

An Efficient Switch Model for Simulating Large Power Systems with Many Power Converters

Fabian M. Uriarte and Salman Mashayekh

Abstract—Frequent matrix factorizations due to power electronic switch commutations are computationally expensive. This paper addresses this burden by treating power electronic switches as dependant sources instead of time-varying resistors. The resulting network matrix is constant, and does not require re-factorization when switches commute. Three power electronic switches are presented and discussed for both nodal and mesh equation formulations. A case study at the end shows that the run-time for a power system with >200 switches when using the switching approach presented herein can reduce from 8 hours to under 15 minutes, which is a speedup of 40x.

Index Terms—diode, effective, power, matrix, mesh, model, network, time, shipboard, simulation, switch, time.

I. INTRODUCTION

A burdensome aspect of power system simulations with multiple power electronic switches is the refactoring of the network matrix when power electronic switches commute. Refactoring a network matrix implies accessing and updating its entries, and re-computing the LU factors before advancing the simulation time. Testimony of this bottleneck can be gathered from the speed at which power system simulations with multiple power converters take place; particularly when the network matrix order nears or exceeds $O(10^3)$. Moreover, as the number of power electronic switches in a power system simulation increases, refactoring occurs more often, which contributes to increasingly run-times.

Attempts at time domain simulations with constant network matrices have not found a suitable application in power system simulations. For example, in [1],[2], power electronic switches were modeled as parasitic inductances and capacitances in the *on* and *off* states, respectively. This approach resulted in a constant network matrix where only the right-hand side was updated when a switch commutated. A requirement for this approach, however, is that the simulation time step increment Δt must be small (e.g., $\Delta t = 2 \times 10^{-9}$ s) to avoid current through switches that are *off*. Simulations with small Δt counteract efficiencies gained from this modeling approach, and may not be appropriate in power system simulations where typically $\Delta t = 50 \times 10^{-6}$ s.

An approach that maintains the network matrix *quasi*-constant is the one used by *PSCAD* [3],[4]. This approach

starts the LU re-factorization from the diagonal entry corresponding to the lowest-numbered node incident to a commutating switch, and avoids re-factorization of the *entire* network matrix. This approach is well suited for power system simulations; however, it requires control over the LU factorization routine to indicate the starting diagonal entry. From a pragmatic, non-commercial perspective, it is common (and preferable at first) to use available math libraries [5] (e.g., for .NET or Java, or a standard program such as *MATLAB* as a simulation kernel). In such cases, the starting diagonal entry of the LU factorization algorithm cannot be controlled, which makes *PSCAD*'s algorithm difficult to integrate in new programs.

This paper presents a switch model that creates a constant network matrix. When a switch commutates, the right-hand side of the equations are updated with the superposition of a new vector term, which is re-calculated only when a switch commutates. Section II presents the three switch models used in this paper to exemplify the switching approach. Section III presents the network equations required for time domain simulations. Section IV suggests a sequence to solve said equations. Section V presents two case studies: an illustrative three-phase inverter case, and a larger case to present the possible run-time speedup in larger power systems. Finally, the paper is concluded in chapter VI.

II. SWITCH MODELS

The power system equations typically used in time domain (i.e., EMTP [6]) simulations are given in (1). (The common superscripts k and $k+1$ normally used to indicate the time step number are elided for brevity.) In (1), \mathbf{A}_{orig} represents the network immittance matrix (i.e., mesh resistance or nodal conductance matrix), \mathbf{x}_{orig} the vector of unknown (mesh currents or node voltages), and \mathbf{b}_{orig} the source vector (voltage impressions or current injections). The subscripts *orig* will help distinguish the terms in (1) from those that will be presented.

$$\mathbf{A}_{orig} \mathbf{x}_{orig} = \mathbf{b}_{orig} \quad (1)$$

A switch model where \mathbf{A}_{orig} (or a variant of it) remains constant in time domain simulations is proposed. A constant network matrix is possible if the switches are modeled as time-varying *sources* instead of time-varying *resistors* as is typical. (In the remainder of this paper, modeling switches as time-varying resistors will be termed the *classical* approach.)

If sources are used to represent switches, topology changes modify \mathbf{b}_{orig} instead of \mathbf{A}_{orig} . In this new approach, the formulation takes the form of (2), where $\mathbf{A}_{const} \neq \mathbf{A}_{orig}$ is a

F. M. Uriarte is with the Center for Electromechanics at The University of Texas at Austin, Austin, TX, USA 78758. S. Mashayekh is with the Dept. of Electrical and Computer Engineering at Texas A&M University, College Station, TX, USA 77843 (e-mails: f.uriarte@cem.utexas.edu, s.mashayekh@tamu.edu)

variant of \mathbf{A}_{orig} , or *new* constant-coefficient network matrix, and \mathbf{b}_{switch} a *new* vector term containing the contributions of all switches. For example, in the case of a *mesh* formulation, closed switches having zero voltage drops across them contribute a zero-voltage impression to \mathbf{b}_{switch} ; open switches a voltage impression necessary to counter-act (i.e., cancel out) a switch's through current. In the case of a *nodal* formulation, closed switches sink necessary current in \mathbf{b}_{switch} to satisfy their through current, whereas open switches sink zero amps.

$$\mathbf{A}_{const} \mathbf{x}_{orig} = \mathbf{b}_{orig} + \mathbf{b}_{switch} \quad (2)$$

The switch model presented here abstracts the time-varying property of a switch as a *source*. This paper presents three switch types to exemplify this modeling approach presented herein: a diode, a diode with an RC snubber, and an IGBT with diode and snubber. These switch types are shown atop Fig. 1. Referring to Fig. 1a, in the classical approach, diodes are modeled as time-varying resistances with a series *on*-voltage source. The resistance R_D represents the state of the switch: $R_D = R_{on}$ when the diode is *on*, or $R_D = R_{off}$ when the diode is *off*. (Common values for R_{on} and R_{off} are 1 m Ω and 1 M Ω , respectively.)

The classical switching approach is computationally expensive because when a switch (e.g., a diode) toggles, re-factorization of \mathbf{A}_{orig} in (1) is required. This re-factorization can dominate the solution time at each time step, more so when the following conditions are met: hundreds of switches are in the power system model, when the simulation *end* time requires $\geq O(10^3)$ integration time steps, and when \mathbf{A}_{orig} is large ($\geq O(10^3)$). It is not unusual for these conditions to occur simultaneously [7]. One example is the simulation of all-electric ships, where their simulation run-times are too time consuming to be practical [7-9].

The next subsections present the switch models for nodal and mesh formulations. The explanations start from the basic case of a diode. Then, the diode is appended with a parallel RC (resistive-capacitive) snubber, and finally, with an anti-parallel switch. This final arrangement is commonly known as an IGBT with anti-parallel diode and snubber. The sequence in which these switches are introduced is pedagogic, and incremental in complexity. This chosen sequence facilitates documentation and the explanation of the modeling technique presented in this paper.

A. Nodal formulations

1) Diode

The following explanations make reference to Fig. 1a, Fig. 1d, and Fig. 1e. What is proposed is a *decomposition* of the equivalent resistance R_D into two parallel resistances: R_1 , which is constant, and R_2 , which is time-varying. Resistance R_1 does not change through the simulation and is stamped to the network matrix \mathbf{A}_{const} in (2). To avoid re-factorizing \mathbf{A}_{const} when the diode toggles, R_2 is enclosed in the current source i_2 , and modifies \mathbf{b}_{switch} instead of \mathbf{A}_{const} .

Referring to Fig. 1d and Fig. 1e, $I_{eq}=I_D$ represents the *on*-

voltage source V_D paralleled with R_D . The values of V_D are typically 1 V when *on*, and 0 V when *off*. This Thévenin-Norton transformation eliminates the voltage source, which would unnecessarily increase the order of the nodal conductance matrix by one (modified nodal analysis [10] is avoided to maintain the symmetric-positive-definiteness property in \mathbf{A}_{const}).

The decomposition of $R_{pq}=R_D$ into R_1 and R_2 results in the values shown in Fig. 1e. These values for R_1 and R_2 , when paralleled together, return the original across resistance across nodes p and q as expected. (Other combinations of R_1 and R_2 may be used as long as their parallel combination returns R_D .) Although the entire switch resistance R_{pq} may be enclosed in i_2 , a static term R_1 is necessary to prevent singularities from nodes with only sources connected to it.

2) Diode with Snubber

The following explanations make reference to Fig. 1b, Fig. 1d, and Fig. 1f. In practice, diodes are commonly paralleled with a protective snubber RC snubber. (The discretization of a snubber RC branch is provided in the Appendix). After discretizing the snubber branch, the total resistance across nodes p and q is $R_{pq} = R_1 || R_2 = R_{RC} || R_D$, which results in the values of R_1 and R_2 in Fig. 1f. These values for R_1 and R_2 are different than in the diode case, where R_D was decomposed into R_1 and R_2 ; here, the *combined* resistance R_{pq} is decomposed instead.

The equivalent current source observed between nodes p and q is I_{eq} , which accounts for the influence of *both* the snubber branch historical source I_{histRC} and I_D . The value of I_{eq} is the sum of these two current sources.

3) IGBT with Diode and Snubber

The following explanations make reference to Fig. 1c, Fig. 1d, and Fig. 1g. An IGBT results if the diode and snubber of Fig. 1b is anti-paralleled with another switch, which results in three parallel branches. The decomposition of R_{pq} for the IGBT results in the values of R_1 and R_2 given in Fig. 1g. It is important to note that in the IGBT case, there are two switch branches. As a result, *both* the diode resistance R_D and IGBT resistance R_Q form R_2 .

The equivalent current source I_{eq} accounts for the snubber branch historical current injection, the *on*-voltage of the diode, and the *on*-voltage of the IGBT. The IGBT is oriented having its positive node as q , while the diode and its snubber are oriented with their positive node at p . This arrangement causes the subtractions for I_{eq} in Fig. 1g.

The switch model shown in Fig. 1d is common to the three switch types in nodal formulations. This model can be used to replace diodes, diodes with snubbers (e.g., rectifiers), or IGBTs (e.g., inverters). (A three-phase inverter will be exemplified later in this paper.)

B. Mesh formulations

It is not uncommon to formulate power system in mesh formulations. This section presents the three switch types in a form that is readily suited for such formulations.

1) Diode

The following explanations make reference to Fig. 1a, Fig. 1h, Fig. 1i. Similar to the nodal case, the $R_{pq} = R_D$ is decomposed into R_1 and R_2 . Resistance R_1 does not change throughout the simulation, and is stamped to \mathbf{A}_{const} in (2). R_2 , for mesh formulations, is treated as a dependent *voltage* source v_2 enclosing R_2 . Since all voltage sources enter \mathbf{b}_{switch} , changes to v_2 prevent modifications to \mathbf{A}_{const} .

The decomposition of R_{pq} into R_1 and R_2 results in the values shown in Fig. 1i. These values for R_1 and R_2 , when added together, return the original across resistance across nodes p and q as expected. (Other combinations of R_1 and R_2 may be used as long as their sum returns R_D .) Although the entire switch resistance R_{pq} may be enclosed in v_2 , a static term R_1 is necessary to prevent zero-resistance meshes in switch loops as observed bridge converters. Referring to V_{eq} in Fig. 1h and V_D in Fig. 1a, as the diode changes state, V_{eq} takes on the value of V_D to model the *on*-voltage drop. (The values of V_D are typically 1 V when *on*, and 0 V when *off*.)

2) Diode with snubber

The following explanations make reference to Fig. 1b, Fig. 1h, and Fig. 1j. After discretizing the snubber branch and performing Norton-ThéVénin transformations, the equivalent model in Fig. 1h is obtained. The decomposition of R_{pq} results in the values of R_1 and R_2 shown in Fig. 1j, which are different than for R_1 and R_2 for the diode alone. It should be noted that $R_{pq} = R_1 + R_2$ *always* holds true; however, R_{pq} differs in each switch model. For example, in the diode $R_{pq} = R_1 + R_2 = R_D$, whereas in the diode with snubber $R_{pq} = R_1 + R_2 = R_{RC} || R_D$. The equivalent voltage source observed between nodes p and q is V_{eq} . This voltage source includes the influence of the snubber branch and of the diode—regardless of diode state. The afore values are noted at the bottom of Fig. 1i and Fig. 1j, respectively.

3) IGBT with Diode and Snubber

The following explanations make reference to Fig. 1c, Fig. 1h, and Fig. 1k. The arrangement of an IGBT with a diode and snubber results in three parallel branches. In mesh formulations, this unnecessarily adds two meshes for every IGBT, which can rapidly increase the order of the system in the presence of multiple power converters. As with the diode and snubber, the model shown in Fig. 1h reduces the trapped meshes, and is a welcome benefit from using the proposed switch model in mesh formulations.

C. Implementation Notes

The following notes highlight items requiring attention during implementation.

- Referring to Fig. 1d, the diode's voltage v_{pq} is known from the network solution. To compute i_{pq} , v_{pq} is multiplied by the conductance $G_{pq} = 1/R_{pq}$ and added to $-I_{eq}$. The current source i_2 does *not* enter the calculation of i_{pq} as it is

implicit in the product $v_{pq}G_{pq}$ (i.e.,

$$i_{pq} = v_{pq}G_{pq} - I_{eq} = v_{pq}(G_1 + G_2) - I_{eq} = v_{pq}G_1 + i_2 - I_{eq}.$$

- During implementation, the diode's current i_{pq} is known from the network's mesh vector. To compute the diode's across voltage v_{pq} , i_{pq} is multiplied by R_{pq} and added to V_{eq} as shown in Fig. 1h. The voltage source v_2 need *not* enter the calculation of v_{pq} as it is implicitly included in the multiplication of i_{pq} times R_{pq} (i.e., $v_{pq} = i_{pq}R_{pq} + V_{eq} = i_{pq}(R_1 + R_2) + V_{eq} = i_{pq}R_1 + v_2 + V_{eq}$).
- To update the historical term of the snubber RC branch, the current through the RC branch must be computed *explicitly*, and be saved at every time step. Furthermore, I_{eq} and V_{eq} in the diode with snubber and IGBT models must be updated at *each* time step as they include the influence of the snubber's capacitor (not just *on*-voltages alone).
- During time interpolation to assess the instants of switch commutations, the branch currents of the diode (i_D) and IGBT (i_Q) must be computed explicitly since $i_{pq} \neq i_D \neq i_Q \neq i_{RC}$, and must also be saved for use at the next time step.
- Similarly in the IGBT model, when checking if the diode or IGBT voltages exceed their turn-on voltages, it should be noted that $v_{pq} = v_D = v_{RC} = -v_Q$. The negative sign on the IGBT voltage is due to the orientation of the IGBT switch branch from node q to node p . The diode and snubber are oriented from node p to node q .
- When issuing a toggle command to a switch, it must be clearly specified whether the toggle is meant for the diode part, the IGBT, or *both* simultaneously.
- During interpolations, v_{pq} should be used to assess whether the diode and/or the IGBT have violated their turn-on voltages (not i_D or i_Q).
- To compute v_{pq} during mesh formulations, i_{pq} must be computed first using $i_{pq} = \sum i_p - \sum i_q$ in Fig. 1h. This expression indicates that the total switch current i_{pq} consist of several meshes: some entering node p , and some entering node q . After computing i_{pq} , v_{pq} can be found and interpolation can continue as for the nodal formulation case.

III. NETWORK FORMULATION

This section explains how to formulate the power system equations for a time domain simulation when the proposed model is used. If the contributions of all switch R_2 terms (wrapped as i_2 or v_2 , depending on the formulation chosen) are arranged in a vector \mathbf{y} , augmenting (2) with these unknown sources results in (3).

$$\underbrace{\begin{bmatrix} \mathbf{A}_{const} & \mathbf{V} \\ \mathbf{W}^T & -\mathbf{S}^{-1} \end{bmatrix}}_{\mathbf{A}_{large}} \underbrace{\begin{bmatrix} \mathbf{x}_{orig} \\ \mathbf{y} \end{bmatrix}}_{\mathbf{x}_{large}} = \underbrace{\begin{bmatrix} \mathbf{b}_{orig} \\ \mathbf{0} \end{bmatrix}}_{\mathbf{b}_{large}} \quad (3)$$

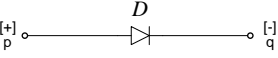
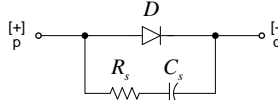
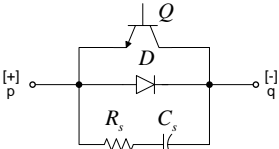
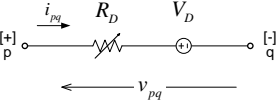
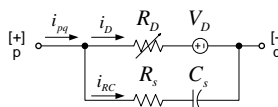
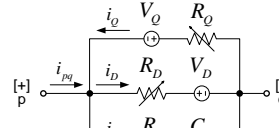
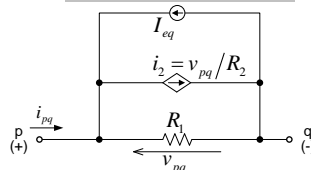
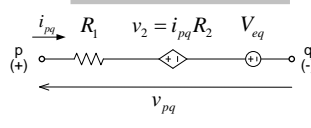
		Classical switch models		
Switch type				
	$R_D = R_{on} \text{ or } R_{off}; V_D = V_{on} \text{ or } V_{off}$	$R_D = R_{on} \text{ or } R_{off}; V_D = V_{on} \text{ or } V_{off}$	$R_D = R_{on} \text{ or } R_{off}; V_D = V_{on} \text{ or } V_{off}$ $R_Q = R_{on} \text{ or } R_{off}; V_Q = V_{on} \text{ or } V_{off}$	
Circuit model				
	(a)	(b)	(c)	
Proposed switch models		Equations		
Nodal formulations				
	$R_1 = 2R_{off} \text{ (const.)}$ $R_2 \rightarrow \begin{cases} = \frac{2R_{off}R_{on}}{2R_{off} - R_{on}} & \text{when on} \\ = R_1 & \text{when off} \end{cases}$ $R_{pq} = R_D; I_{eq} = I_D = \frac{V_D}{R_D}$ $i_D = i_{pq}$	$R_1 = R_{RC} \text{ (const.)}$ $R_2 = R_D$ $R_{pq} = R_{RC} // R_D$ $I_{eq} = I_{histRC} + I_D = I_{histRC} + V_D / R_D$ $i_D = (v_{pq} - V_D) / R_D$ $i_{RC} = i_{pq} - i_D$	$R_1 = R_{RC} \text{ (const.)}$ $R_2 = R_D // R_Q$ $R_{pq} = R_{RC} // R_D // R_Q$ $I_{eq} = I_{histRC} + I_D - I_Q = I_{histRC} + \frac{V_D}{R_D} - \frac{V_Q}{R_Q}$ $i_D = (v_{pq} - V_D) / R_D; i_Q = (-v_{pq} - V_Q) / R_Q$ $i_C = i_{pq} + i_Q - i_D$	
$R_{pq} = \frac{R_1 R_2}{R_1 + R_2}; v_{pq} = v_p - v_q$ $i_{pq} = v_{pq} / R_{pq} - I_{eq} \text{ (d)}$	(e)	(f)	(g)	
Mesh formulations				
	$R_1 = \frac{R_{on}}{2} \text{ (const.)}$ $R_2 \rightarrow \begin{cases} = R_1 & \text{when on} \\ = R_{off} - R_1 & \text{when off} \end{cases}$ $R_{pq} = R_D$ $V_{eq} = V_D$	$R_1 = R_{RC} \text{ (const.)}$ $R_2 = \frac{-R_{RC}^2}{R_{RC} + R_D}; R_{pq} = R_{RC} // R_D$ $V_{eq} = R_{pq} \left(\frac{V_{histRC}}{R_{RC}} + \frac{V_D}{R_D} \right)$ $i_D = (v_{pq} - V_D) / R_D$ $i_{RC} = i_{pq} - i_D$	$R_1 = R_{RC} \text{ (const.)}$ $R_2 = \frac{-R_{RC}^2 (R_D + R_Q)}{R_D R_Q + R_{RC} (R_D + R_Q)}$ $V_{eq} = R_{pq} \left(\frac{V_{histRC}}{R_{RC}} + \frac{V_D}{R_D} - \frac{V_Q}{R_Q} \right)$ $i_D = (v_{pq} - V_D) / R_D; i_Q = (-v_{pq} - V_Q) / R_Q$ $i_C = i_{pq} + i_Q - i_D$	
$R_{pq} = R_1 + R_2; i_{pq} = \sum i_p - \sum i_q$ $v_{pq} = i_{pq} R_{pq} + V_{eq} \text{ (h)}$	(i)	(j)	(k)	

Fig. 1 Classical and proposed switch models for nodal and mesh formulations.

In (3) \mathbf{x}_{orig} and \mathbf{y} are the unknown vectors of the system. If n represents the number of variables in a system (i.e., node voltages or mesh currents), and w the number of switches, then \mathbf{x} and \mathbf{y} are $n \times 1$ and $w \times 1$ vectors, respectively.

\mathbf{A}_{const} is the $n \times n$ network coefficient matrix. Depending on the network formulation type, this matrix is either the *nodal conductance matrix* (nodal formulations) or the *mesh resistance matrix* (mesh formulations). It is emphasized that $\mathbf{A}_{const} \neq \mathbf{A}_{orig}$. The difference is that \mathbf{A}_{orig} contains the total switch resistance terms $R_{pq} = R_1 + R_2$, whereas \mathbf{A}_{const} only contains the R_1 terms of each switch. This makes \mathbf{A}_{const} constant.

\mathbf{V} is a unit-less $n \times w$ matrix that couples each network variable to a switch source as:

$$\mathbf{V}(i, j) \rightarrow \begin{cases} = 1 & \text{if mesh } i \text{ enters switch } j \\ = -1 & \text{if mesh } i \text{ leaves switch } j \\ = 0 & \text{otherwise.} \end{cases} \quad (4)$$

Matrices \mathbf{V} and \mathbf{W} are the same, but noted differently as per the convention suggested in [11]. \mathbf{S} is a $w \times w$ diagonal,

immittance matrix containing the R_2 contributions as defined by (5). Depending on the formulation type, the *immittance* may be a resistance (for nodal formulations) or a conductance (for mesh formulations). When the i^{th} switch toggles, only the i^{th} diagonal in \mathbf{S} is updated.

$$\mathbf{S}(i, j) \rightarrow \begin{cases} s_{ii} = \text{time-varying immittance of switch } i \\ s_{ij} = 0 \end{cases} \quad (5)$$

Equation (3) can be explained as matrix equation in (6) and (7). Equation (6) is a set of Kirchhoff Current Law (KCL) equations in nodal formulations, or Kirchhoff Voltage Law (KVL) equations in mesh formulations. The $\mathbf{V} \cdot \mathbf{y}$ term couples the network variables to the unknown switch sources grouped in \mathbf{y} . Equations in (7) are *constraint* equations that use \mathbf{x}_{orig} to express the state of the switch in terms of the immittances in \mathbf{S} .

$$[\mathbf{A}_{const} \quad \mathbf{V}] \begin{bmatrix} \mathbf{x}_{orig} \\ \mathbf{y} \end{bmatrix} = \mathbf{b}_{orig} \quad (6)$$

$$\begin{bmatrix} \mathbf{W}^T & -\mathbf{S}^{-1} \end{bmatrix} \begin{bmatrix} \mathbf{x}_{orig} \\ \mathbf{y} \end{bmatrix} = \mathbf{0} \quad (7)$$

IV. SOLUTION METHODOLOGY

This section presents a solution approach to solve (3) such that the benefits of using the proposed switch model are manifested.

A. Equations of Solution

Solving for \mathbf{x} in (6), and for \mathbf{y} in (7), results in:

$$\mathbf{x}_{orig} = \mathbf{A}_{const}^{-1} \mathbf{b}_{orig} - \mathbf{A}_{const}^{-1} \mathbf{V} \mathbf{y} \quad (8)$$

$$\mathbf{S}^{-1} \mathbf{y} = \mathbf{W}^T \mathbf{x}_{orig} \quad (9)$$

Substituting \mathbf{x}_{orig} from (8) into (9), and then solving for \mathbf{y} results in the equation pair in (10). In matrix theory (10) is known as Woodbury's method for inverting modified matrices [11-13], which is also of the form of Kron's diakoptics [13],[14], and Ho's modified nodal analysis [10],[15]. A solution of this type is known to be computationally effective when compared to (1).

$$\text{Network solution} \begin{cases} \mathbf{x}_{orig} = \underbrace{\mathbf{A}_{const}^{-1} \mathbf{b}_{orig}}_{\mathbf{x}_1} - \underbrace{\mathbf{A}_{const}^{-1} \mathbf{V} \mathbf{y}}_{\mathbf{x}_2} & (a) \\ \mathbf{y} = \underbrace{(\mathbf{S}^{-1} + \mathbf{W}^T \mathbf{A}_{const}^{-1} \mathbf{V})^{-1}}_{\mathbf{K}} \mathbf{W}^T \mathbf{x}_1 & (b) \end{cases} \quad (10)$$

The following characteristics are noted for (10)a:

- The solution of \mathbf{x}_{orig} is found from the superposition of two vectors \mathbf{x}_1 and \mathbf{x}_2
- Vector \mathbf{x}_1 is an *interim* solution with all switch dependant sources at a zero value
- Vector \mathbf{x}_2 is an interim solution with *only* the switch dependant sources energizing the network
- Matrices \mathbf{A}_{const} and $\mathbf{V}=\mathbf{W}$ are *constant*
- Vector \mathbf{b}_{orig} is independent of switch states
- Matrix \mathbf{K} is updated only if a switch toggles

The following characteristics are noted for (10)b:

- The product $\mathbf{W}^T \mathbf{A}_{const}^{-1} \mathbf{V}$ is also constant
- In $\mathbf{W}^T \mathbf{x}_1$, \mathbf{x}_1 is *passed-in* from (10)a
- The only time-varying matrix is \mathbf{S} , which changes *only* when a switch commutates
- Only *one* entry (per switch) changes in \mathbf{S} when a switch toggles

B. Implementation Notes

Equation (10) suggests using matrix inverses. A more effective computer implementation is provided as the following four-step solution [11] in (11):

$$\begin{cases} 1. & \mathbf{A}_{const} \mathbf{x}_1 = \mathbf{b}_{orig} \\ 2. & \mathbf{K} \mathbf{y} = \mathbf{W}^T \mathbf{x}_1 \\ 3. & (\mathbf{A}_{const} \mathbf{V}) \mathbf{x}_2 = \mathbf{y} \\ 4. & \mathbf{x}_{orig} = \mathbf{x}_1 - \mathbf{x}_2 \end{cases} \quad (11)$$

Step 1) At every time step of a time domain simulation, the system is solved for \mathbf{x}_1 . Since \mathbf{A}_{const} is constant, so are its LU factors, which are stored prior to starting a simulation

Step 2) \mathbf{y} is solved for using \mathbf{x}_1 found in Step 1). Only if a switch toggles, \mathbf{S} and \mathbf{K} are updated in (10)b *prior* to solving for \mathbf{y}

Step 3) Vector \mathbf{x}_2 is computed noting that the LU factors for $\mathbf{A}_{const} \mathbf{V}$ are also constant

Step 4) The system solution is found from the superposition of \mathbf{x}_1 and \mathbf{x}_2 as given in (10)a

It may appear that the three LU solutions in (11) involve more work than refactoring \mathbf{A}_{orig} after a switch toggle. This may be the case in simulations with un-frequent commutations. In simulations with hundreds of switches, switches commute often. The three LU solutions in (11) at each step require less effort than re-factorizing \mathbf{A}_{orig} .

The burden of the proposed switch model lies in refactoring \mathbf{K} when a switch commutates. However, re-factorization of \mathbf{K} is less burdensome than the re-factorization of \mathbf{A}_{orig} as the dimensions of \mathbf{K} are significantly smaller than those of \mathbf{A}_{orig} . This is demonstrated with the final case study presented by this paper.

An advantage of the formulation provided is that every time a switch toggles, only *one* entry in \mathbf{S} is updated. In the classical switch model, in a nodal formulation, four matrix elements in \mathbf{A}_{orig} are accessed and updated when a switch commutates (assuming an ungrounded switch). In mesh formulations, m^2 elements are accessed and updated for every switch that commutates, where m is the number of meshes that traverse the switch (e.g., when $m=4$ meshes traverse a switch, 16 matrix entries are updated each time said switch toggles). (Frequent matrix-entry access times may cause bottlenecks, and depend on the programming efficiency and matrix storage implemented in a program.)

V. CASE STUDIES

Two case studies are presented. The first case exemplifies how to formulate the network matrices of a three-phase inverter in a nodal and mesh formulation. The second case study examines the speed gain observed from a large-scale power system simulation that is computationally complex.

A. Case 1: Three-Phase Inverter

Consider a three-phase inverter driven by an ideal DC voltage source V_{DC} , and loaded by a wye-connected resistive load (R_a , R_b , and R_c). Replacing each inverter IGBT with the model shown in Fig. 1d and Fig. 1h results in the equivalent circuits shown in Fig. 2 and Fig. 3 for a nodal and mesh formulation, respectively.

Referring to Fig. 2, the network matrices corresponding to a nodal formulation are given in (12)-(17), where $G_{li} = 1/R_{li}$ represents the conductance of R_{li} for the i^{th} switch.

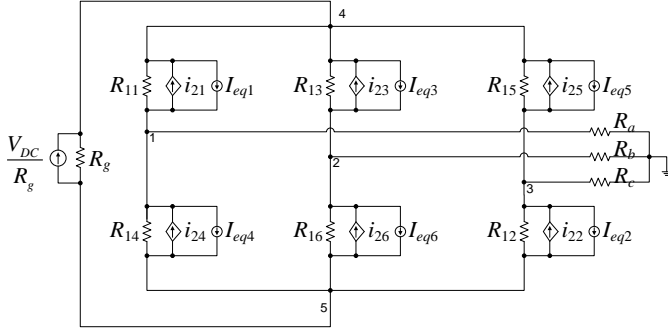


Fig. 2. Voltage-source inverter show for a nodal formulation

$$\mathbf{A}_{const} = \begin{bmatrix} \begin{pmatrix} G_{11} + G_{14} \\ +G_a \end{pmatrix} & \cdot & \cdot & -G_{11} & -G_{14} \\ \cdot & \begin{pmatrix} G_{13} + G_{16} \\ +G_b \end{pmatrix} & \cdot & -G_{13} & -G_{16} \\ \cdot & \cdot & \begin{pmatrix} G_{15} + G_{12} \\ +G_c \end{pmatrix} & -G_{15} & -G_{12} \\ -G_{11} & -G_{13} & -G_{15} & \begin{pmatrix} G_g + G_{11} \\ +G_{13} + G_{15} \end{pmatrix} & -G_g \\ -G_{14} & -G_{16} & -G_{12} & -G_g & \begin{pmatrix} G_g + G_{14} \\ +G_{16} + G_{12} \end{pmatrix} \end{bmatrix} \quad (12)$$

$$\mathbf{x} = [v_1 \quad v_2 \quad v_3 \quad v_4 \quad v_5]^T \quad (13)$$

$$\mathbf{y} = [i_{21} \quad i_{22} \quad i_{23} \quad i_{24} \quad i_{25} \quad i_{26}]^T \quad (14)$$

$$\mathbf{b}_{orig} = \begin{bmatrix} I_{eq1} - I_{eq4} \\ I_{eq3} - I_{eq6} \\ I_{eq5} - I_{eq2} \\ V_{DC}/R_g - I_{eq1} - I_{eq3} - I_{eq5} \\ -V_{DC}/R_g + I_{eq4} + I_{eq6} + I_{eq2} \end{bmatrix} \quad (15)$$

$$\mathbf{V} = \mathbf{W}^T = \begin{bmatrix} 1 & \cdot & \cdot & -1 & \cdot & \cdot \\ \cdot & \cdot & 1 & \cdot & \cdot & -1 \\ \cdot & -1 & \cdot & \cdot & 1 & \cdot \\ -1 & \cdot & -1 & \cdot & -1 & \cdot \\ \cdot & 1 & \cdot & 1 & \cdot & 1 \end{bmatrix} \quad (16)$$

$$-\mathbf{S}^{-1} = -d \operatorname{ag}(G_{21}^{-1}, G_{22}^{-1}, G_{23}^{-1}, G_{24}^{-1}, G_{25}^{-1}, G_{26}^{-1}) \quad (17)$$

Referring to Fig. 3, the corresponding network matrices for mesh formulation are given in (18)-(23). It should be noticed that proposed switch model reduces the three parallel branch arrangement in Fig. 1c to *one* branch as shown in Fig. 1h. This reduction is an important advantage only observed in mesh formulations, which keeps the order of the network matrix from increasing unnecessarily due to trapped meshes in electronic switches (e.g., one mesh between the IGBT and the diode, and one between the diode and the snubber).

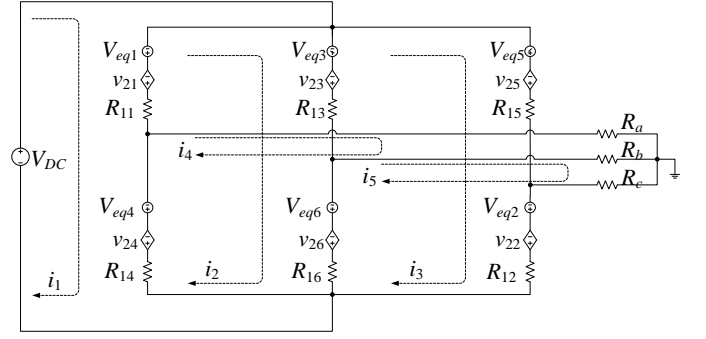


Fig. 3. Voltage-source inverter show for a mesh formulation

$$\mathbf{A}_{const} = \begin{bmatrix} R_{11} + R_{14} & -R_{11} - R_{14} & \cdot & -R_{14} & \cdot \\ -R_{11} - R_{14} & \begin{pmatrix} R_{11} + R_{13} \\ +R_{16} + R_{14} \end{pmatrix} & -R_{11} - R_{13} & R_{16} + R_{14} & -R_{16} \\ \cdot & -R_{11} - R_{13} & \begin{pmatrix} R_{13} + R_{15} \\ +R_{12} + R_{16} \end{pmatrix} & -R_{16} & R_{12} + R_{16} \\ -R_{14} & R_{16} + R_{14} & -R_{16} & \begin{pmatrix} R_a + R_b \\ +R_{16} + R_{14} \end{pmatrix} & -R_b - R_{16} \\ \cdot & -R_{16} & R_{12} + R_{16} & -R_b - R_{16} & \begin{pmatrix} R_b + R_c \\ +R_{12} + R_{16} \end{pmatrix} \end{bmatrix} \quad (18)$$

$$\mathbf{x} = [i_1 \quad i_2 \quad i_3 \quad i_4 \quad i_5]^T \quad (19)$$

$$\mathbf{y} = [v_{21} \quad v_{22} \quad v_{23} \quad v_{24} \quad v_{25} \quad v_{26}]^T \quad (20)$$

$$\mathbf{b}_{orig} = \begin{bmatrix} V_{DC} - V_{eq1} - V_{eq4} \\ V_{eq1} - V_{eq3} - V_{eq6} + V_{eq4} \\ V_{eq3} - V_{eq5} - V_{eq2} + V_{eq6} \\ -V_{eq6} + V_{eq4} \\ -V_{eq2} + V_{eq6} \end{bmatrix} \quad (21)$$

$$\mathbf{V} = \mathbf{W}^T = \begin{bmatrix} -1 & \cdot & \cdot & -1 & \cdot & \cdot \\ 1 & \cdot & -1 & 1 & \cdot & -1 \\ \cdot & -1 & 1 & \cdot & -1 & 1 \\ \cdot & \cdot & \cdot & 1 & \cdot & -1 \\ \cdot & -1 & \cdot & \cdot & \cdot & 1 \end{bmatrix} \quad (22)$$

$$-\mathbf{S}^{-1} = -d \operatorname{ag}(R_{21}^{-1}, R_{22}^{-1}, R_{23}^{-1}, R_{24}^{-1}, R_{25}^{-1}, R_{26}^{-1}) \quad (23)$$

This case presented the matrices required to formulate the simulation of a three-phase inverter in both nodal and mesh variables. The matrices are of deliberate small size so that the equations shown can be traced longhand. What is important to note is that when an IGBT, or its diode commutates, only the i^{th} diagonal element in (17) or (23) needs to be updated. Finally, in both cases, (12) and (18) are constant throughout a simulation.

B. Case 2: Shipboard Power System

The proposed switch model has been tried on the notional AC-radial shipboard power system model presented in [16]. This power system comprises 397 power apparatus, of which 19 are motor drives consisting of 6-diode rectifiers and 6-IGBT inverters. The total number of power electronic

switches arising from these 19 motor drives is 342 (19 inverters = 114 IGBTs with 114 diodes; 19 rectifiers = 114 diodes). However, because IGBTs with diodes reduce to the single branches as shown in Fig. 1d and Fig. 1h, the order of \mathbf{S} reduces to 228. This is a welcome advantage of the proposed switch model.

This type of power system model is computationally complex, and takes a significant amount of time to simulate with commercial simulation tools. Enumerations of the number of meshes, nodes, branches, and switches for the power system of this case study are listed in Table I and II. The ratio between network variables and switches is also shown in Table II, and is a measure of whether speedup results from the proposed method. The larger is this ratio, the larger is the speed gain when compared to the classical switching approach.

Comparing the run-time reductions observed in Table III for the nodal and mesh formulation cases, the run-time reduced from several hours to under one hour. This reduction corresponds to an improvement of 26 and 38, respectively. These gains come from making the network coefficient constant, and only refactoring \mathbf{K} when a switch commutates.

VI. CONCLUSIONS

It is faster to re-factor a matrix of size equal to the number of switches than a matrix of size equal to the number of network variables. Moreover, the larger is the ratio of network variables to number of switches, the larger the speedups will be. Stated differently, by treating switches as sources \mathbf{y} , when switches commutate only \mathbf{S} is updated, and is more effective than updating \mathbf{A}_{orig} .

Experiments show that the proposed switch model is advantageous when $\text{rank}(\mathbf{A}_{large}) > 500$ and when the switch ratio is ≥ 5 . From Table II, the network-variable to switch ratios were ~ 5 and ~ 6 for the nodal and mesh formulation case, respectively. This ratio is also responsible for the speedups observed. If this ratio is low there is no noticeable advantage when using the proposed model vs. the traditional switching model. Large ratios occur frequently in the simulation of power systems such as shipboard power systems and micro grids. In these cases, as has been demonstrated, it is beneficial to use the proposed switch model over the classical switch model.

Use of the proposed switch model permits using readily available math libraries for .NET, or other math programs such as *MATLAB* as the computational engine. This significantly reduces the implementation time of a power system simulator.

TABLE I
MATRIX DIMENSIONS

Matrix	Dimensions
\mathbf{A}_{const}	1448 x 1448
\mathbf{A}_{large}	1676 x 1676
\mathbf{V}	1448 x 228
\mathbf{S}	228 x 228

TABLE II
ELEMENT COUNT

Element	Count
Branches	2580
Meshes	1448
Nodes	1133
Switches	228
Switch Ratio	
Nodes : Switches	4.97
Meshes : Switches	6.35

TABLE III
PERFORMANCE ADVANTAGES

	Nodal	Mesh
Classical	7h 43 min.	8h 17 min.
Proposed	17.8 min.	13 min.
Speed gain	26	38

VII. APPENDIX

This section describes how snubber branches are discretized using the root-matching technique [4],[17],[18]. Consider the series-RC snubber branch shown in Fig. 1b, where, hereafter, R and C represent the snubber resistance and capacitance, respectively. The admittance transfer function for this branch is given in (24) where s is the LaPlace operator.

$$Y(s) = I(s)/V(s) = sC/(1 + sRC) \quad (24)$$

Using the final value theorem with a ramp input, the final value of $Y(s)$ is given by (25). Transferring the pole $\omega_{pole} = -1/RC$ to z -domain results in z -domain transfer function in (26), where Δt is the discretization time step increment. Taking the final value of (26), also with a ramp input, results in (27), where k is a gain that ensures the s - and z -domain final values match. Equation (25) to (27) results in (28).

$$\lim_{s \rightarrow 0} \left(\frac{1}{s^2} Y(s) \right) = C \quad (25)$$

$$Y[z] = I[z]/V[z] = k(z-1) / (z - e^{\Delta t \omega_{pole}}) \quad (26)$$

$$\lim_{z \rightarrow 1} \left(\frac{z \Delta t}{(z-1)^2} Y(z) \frac{z-1}{z} \right) = \Delta t k / (1 - e^{-\Delta t \omega_{pole}}) \quad (27)$$

$$k = C (1 - e^{-\Delta t \omega_{pole}}) / \Delta t \quad (28)$$

For a *nodal* formulation, solving for the *current* in (26) results in (29). Taking the inverse z -transform of (29) results in the discrete time domain form of (30) from which the Norton equivalent in Fig. 4 is obtained.

$$I[z] = kV[z] - z^{-1} (kV[z] - e^{-\Delta t \omega_{pole}} I[z]) \quad (29)$$

$$i^{k+1} = kv^{k+1} - (kv^k - e^{-\Delta t \omega_{pole}} i^k) \quad (30)$$

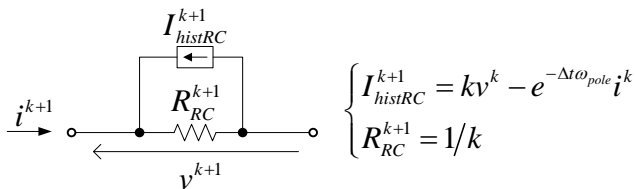


Fig. 4. Norton equivalent for series RC snubber (nodal formulations)

For a *mesh* formulation, the ThéVénin equivalent can be obtained from a Norton-ThéVénin transformation starting from Fig. 4. This transformation results in the voltage expression in (31) and equivalent shown in Fig. 5.

$$v^{k+1} = \frac{1}{k} i^{k+1} + \left(v^k - \frac{e^{-\Delta t \omega_{pole}}}{k} i^k \right) \quad (31)$$

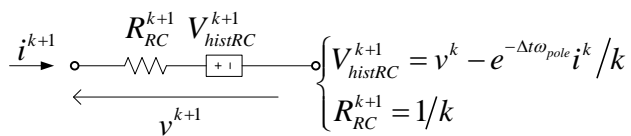


Fig. 5. ThéVénin equivalent for series RC snubber (mesh formulations)

REFERENCES

- [1] H. Macbahi, A. Ba-Razzouk, and A. Chériti, "Decoupled parallel simulation of power electronics systems using Matlab-Simulink," in *Proc. 2000 International Conference on Parallel Computing in Electrical Engineering (PARELEC'00)*, pp. 232-236, 2000, 2000.
- [2] P. Pejovic and D. Maksimovic, "A method for fast time-domain simulation of networks with switches," *IEEE Trans. Power Electronics*, vol. 9, pp. 449-456, 1994.
- [3] Power System Simulator PSCAD (Jan. 2010). <http://pscad.com/>. [Online]. Available:
- [4] N. Watson and J. Arrillaga. (2003). *Power Systems Electromagnetic Transients Simulation*. (1st ed.) London: IEE.
- [5] Math.NET Project (Nov. 2010). <http://www.mathdotnet.com/>. [Online]. Available:
- [6] H. W. Dommel, "Digital computer solution of electromagnetic transients in single- and multiphase networks," *IEEE Trans. Power Apparatus and Systems*, vol. PAS-88, pp. 388-399, Apr. 1969.
- [7] R. Hebner, J. Herbst, and A. Gattozzi, "Large scale simulations of a ship power system with energy storage and multiple directed energy loads," in *Proc. 2010 Grand Challenges in Modeling & Simulation (GCMS 2010)*, pp. 430-435, July 11-14, 2010.
- [8] Q. Huang, J. Wu, J. L. Bastos, and N. N. Schulz, "Distributed Simulation Applied to Shipboard Power Systems," in *Proc. 2007 Electric Ship Technologies Symposium, 2007. ESTS '07. IEEE*, pp. 498-503, 2007.
- [9] P. T. Norton, P. Deverill, P. Casson, M. Wood, G. Dudgeon, and A. Bennet, "The reduction of simulation software execution time for models of integrated electric propulsion systems through partitioning and distribution," in *Proc. 2007 Electric Ship Technologies Symposium (ESTS)*, pp. 53-59, May 22-23, 2007.
- [10] C.-W. Ho, "The modified nodal approach to network analysis," in *Proc. 1974 IEEE International Symposium on Circuits and Systems*, pp. 505-509, 1974.
- [11] I. S. Duff, A. M. Erisman, and J. K. Reid. (1986). *Direct Methods for Sparse Matrices*. Oxford: Oxford University Press.
- [12] H. V. Henderson and S. R. Searle, "On Deriving the Inverse of a Sum of Matrices," *SIAM Review*, vol. 23, pp. 53-60, 1981.
- [13] A. Klos, "What Is Diakoptics?," *International Journal of Electrical Power & Energy Systems*, vol. 4, pp. 192-195, 1982.
- [14] G. Kron. (1963). *Diakoptics: The Piecewise Solution of Large-Scale Systems*. London: MacDonald & Co.

- [15] C.-W. Ho, A. E. Ruehli, and P. A. Brennan, "The modified nodal approach to network analysis," *IEEE Trans. Circuits and Systems*, vol. CAS-22, pp. 504-509, Jun. 1975.
- [16] K. L. Butler-Purry and N. D. R. Sarma, "Geographical information systems for automation of shipboard power systems," *Naval Engineers Journal*, vol. 118, pp. 63-75, 2006.
- [17] N. R. Watson and G. D. Irwin, "Electromagnetic transient simulation of power systems using root-matching techniques," *IEE Proceedings: Generation, Transmission and Distribution*, vol. 145, pp. 481-486, 1998.
- [18] J. M. Smith. (1987). *Mathematical Modeling and Digital Simulation for Engineers and Scientists*. (2nd ed.) Washington, DC: John Wiley.

BIOGRAPHIES

Fabian Marcel Uriarte obtained his BS and MS in electrical engineering from Virginia Tech in 2002 and 2003 respectively. He obtained his PhD in electrical engineering from Texas A&M University in 2010 in the area of multicore simulation of power systems. His research interests are in parallel computing, real-time simulation, and modeling and simulation of power systems and power electronic converters.

Salman Mashayekh (S'09) received his B.S. and M.S. in Electrical Power Systems from University of Tehran, Iran, in 2006 and 2008, respectively. He joined the Power System Automation Lab in Texas A&M University as a PhD student in 2008. His research interests are in power management for isolated power systems, cyber security of smart grids, and modeling and simulation of power systems.