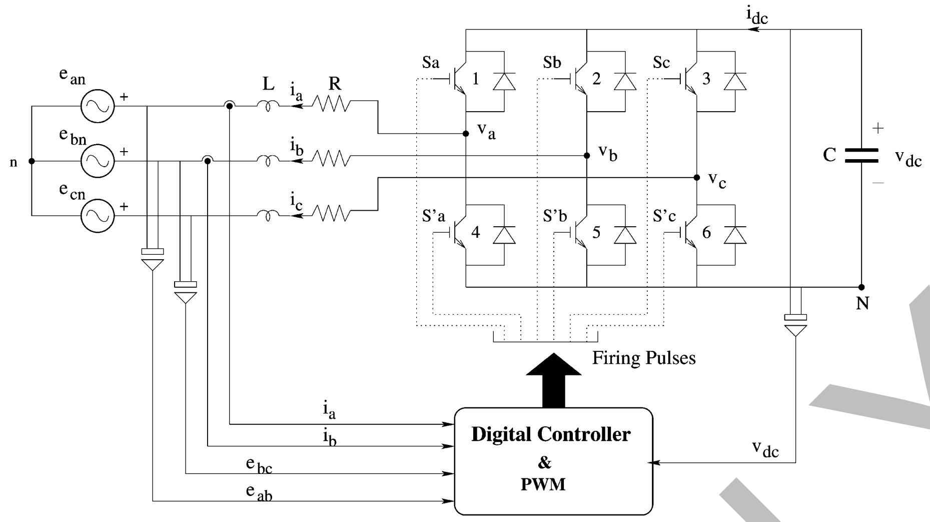

Fig. 4. D-STATCOM system and its digital controller.

Euler yields $\mathbf{x_{11}}$ at a step of $\triangle t/4$. Then, the system state $\mathbf{x}_A$ at time $t_A^{(1)}$ is extrapolated from the state $\mathbf{x_1}$ and $\mathbf{x_{11}}$. The state at t_A is then updated with the switching events. Now, with the updated state $\mathbf{x}_A$, the simulation is transferred to Time-Grid 2 and proceeds with a fixed time-step $\triangle t$ from time $t_A^{(2)}$ onward and expected to reach at time $t_2^{(2)}$. However, the simulation is interrupted again by a second switching event Ⓑ at time $t_B^{(2)}$. As the distance between $t_A^{(2)}$ and $t_B^{(2)}$ is less than $\triangle t/2$ (Case I), a single extrapolation is performed based on the state at $\mathbf{x_A}$ at $t_A^{(2)}$ and $\mathbf{x_1}$ at $t_1^{(1)}$ to find the state $\mathbf{x_B}$ at $t_B^{(2)}$. The state $\mathbf{x_B}$ at $t_B^{(2)}$ is then updated with the switching changes and the simulation moves to Time-Grid 3 and starts from $t_B^{(3)}$. From $t_B^{(3)}$ we assume that there is no switching event in the next time step and a full time-step calculation is performed to reach the state $\mathbf{x_2}$ at time $t_2^{(3)}$. From the time $t_2^{(3)}$ the simulation starts for a full step $\triangle t$ but is interrupted by an event Ⓒ at time $t_C^{(3)}$. Since the distance between the time $t_2^{(3)}$ and $t_C^{(3)}$ is greater than $\triangle t/2$ but less than $3\triangle t/4$ (Case III), first a trapezoidal integration is performed to calculate the state $\mathbf{x_{21}}$ at $t_2^{(3)} + \triangle t/2$. Thereafter, linear extrapolation between $\mathbf{x_2}$ and $\mathbf{x_{21}}$ is performed to reach the state $\mathbf{x_C}$ at $t_C^{(3)}$. The simulator then updates the switching changes and moves to a new time-grid (not shown in the figure). The simulation follows the similar approach for all the future switching events and the states are recorded. Clearly, the output data of the simulation will be unevenly spaced due to the changes in the original time-grid. Linear interpolation is used to get an evenly spaced data set.

The benefits of implementing this variable time grid algorithm are as follows:

- No post event correction is necessary as the algorithm always uses the corrected states.
- The algorithm is suitable for implementing in real-time as it always advances with correct state values needed for the controller to produce gating signals.
- The internal simulation time step remains always unchanged and recalculation of $[\boldsymbol{\alpha\beta}]$ is obviated.

IV. Case Study: D-STATCOM System

A. System Model

Fig. 4 illustrates a generic experimental set-up of the D-STATCOM power system and its digital controller. The system consists of a three-phase insulated-gate bipolar transistor (IGBT) bridge (switching frequency $f_{\rm sw} = 1$ kHz) connected via a series impedance ($R = 0.5\ \Omega$ and $L = 3.0$ mH) to the ac bus assumed to be a balanced voltage source ($E = 110$ V rms $ll, f = 60$ Hz). L represents the filter inductance and the leakage inductance of the converter transformer. R accounts for the converter and transformer conduction losses. The dc side capacitor is $C = 4900\ \mu$F. The switches in the voltage-source converter (VSC) are modeled as ideal bi-directional switches with gate turn-on and turn-off controls. The VSC model is based on discrete switching functions. Three switching signals $S_k, k = \{a, b, c\}$ control the upper switches in each leg of the converter. The lower switches in each leg of the converter are switched in a complementary manner. Accordingly, the output voltages of the converter with respect to the negative dc bus N are given as

$$v_{kN} = S_k v_{\rm dc}; \quad k = \{a, b, c\}. \tag{18}$$

Under balanced conditions the converter output voltages with respect to the AC system neutral n are given as

$$v_{kn} = \frac{2}{3} v_{kN} - \frac{1}{3} \sum_{\substack{i=\{a,b,c\}\\ i\neq k}} v_{iN}. \tag{19}$$

Taking i_a, i_b, and $v_{\rm dc}$ as the states, the time-domain model of the system can be represented by three differential equations

$$\frac{di_a}{dt} = -(R/L)i_a + (1/L)(v_{an} - e_{an}) \tag{20}$$

$$\frac{di_b}{dt} = -(R/L)i_b + (1/L)(v_{bn} - e_{bn}) \tag{21}$$

$$\frac{dv_{\rm dc}}{dt} = -(1/C)(i_{\rm dc}). \tag{22}$$

The current in phase c, can be expressed as a linear combination of i_a and i_b. The DC side current is given as

$$i_{\text{dc}} = \sum_{k=\{a,b,c\}} S_k i_k. \tag{23}$$

The above equations are solved by the simulator at every time-step $\triangle t$.

B. Digital Controller

The control design [25] has been carried out in the synchronous dq frame. The control algorithm is executed every T_s which is the controller sampling period. First a coordinate transformation from abc frame to dq frame is performed on the system signals. Then a decoupled control of the currents i_d and i_q is performed using P-I compensator's for each of the current loops. The DC-link voltage is regulated through an external feedback loop. The resulting control quantities- modulation index and phase angle of the control signal -are then sent to the pulsewidth modulator (PWM) for the generation of switching signals based on the sampling technique approach [26]. A delay of one sampling period T_s is introduced in the controller implementation in order to account for the finite execution time taken by the DSP.

V. Results and Discussion

All of the algorithms, I through VIII, have been used to simulate the D-STATCOM system and their performance was compared with that of a fixed time-step algorithm (Fig. 1). Results were obtained through off-line simulation of programs written in C. It was observed that as the computational complexity of the algorithms decreased, their execution speed increased, however, at the cost of a small inaccuracy. Both open-loop and closed-loop control studies were performed. In the modulator, sinusoidal PWM scheme with 1-kHz switching frequency was used in which only half of the carrier signal was implemented in one sampling period T_s of the digital controller (i.e., $T_s = (1/2)T_{\text{sw}}$ or, $f_s = 2f_{\text{sw}}$ where $f_{\text{sw}} = 1$ kHz). Such an approach was taken in order to limit the number of discrete switching events to a maximum of three in one simulation time-step $\triangle t$. Fig. 5 illustrates the multiple switching phenomenon using one period of the triangular carrier used in PWM. The snapshot of PWM produced here starts at time t $= 2500$ μs and ends at t $= 3500$ μs with $\triangle t = 100$ μs. The control signals computed for the three phases intersect with the first leg of the carrier signal at t $= 2550$ μs (phase c), t $= 2829$ μs (phase b) and t $= 2869$ μs (phase a). The switching signals $S_k (k = a, b, c)$ are also shown in Fig. 5. Evidently, the switching of phase c is the only event during the time interval t $= 2500$ μs to t $= 2600$ μs, whereas switching of phase a and phase b constitute two events in the time interval t $= 2800$ μs to t $= 2900$ μs. Similar behavior can be seen during the second leg of the carrier signal. It can also be observed that the case of two switching events occurs when two control values are close in their magnitude. As the time-step $\triangle t$ is increased, multiple switchings occur more often. This fact is corroborated in Table I which records the frequency of time-steps with no events, one event and two events.

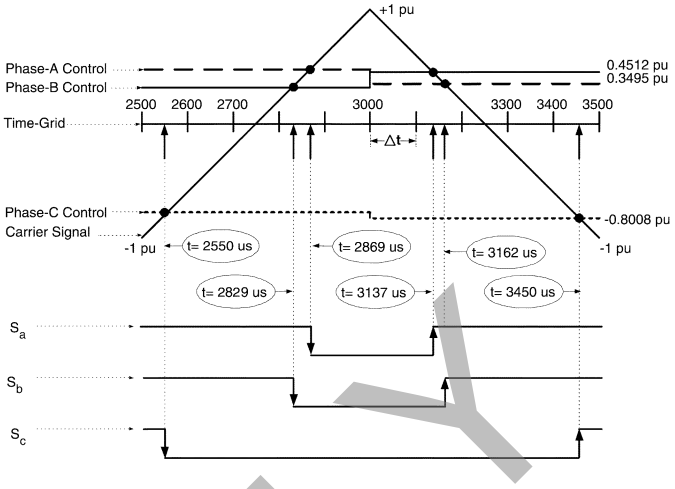

Fig. 5. Illustration of multiple switchings through a detailed view of one complete cycle of PWM for D-STATCOM system.

TABLE I
Variation of Switching Events With Simulation Step-Size for a Simulation Period of 2 s With Carrier Frequency of 1 kHz

$\triangle$t μs	Time-steps with 0 events		Time-steps with 1 event		Time-steps with 2 events	
	No.	%	No.	%	No.	%
10	188038	94.02	11926	5.96	34	0.02
50	28054	70.14	11895	29.74	49	0.12
100	10086	50.43	7833	39.17	2080	10.40
200	1767	17.67	5924	59.25	2308	23.08

A. Open-Loop Control

Under open-loop conditions, the output of the VSC was varied by varying the modulation index $m_a (= V_{\text{control}}/V_{\text{tri}})$ and the phase angle δ of the control signal. The value $m_a = 0.8$ and $\delta = 10°$ were chosen in order to get high fundamental error and THD using fixed time-step algorithm so that the improvements due to the corrected algorithms can be clearly demonstrated. To calculate the fundamental error, the reference is chosen as a case of time-step, $\Delta t = 1$ μs. This Δt is 1000 times smaller than the switching time period T_s (=500 μs) and it has been selected to serve as a benchmark against which the results obtained with limited step-sizes $\Delta t = (10, 50\,100\,150$ μs) were compared. Results for the fixed time-step approach were also verified against simulations on PSCAD/EMTDC Version 3. The two major effects observed due to the changes in Δt were

1) A significant change in the fundamental component of the phase current.
2) A marked crowding of the frequency spectrum with noncharacteristic harmonics and, hence, a degradation in THD.

The change in fundamental component is expressed in terms of percentage absolute error which is defined as

$$\varepsilon = \left(\frac{|(I_1)_{\Delta t} - (I_1)_{1\ \mu\text{s}}|}{(I_1)_{1\ \mu\text{s}}} \right) 100 \tag{24}$$

where $(I_1)_{\Delta t}$ is the fundamental component of current at a particular time-step and $(I_1)_{1\ \mu\text{s}}$ is the fundamental component

TABLE II
FUNDAMENTAL ERROR ε AND THD OF I_a FOR VARYING Δt AND THE TOTAL SIMULATOR OPERATIONS. (A) FULL-STATE COMPUTATION, (B) LINEAR INTERPOLATION, (C) LINEAR EXTRAPOLATION, (D) $[\alpha, \beta]$ RE-CALCULATION

Algorithms	$\triangle t = 10\mu s$		$\triangle t = 50\mu s$		$\triangle t = 100\mu s$		$\triangle t = 150\mu s$ $\mathbf{x_2}$ uncorrected		$\triangle t = 150\mu s$ $\mathbf{x_2}$ corrected	**Simulator Operations**				
	ε (%)	$THD(\%)$	ε (%)	$THD(\%)$	ε (%)	$THD(\%)$	ε (%)	$THD(\%)$	$THD(\%)$	A	B	C	D	Total
Fixed $\triangle t$	2.40	16.19	16.95	16.85	33.66	16.64	38.25	46.15	46.15	1	0	0	0	1
I	0.30	16.33	0.58	16.36	0.92	16.32	3.10	19.22	19.22	3	3	0	0	6
II	0.23	16.42	0.40	18.72	3.80	29.03	0.57	35.45	18.70	2	2	1	0	5
III	0.29	16.43	0.35	18.78	9.30	28.75	4.80	36.32	18.17	2	2	0	1	5
IV	0.29	16.43	0.34	18.72	7.60	37.32	1.10	38.03	23.01	1	1	2	0	4
V	0.29	16.43	0.34	18.78	6.25	36.47	6.01	39.73	24.08	1	1	1	1	5
VI	0.24	16.37	0.98	18.63	4.80	29.03	0.75	37.46	20.38	1	1	1	0	3
VII	0.17	16.37	0.63	18.64	3.72	28.58	3.61	38.00	20.38	1	1	0	1	3
VIII	0.80	16.29	1.30	17.93	1.01	27.5	6.8	40.51	-	-	-	-	-	-

of current with 1 μs time-step. The percentage THD for the current waveform simulated using a certain time-step is defined as

$$\%\text{THD} = \left(\sqrt{\sum_{h \neq 1} \left(\frac{I_h}{I_1} \right)^2} \right) 100. \qquad (25)$$

For the fixed time-step algorithm $(I_1)_{1\ \mu\text{s}} = 17.28$ A and the THD $= 16.28\%$. The ε and THD for different algorithms with step-sizes of 10, 50, 100, and 150 μs are also presented in Table II. The table also summarizes the number of operations required by each of the proposed algorithms. Algorithm I and II are the most rigorous and accurate while other algorithms bear relatively less computational burden and are therefore less accurate in comparison. The accuracy of these algorithms (III through VII) can be further improved through a correction of $\mathbf{x_2}$ state by interpolation. A remarkable improvement in performance especially in THD was observed when state $\mathbf{x_2}$ was corrected (as seen from Table II). The following conclusions can be drawn from the results:

- In the fixed time-step algorithm, as the time-step is increased from $\triangle t = 10$ μs to $\triangle t = 100$ μs, we find that the fundamental current increases drastically (from 17.1 A to 24.7 A), due to not accounting the switching events at their right instants, hence the percentage fundamental error ε increases from 2.4% to 33.66%. The harmonic current also increases at almost the same rate. Therefore, THD remains almost unchanged (around 16%). However, at much higher time-steps e.g., $\triangle t = 150$ μs, the harmonic current increases at a faster rate than the fundamental current, which is why we get a higher THD (46%) at that time-step.
- A time-step of over 50 μs (for the study system with a switching frequency of 1 kHz) is inadequate for simulation using the fixed time-step approach due to a high error (the error increases to 38.25% for $\triangle t = 150$ μs.
- As the time-step varies from $\triangle t = 10$ μs to $\triangle t = 100$ μs, Algorithm I treats the switching events accurately, thereby reducing both the fundamental and the harmonic currents. Therefore, both the percentage rms error and THD remain small (close to their values at $\triangle t = 10$ μs.
- For Algorithms II through VII, both THD and ε increase, with the increase of time-step, however, the increase in ε is less pronounced compared to that for the fixed time-step algorithm. Even at a smaller time-step such as $\triangle t = 10$ μs, the error is less than 0.8% for all the proposed algorithms, whereas fixed time-step algorithm shows an error of 2.4%. The ε decreases to a range of 0.5%–9% from 38.2% for the fixed time-step algorithm when $\triangle t = 150$ μs.
- Algorithms II through VII are less accurate in comparison with Algorithm I, and therefore produce significant harmonic current at higher time-steps. However, they are still able to compensate for the fundamental current and keep its value low. Therefore, the percentage rms error is small but the THD for these algorithms remains high at higher time-steps. Furthermore, Algorithms II to VII do not use the corrected state $\mathbf{x}_2$ as the the output at time t_2 in order to conform to the real-time condition that it would not be possible to send the corrected value to the controller.
- If state $\mathbf{x}_2$ is allowed to be corrected, a significant improvement in THD is observed. The results in Table II for a 150-μs time-step reveal that all algorithms (except Algorithm VIII) show THD in the range of 18% to 24%, whereas fixed time-step algorithm shows a THD of 46%. Offline simulators can exploit the benefit of correcting state $\mathbf{x_2}$.

For any given $\triangle t$, the accuracy of all the algorithms was found to be similar to that of the fixed time-step algorithm using one-tenth $\triangle t$. Using similar simulation parameters and environment, the algorithm used in HYPERSIM [19] has been extended for multiple switching events. A comparative study revealed that all the proposed algorithms produce results that are in close agreement with the results obtained using the HYPERSIM algorithm until $\triangle t = 100$ μs. At 10-, 50-, and 100-μs time-steps, the HYPERSIM algorithm yielded an error of 0.34, 0.98, and 4.6 (%) respectively, however, at 150 μs or higher it produced an increasing error (higher than 30%).

A comparison of steady-state current for fixed time-step algorithm and Algorithm VII is shown in Fig. 6. The current waveform with fixed time-step algorithm differs significantly when the time-step is changed from 10 μs to 150 μs. However Algorithm VII with 150 μs is in close conformity with the 10-μs case of fixed time-step algorithm. The difference is more obvious from the Fig. 7, where the frequency spectrum is plotted. Fixed time-step algorithm with $\triangle t = 150$ μs shows a substantial difference in fundamental component in comparison with

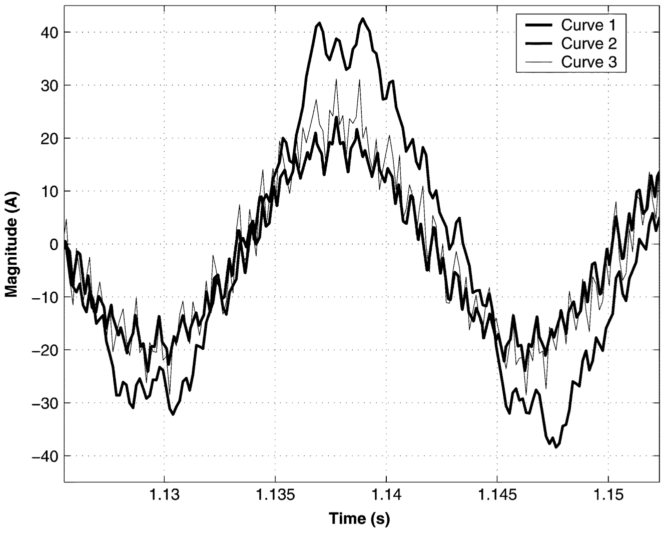

Fig. 6. Detailed view of the steady–state current I_a. Curve 1: Fixed time-step algorithm with $\triangle t = 10\ \mu$s, Curve 2: Fixed time-step algorithm with $\triangle t = 150\ \mu$s, Curve 3: Algorithm VII with $\triangle t = 150\ \mu$s.

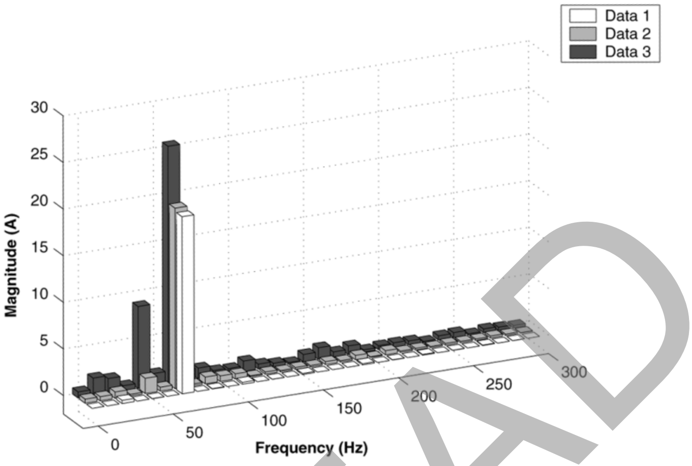

Fig. 7. Frequency spectrum of I_a. Data 1: Fixed time-step algorithm with $\triangle t = 10\ \mu$s, Data 2: Algorithm VII with $\triangle t = 150\ \mu$s, Data 3: Fixed time-step algorithm with $\triangle t = 150\ \mu$s.

the other two cases. It also induces noncharacteristic harmonics which eventually result in higher THD.

The behavior of various algorithms with the increase in $\triangle t$ can be further understood from Fig. 8 where the ε has been plotted with respect to $\triangle t$. It is clear that for any value of time-step, all the algorithms produce significantly less error than that of fixed time-step. Some algorithms show a little increase in error at a time-step of 100 μs. Algorithms I and III produce the best result for a time-step of 150 μs and as expected the error becomes high for other algorithms, however, it is much less than that of fixed time-step algorithm. Therefore, it can be concluded that all the algorithms substantially improve the simulation accuracy. The variation of THD as a function of time-step for various algorithms using corrected state $\mathbf{x_2}$ is shown in Fig. 9. Fixed time-step algorithm shows almost same THD until $\triangle t = 100\ \mu$s after which it increases dramatically. On the other hand, most of the algorithms (except Algorithm VIII) show a lower THD than the case of fixed time-step algorithm. Algorithm VIII produces a higher ε and THD compared to the fixed time-grid

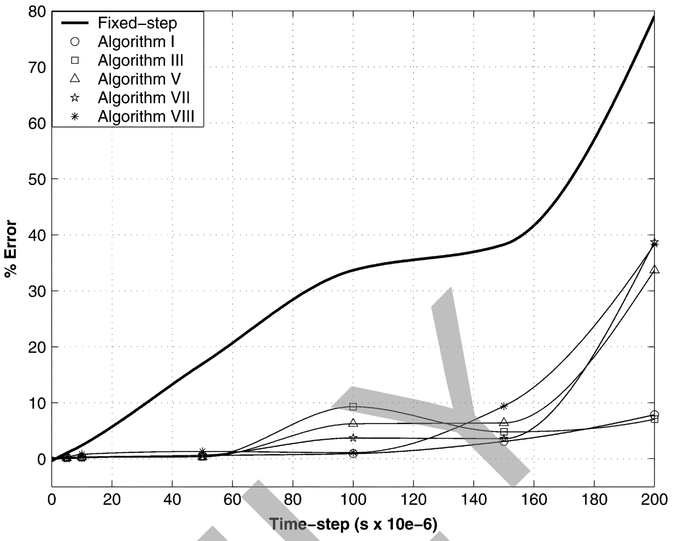

Fig. 8. Plot of percentage fundamental error ε with respect to simulation time-step.

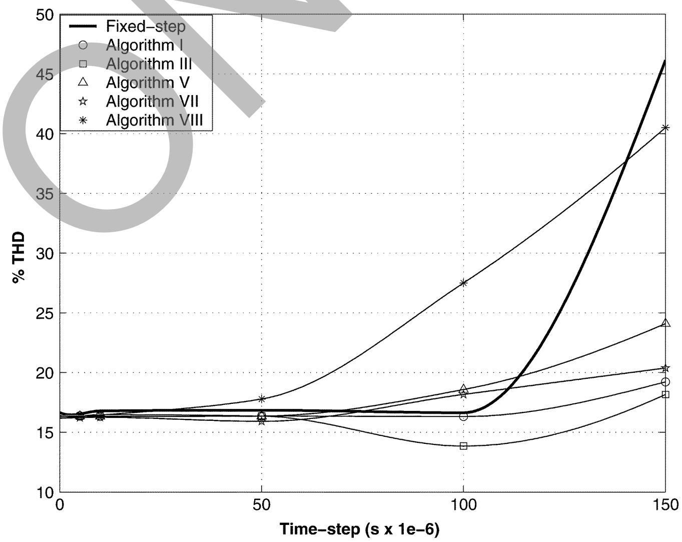

Fig. 9. Plot of THD with respect to simulation time-step.

algorithms for a large time-step of 150 μs or higher. The reason is that the algorithm resorts to frequent application of backward Euler and linear extrapolation (due to an increased step-size resulting in more switching events within a time-step) which is less accurate than the trapezoidal technique. However, it produces results as good as or even better than other algorithms at smaller time-steps (<100 μs).

B. Closed-Loop Control

Fig. 10 shows the D-STATCOM closed-loop response. A step response in i_q was simulated. The simulations were carried out for all the algorithms using various step-sizes, however, results are shown for the fixed step-size algorithm for $\triangle t = 10\ \mu$s and for $\triangle t = 100\ \mu$s and Algorithm VII for $\triangle t = 100\ \mu$s. The predominant effect observed due to an increase in the time step was the appearance of noise in the control quantities. As seen from Fig. 10, the traces of i_q for $\triangle t = 10\ \mu$s are smooth whereas those for $\triangle t = 100\ \mu$s contain a noise superimposed on the

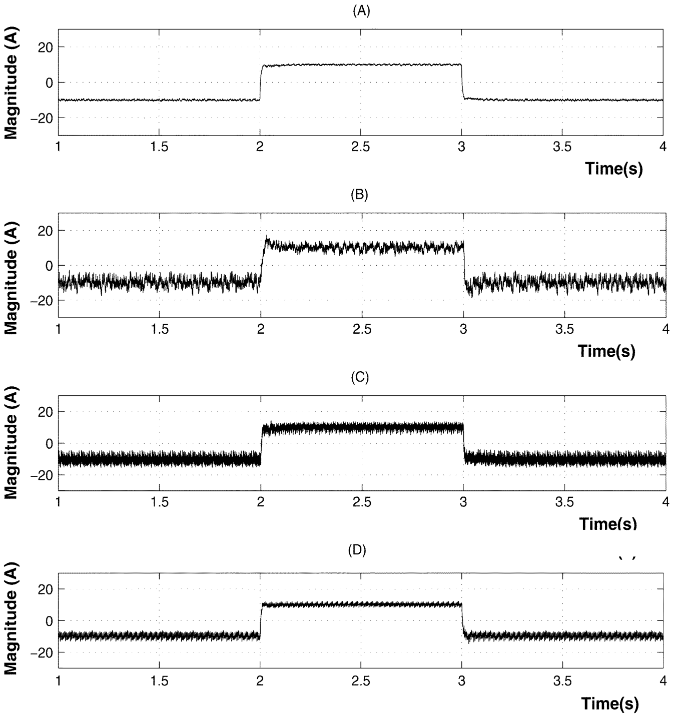

Fig. 10. Step response in i_q simulated using the (A) Fixed time-step algorithm with $\triangle t = 10\ \mu$s, (B) Fixed time-step algorithm with $\triangle t = 100\ \mu$s, (C) Algorithm VII with $\triangle t = 100\ \mu$s, and (D) Algorithm VII with $\triangle t = 100\ \mu$s but $\mathbf{x_2}$ corrected.

TABLE III
COMPARISON OF MEAN ABSOLUTE DEVIATION (MAD) AND SIGNAL-TO-NOISE RATIO (SNR) (dB) OF i_q PRODUCED BY DIFFERENT ALGORITHMS FOR $\triangle t = 100\ \mu$S

	$\mathbf{x_2}$ Uncorrected		$\mathbf{x_2}$ Corrected	
Algorithms	**MAD**	**SNR**	**MAD**	**SNR**
Fixed △t	1.79	29.32	-	-
I	-	-	0.05	101.4
II	1.32	34.8	0.06	96.32
III	1.37	35.32	0.52	55.24
IV	1.62	32.78	0.63	50.14
V	1.6	32.06	0.76	49.36
VI	1.61	33.54	0.65	49.38
VII	1.6	33.54	0.64	49.56
VIII	1.8	30.1	-	-

average value. A Fourier analysis proved that the noise is mainly white with unlimited bandwidth. Table III compares the mean absolute deviation (MAD) and the signal-to-noise ratio $(\mathrm{SNR} = 10\log_{10}(P_S/P_N)$ dB, where P_S is the signal power and P_N is the noise power) of i_q for all the algorithms. Algorithm I shows the best signal quality as it always uses the true state results. Results in Table III reveal that the signal quality deteriorates slightly as we move from Algorithm II to Algorithm VII. It can be seen that [Fig. 10(C)] for Algorithm VII using $\triangle t = 100\ \mu$s, the output i_q shows a MAD of 1.6 around its reference value (10 A), however, the fixed time-step algorithm shows a MAD of 1.79. The SNR for Algorithm VII is 33.54 dB and it is 29.32 dB for the fixed time-step algorithm. Table III reveals that if corrected states are used for control, MAD would be reduced to 0.64 and SNR would be increased to 49.56 dB. In a practical set-up of the digital controller the noise may be reduced by using low-pass anti-aliasing filters on the system signals before they are sampled. In an off-line simulation program the noise in the control quantities may also be substantially reduced by using linear interpolation in the control algorithm [12].

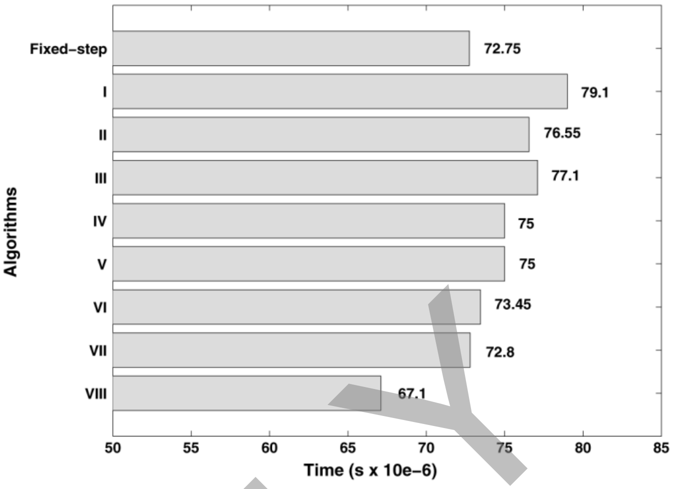

Fig. 11. Execution time for the time-step involving multiple switching events, for various algorithms with $\triangle t = 100\ \mu$s.

C. Execution Time

The execution time required for the off-line simulation of these algorithms depends on several factors such as the processor speed, hardware [such as random-access memory (RAM) and bus speed], and the operating system. Notwithstanding the minor variations in processor time due to multi-tasking, the testing conditions for all the algorithms were maintained the same. The simulation programs were coded in C and all the programs were executed on the same hardware using a 1.5-GHz Pentium IV processor running Windows 2000. The CPU time required to simulate the test system for the worst case of multiple switching (two switching events in one time-step) is shown in Fig. 11. The increased complexity in correcting the switching events using different algorithms increases the execution time by a maximum of only 9% (Algorithm I), compared to the fixed time-step approach. However, the improvement in the simulation accuracy is significant.

Using the proposed algorithms, an accurate simulation is possible even with a larger step-size, which makes these algorithms suitable for real-time application. Even though the difference in CPU execution time among different algorithms is not very large, it has been found that algorithms that do not involve rigorous calculations (II through VII) as compared to Algorithm I, requires less CPU time; for example, the CPU time requirement for Algorithm VII is 7.84% lower than that of Algorithm I. Algorithm VIII takes the least amount of time of all the proposed algorithms. The D-STATCOM system, due to its modeling simplicity is not able to show a larger CPU time difference between algorithms, however, a larger system with more modeling complexity such as multi-pulse HVDC system would be able to show definite computational advantages in favor of Algorithm II through VII. And there lies the motivation of using the proposed algorithms even with uncorrected states.

VI. Conclusion

This paper provides algorithms suitable for the accounting of multiple switching events accurately and efficiently in the digital simulation of power-electronic systems based on forced-commutated devices. A PWM VSC-based D-STATCOM system is used for simulation studies. Results obtained using these algorithms indicate excellent performance in comparison with the fixed time-step algorithm. The following points can be concluded:

- Algorithms clearly describe how two or three switching events are handled in digital simulation.
- A full comparison of the performance of the proposed algorithms has been presented. All algorithms show significant improvement in accuracy under both open-loop and closed-loop control.
- Despite enforcing real-time constraints such as having uncorrected states in the simulation, the performance of the proposed algorithms was found to be very good.
- It is important to note that the improvement in accuracy of the algorithms comes at a cost of only a 9% (maximum) increase in execution time in comparison with the fixed time-step algorithm.

The algorithms presented here are amenable for real-time implementation on a stand-alone digital signal processor (DSP) or general microprocessor platforms.

References

[1] D. Brandt, R. Wachal, R. Valiquette, and R. Wierckx, "Closed loop testing of a joint VAR controller using a digital real-time simulator," *IEEE Trans. Power Syst.*, vol. 6, no. 3, pp. 1140–1146, Aug. 1991.

[2] M. Kezunovic, V. Skendzic, M. Aganagic, J. K. Bladow, and S. M. McKenna, "Design, implementation and validation of a real-time digital simulator for protective relay testing," *IEEE Trans. Power Del.*, vol. 11, no. 1, pp. 158–164, Jan. 1996.

[3] D. Jakominich, R. Krebs, D. Retzmann, and A. Kumar, "Real time digital power system simulator design considerations and relay performance evaluation," *IEEE Trans. Power Del.*, vol. 14, no. 3, pp. 773–781, Jul. 1999.

[4] A. M. Gole and V. K. Sood, "A static compensator model for use with electromagnetic transient simulation programs," *IEEE Trans. Power Del.*, vol. 5, no. 3, pp. 1398–1405, Jul. 1990.

[5] J. M. Zahavir, J. Arrillaga, and N. R. Watson, "Hybrid electromagnetic transient simulation with the state variable representation for HVDC converter plant," *IEEE Trans. Power Del.*, vol. 8, no. 3, pp. 1591–1598, Jul. 1993.

[6] H. W. Dommel, "Digital computer solution of electromagnetic transients in single and multiphase networks," *IEEE Trans. Power App. Syst.*, vol. 88, no. PAS-4, pp. 388–399, Apr. 1969.

[7] J. R. Marti and L. R. Linares, "Real-time EMTP based transient simulation," *IEEE Trans. Power Syst.*, vol. 9, no. 3, pp. 1309–1317, Aug. 1994.

[8] S. Acevedo, L. R. Linares, J. R. Marti, and Y. Fujimoto, "Efficient HVDC converter model for real time transient simulation," *IEEE Trans. Power Syst.*, vol. 14, no. 1, pp. 166–171, Feb. 1999.

[9] J. R. Marti, S. Acevedo, L. R. Linares, H. W. Dommel, and Y. Fujimoto, "Accurate solution of HVDC converters in real-time transients simulation," in *Proc. Int. Conf. Power Systems Transients*, Seattle, WA, Jun. 1997, pp. 455–459.

[10] B. Kulicke, "Simulationsprogramm NETOMAC: Differenzleitwertverfahren bei kontinuierlichen und diskontinuierlichen Systemen," *Siemens Forschungs- und Entwicklungsberichte*, vol. 10, no. 5, pp. 299–302, 1981.

[11] T. L. Maguire and A. M. Gole, "Digital simulation of flexible topology power electronic apparatus in power systems," *IEEE Trans. Power Del.*, vol. 6, no. 4, pp. 1831–1840, Oct. 1991.

[12] P. Kuffel, K. Kent, and G. Irwin, "The implementation and effectiveness of linear interpolation in digital simulation," in *Proc. Int. Conf. Power Systems Transients*, Lisbon, Portugal, Sep. 1995, pp. 499–504.

[13] A. M. Gole, I. T. Fernando, G. D. Irwin, and O. B. Nayak, "Modeling of power electronic apparatus: Additional interpolation issues," in *Int. Conf. Power Systems Transients*, Seattle, WA, Jun. 1997, pp. 23–28.

[14] X. Lei, E. Lerch, D. Povh, and O. Ruhle, "A large integrated power system software package—NETOMAC," in *Proc. Int. Conf. Power System Technology*, vol. 1, Aug. 1998, pp. 17–22.

[15] J. Arrillaga, *Power Systems Electromagnetic Transients Simulation*. London, U.K.: Inst. Elect. Eng. Press, 2002, pp. 218–230.

[16] *PSCAD/EMTDC Brochure*, 2001.

[17] G. D. Irwin, D. A. Woodford, and A. Gole, "Precision simulation of PWM controller," in *Proc. Int. Conf. Power System Transients (IPST)*, 2001.

[18] B. D. Kelper, L. A. Dessaint, K. A Haddad, and H. Nakra, "A comprehensive approach to fixed-step simulation of switched circuits," *IEEE Trans. Power Electron.*, vol. 17, no. 2, pp. 216–224, Mar. 2002.

[19] V.-Q. Do, D. McCallum, P. Giroux, and B. D. Kelper, "A backward-forward interpolation technique for a precise modeling of power electronics in HYPERSIM," in *Proc. Int. Conf. Power Systems Transients*, Rio de Janeiro, Brazil, Jun. 2001, pp. 337–342.

[20] K. Strunz, L. R. Linares, J. R. Marti, O. Huet, and X. Lombard, "Efficient and accurate representation of asynchronous network structure changing phenomena in digital real-time simulators," *IEEE Trans. Power Syst.*, vol. 15, no. 2, pp. 586–592, May 2000.

[21] V. R. Dinavahi, M. R. Iravani, and R. Bonert, "Real-time digital simulation of power electronic apparatus interfaced with digital controllers," *IEEE Trans. Power Del.*, vol. 16, no. 4, pp. 775–781, Oct. 2001.

[22] V. R. Dinavahi and M. R. Iravani, "Coupling digital controllers with real-time digital simulators for switched power systems," in *Proc. Int. Conf. Power Systems Transients*, Rio de Janeiro, Brazil, Jun. 2001.

[23] M. Zou, J. Mahseredjian, G. Joos, B. Delourme, and L. G. Lajoie, "Interpolation and reinitialization for the simulation of power electronic circuits," in *Proc. Int. Conf. Power System Transients*, Sep. 28–Oct. 2, 2003.

[24] A. M. Gole, A. Keri, C. Nwankpa, E. W. Gunther, H. W. Dommel, I. Hassan, J. R. Marti, J. A. Martinez, K. G. Fehrle, L. Tang, M. F. McGranaghan, O. B. Nayak, P. F. Ribeiro, R. Iravani, and R. Lasseter, "Guidelines for modeling power electronics in electric power engineering applications," *IEEE Trans. Power Del.*, vol. 12, no. 1, pp. 505–514, Jan. 1997.

[25] C. Schauder and H. Mehta, "Vector analysis and control of advanced static VAR compensators," *Proc. Inst. Elect. Eng. C*, vol. 140, no. 4, pp. 299–306, Jul. 1993.

[26] J. Holtz, "Pulsewidth modulation—A survey," *IEEE Trans. Ind. Electron.*, vol. 39, no. 5, pp. 410–419, Dec. 1992.

M. O. Faruque (S'03) received the B.Sc.Engg. degree from Chittagong University of Engineering and Technology (erstwhile BITC), Bangladesh, in 1992, and the M.Eng.Sc. degree in power engineering from the University of Malaya, Malaysia, in 1999. He is currently pursuing the Ph.D. degree in the Department of Electrical and Computer Engineering at the University of Alberta, Edmonton, AB, Canada.

His research interests include FACTS, HVDC, and real-time digital simulation of power electronics and power systems.

Venkata Dinavahi (M'00) received the B.Eng. degree from Nagpur University, India, in 1993, and the M.Tech. degree from the Indian Institute of Technology, Kanpur, India, in 1996. He received the Ph.D. degree in electrical and computer engineering from the University of Toronto, Toronto, ON, Canada, in 2000.

Currently, he is an Assistant Professor at the University of Alberta, Edmonton, AB, Canada. He is a Professional Engineer in the Province of Alberta. His research interests include electromagnetic transient analysis, power electronics, and real-time digital simulation and control.

Wilsun Xu (M'90–SM'95) received the Ph.D. degree from the University of British Columbia, Vancouver, BC, Canada, in 1989.

Currently he is a Professor with the University of Alberta, Edmonton, AB, Canada. He was an Engineer with BC Hydro, Vancouver, BC, Canada, from 1990 to 1996. His main research interests are power quality and harmonics.