Multicore Simulation of an Ungrounded Power System

Fabian Marcel Uriarte

***Abstract*—An approach to partition power system simulations formulated in mesh currents as variables is presented. Time domain simulations of large-scale power systems are notoriously slow due to the system order and amount of time-varying components. A common approach to reduce simulation run-time is to partition power systems using Bergeron's travelling-wave model and to simulate the subsystems on a distributed computer. In this paper, power systems are partitioned using a new mesh tearing approach, and the subsystems distributed across the cores of a quad-core computer. A notional shipboard power system model was partitioned using the proposed partitioning approach, which is used to show how the simulation run-time reduces as a function of the number of partitions and number of power converters.**

***Index Terms*—core, delta, mesh, multicore, parallel, partition, power, ship, system, tearing, ungrounded.**

I. Introduction

Power system researchers often face the problem of slow time domain simulations. The speed at which these simulations execute is attributed to–at least–the following reasons: system order, integration step size, and the multitude of time-varying components. What is not always recognized, however, is that unpartitioned simulation approaches are computationally inefficient for large systems. This limitation hinders the ability for researchers to advance technology, run many case studies per day, and test new methodologies at a desirable rate. The aforementioned reasons have motivated to seek for alternate (partitioned) simulation methods to reduce simulation run-time.

Two common partitioning approaches to speed up power system simulations are Bergeron's travelling-wave line model [1-4], and the insertion of time step (Δt) delays (a.k.a. *stublines* [1]). Bergeron's model block-diagonalizes the coefficient matrix of a power system model via the travel delay of transmission lines, where each subsystem can be solved independently. The limitation of Bergeron's line model, however, is that the simulation time step Δt must be an integer fraction of the travelling-wave travel delay [2]. Consequently, when systems only have short transmission lines (e.g., distribution systems, shipboard power systems, microgrids, etc.), the travel delays are $O(10^{-9})$ or $O(10^{-12})$ seconds, and makes Bergeron's model impractical.

Partitioning through Δt delays is effective in some simulation scenarios. A partitioning scheme for short distribution lines was presented in [3], where the trapezoidal rule of integration was converted into an explicit integration algorithm. With explicit integration, inductors and capacitors can be modeled as historical current and voltage sources, respectively. (From circuit theory, voltage and current sources can be torn apart without violating physical properties [4].) In [1],[5], the use of stublines showed accurate simulation results despite the introduction of a Δt delay. Although suitable for some simulation conditions, forcing Δt delays in time domain simulations (where none physically exist) may accumulate phase drifts and drive longtime simulations unstable [6]. Furthermore, the likelihood of instability increases with the presence of time-step delayed and/or time-varying power apparatus models such as machines and power converters.

This paper presents a power system partitioning approach to speed up time domain simulations, and is suitable for shipboard power systems (SPSs). Current technology naval warships consist of complex electrical (~7.5MW) and mechanical (~80MW) systems that demand vast computational resources for detailed computer-aided analysis. A consequence of these saliencies is that some traditional analysis methods used in terrestrial systems fail (e.g., phasor analysis does not apply to non-linear, unbalanced, AC/DC power systems). Some characteristics that make vessel power systems different from terrestrial power systems are that SPSs:

- are predominantly, and intentionally ungrounded (e.g., delta power systems),
- have many non-linear and time-varying loads,
- have finite-inertia generation,
- are tightly coupled,
- have many protective devices (and many types)
- have pulsed loads,
- many motor drives,
- have unbalanced single-phase loads,
- have a ring topology,
- have fast generation controllers,
- have three-phase and single-phase sides,
- have several AC and DC voltage levels,
- are divided into DC distribution zones, and
- operate at frequencies of 60 Hz, 400 Hz, etc.

Next generation naval warships will put even more stress on the electrical system as the propulsion will rely on induction motors with consumptions in the megawatt range. As it is today, power converters force constant topological changes and inject significant harmonics, which already presents an added burden to conventional computer simulation methods. Henceforth, specialized simulation methods capable of simulating SPSs with the above characteristics, and that can execute in *reasonable* time are sought.

Some of the demands of simulating SPSs have been met by research centers such as the Center for Advanced Power Systems at Florida State University [5], Drexel University [7],[8], Texas A&M University [9], and Mississippi State University [10], etc. [11],[12]. These institutions have acquired real-time computing resources that, although reduce the simulation run-time of their SPS models, are not accessible by everyone in the ship research community.

In lieu of the simulation need, and expensive resources to make this possible, an alternative simulation approach has emerged recently: the use of low-cost multicore computers to overcome demanding simulating scenarios. Despite the improvements in computer speed today, it is always necessary to examine methods that can divide the work into smaller tasks such as presented here. The work herein meets said computationally intensive simulation scenarios by presenting a specialized simulation method. The advantages of this method are exemplified by simulating a large-order (359 power apparatus) SPS, where it is shown that simulation run-time reduced up to three orders of magnitude.

The remainder of this paper is organized as follows. Section II describes the power system model used in this paper. Section III presents the simulation methodology. Section IV presents a case study in three variations to assess the speed gain of the proposed partitioning approach.

II. The Power System Model

This section presents an overview of the power system shown in Fig. 13, which is based on the delta-ungrounded AC-radial SPS in [13],[14]. The power apparatus and nodes that comprise the power system model are given in Table I and Table II, respectively. As seen from the power apparatus and node counts, this system is considered a *large* model for time domain simulations when compared to typical simulations. A detrimental consequence of a model this size is that its simulation takes anywhere from hours to days depending on what power apparatus are included in a simulation run.

TABLE I
LIST OF POWER APPARATUS INCLUDED IN THE POWER SYSTEM MODEL

Acronym	Component Description	Number of
GEN	Synchronous generator	3
MOT	Induction motor	19
RCT	3φ rectifier	19
INV	3φ inverter	19
Cbl	1φ cable	33
CBL	3φ cable	108
Lod	1φ static load	33
LOD	3φ static load	13
XFM	Transformer	11
BRK	Circuit breaker	83
ABT	Automatic bus transfer	15
MBT	Manual bus transfer	13
LVP	Low-voltage protective device	17
LVR	LVPs w/recloser	2
FLT	3φ fault	9
	Total	397

TABLE II
BUS TYPES AND COUNT

Type	Number of
Three-phase	266
Single-phase	85
Total	351

The synchronous generators (2.5 MW, 450 V, 60 Hz, eight poles) were modeled as delta-connected controlled voltage sources behind static impedances. The voltage controllers were modeled as PID controllers that regulated the line-to-line voltage to 450 VAC. The induction motors were modeled as a motor drive in front of a set of windings. The motor drives were modeled as uncontrolled three-phase rectifiers (6 diodes) and PWM-controlled voltage-source inverters (6 IGBTs). The motor windings were modeled using approximate per-phase models [15] connected in delta. The 19 motor drives introduce 12 x 19 =

228 power electronic switches: the most burdensome aspect of the simulation. All cables (single-phase and three-phase) were modeled as nominal-π line models, with shunt capacitance and conductance between phases, but not to the ground. A ground plane was not included to model the fact that SPSs are intentionally ungrounded, and to reduce the mesh current count. The loads (three-phase and single-phase) were modeled as static impedance loads, where the three-phase loads were connected in delta. The transformers were modeled as Δ-Δ 450:120V step-down transformers using single-phase T-models. Circuit breakers were modeled as three on/off switches with instantaneous over-current relays. Bus transfers were modeled as three-pole double-throw (six) switches also as on/off switches, but with under-voltage relays. Low-voltage protective devices were modeled as circuit breakers with under-voltage relays. In addition to the aforementioned power apparatus, nine three-phase 50 mΩ faults were included. The fault locations were obtained from the geographical information system (GIS) presented in [14], and were sequentially applied at times t = 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, and 0.9 s, respectively. (More details on this SPS model can be found in [13].)

III. Simulation Methodology

This section presents the simulation methodology as three stages: discretization, partitioning, and simulation. The discretization stage explains how the power apparatus models were discretized for a computer implementation formulated in mesh currents as variables. The partitioning stage explains the partitioning approach used to simulate the example power system as subsystems. The simulation stage presents implementation details about the multithreaded program that was developed

A. Discretization

Each of the power apparatus listed in Table I was discretized by replacing its inductors and capacitors with the discretized (equivalent) branches shown in the lower part of Fig. 1. These discretized branches have *series* historical voltage sources to accommodate for a network formulation in *mesh currents* as variables (i.e., as opposed to *parallel* historical current sources for formulations in *node voltages* as variables).

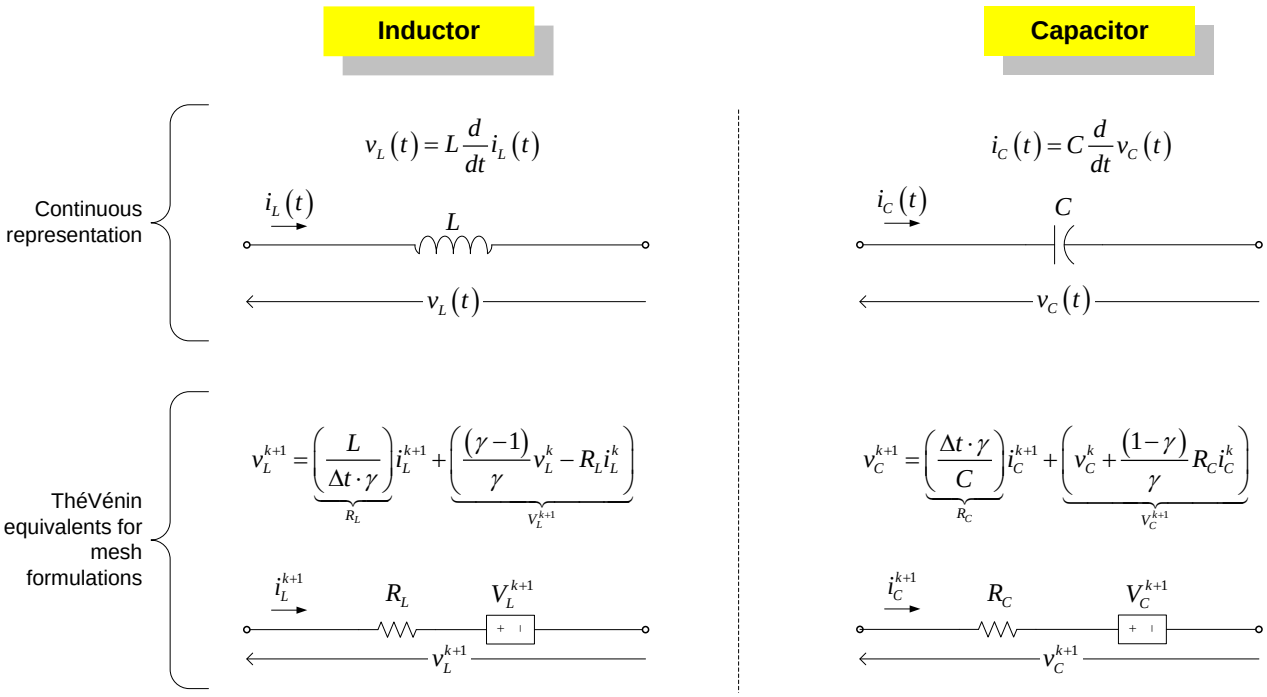

Fig. 1. Discretization of inductors and capacitors for formulations in mesh currents as variables.

In Fig. 1, Δt represents the time step increment, $k+1$ represents the present simulation time step, k represents the previous simulation time step, and γ a tunable parameter [16] used to change integration algorithms during run-time (e.g., $\gamma = \frac{1}{2}$ for the trapezoidal rule of integration; $\gamma = 1$ for backward Euler integration).

Discretization is exemplified by comparing the continuous representation of a three-phase cable model to its discretized counterpart as shown in Fig. 11. All continuous inductors were replaced by discrete inductors. To avoid trapped meshes inside the shunt *RC* branches, the *RC* branches were lumped into one series branch using Norton and ThéVénin transformations [17],[18].

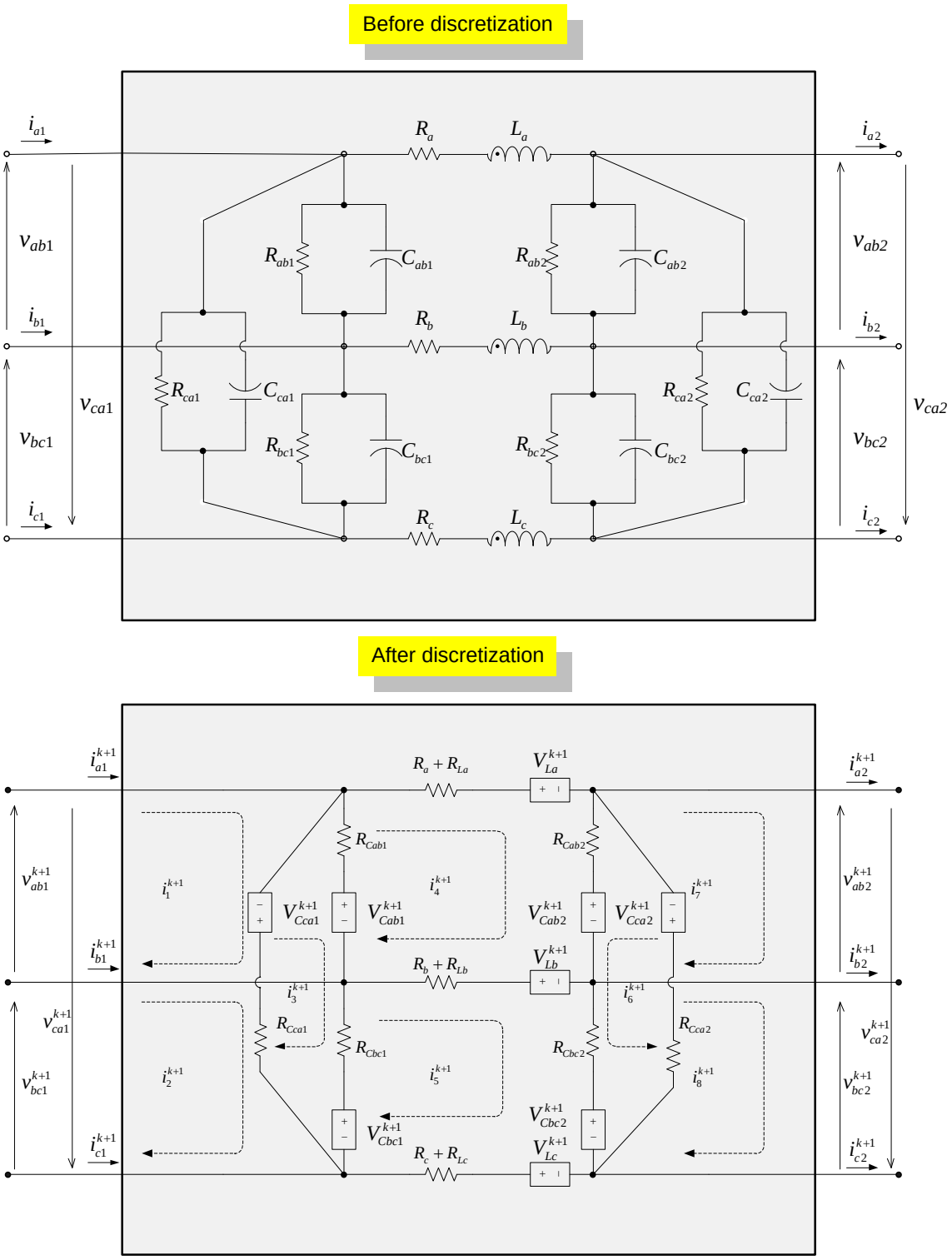

Fig. 2. Discretization of a three-phase cable. All inductors and capacitors are replaced by the discretized forms presented in Fig. 1. The line-to-line RC branches were combined into a single branch using ThéVénin-Norton reductions.

B. Partitioning

To form SPS subsystems from the power system model shown in Fig. 13, power apparatus were interconnected everywhere *except* across the boundaries of two or more subsystems. This interconnection approach is a piecewise one, where only power

apparatus belonging to the *same* subsystem are interconnected. After all power apparatus are interconnected, the resulting system is a set of isolated subsystems.

Determining which power apparatus belong to each subsystem requires search heuristics. This aspect of the power system partitioning was handled with graph theory [19]. A graph of the power system in Fig. 13 was created, where each vertex corresponded to one power apparatus, and where the vertex weights were the number of mesh currents in the corresponding power apparatus. Each graph edge corresponded to a single- or three-phase node. The partitioning and balancing of this weighted graph was carried out with graph partitioning software hMETIS [20]. By giving hMETIS the system graph as a text file, hMETIS returned p sets of vertices, where each vertex set consisted of the power apparatus in each subsystem. After the knowing the power apparatus that belong to each subsystem, the mesh resistance[1] matrix $\mathbf{R}_{mesh}$ for each subsystem was created[2].

By having a partitioned power system, a new set of boundary equations must be solved as well. Consider the abstraction of a three-phase network in Fig. 3, where the mesh currents traverse across both sides of the network. The situation in Fig. 3 depicts a candidate disconnection point from which the network can be torn. At a disconnection point as such, two voltage sources of unknown value are inserted as shown in Fig. 4. The introduction of these voltage sources *tear* the three mesh currents i_{ab}, i_{bc}, and i_{ca} into six mesh currents: i_{ab1}, i_{ab2}, i_{bc1}, i_{bc2}, i_{ca1}, and i_{ca2}. From circuit theory, the voltage sources can be split as shown in Fig. 5 without violating physical properties.

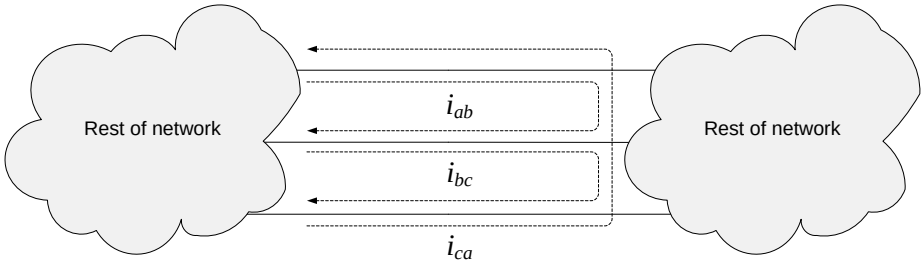

Fig. 3. A disconnection point where mesh currents couple different areas of a network.

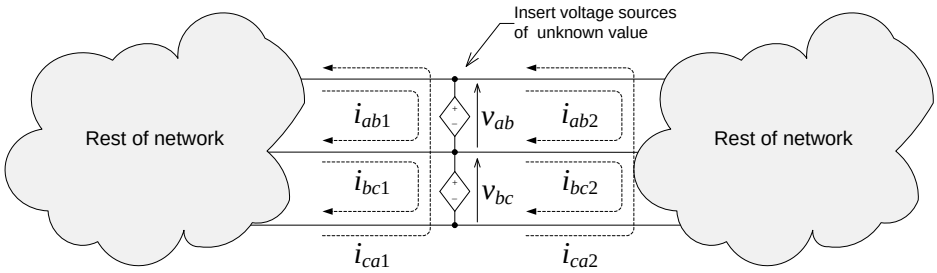

Fig. 4. Voltage sources of unknown value are inserted at a disconnection point to tear the mesh currents.

[1] After discretization, a power system becomes a purely resistive network
[2] The partitioning approach in this work assumes a formulation in mesh current as variables

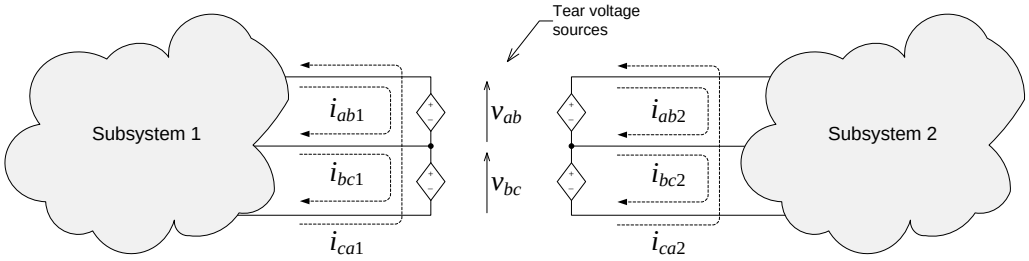

Fig. 5. Longitudinal tearing of unknown voltage sources partitions the network and forms subsystems.

A partitioned network, as shown in Fig. 5, can be formulated in doubly-bordered block-diagonal form as given in (1), where the unknown voltage source values are arranged in $\mathbf{u}^{k+1}$ (k+1 notation is omitted for brevity: e.g., **u**). The advantage of a formulation like (1) is manifested through its solution, which is given as (3) and (4) [21],[22].

Equation (1) comprises two rows of matrix equations. The top row relates each subsystem's mesh equations to the boundary variable vector **u**. The bottom row states that the net (mesh) current through the voltage sources inserted in Fig. 4 is zero, which is true of the original network in Fig. 3. Referring to Fig. 3, the equations at a disconnection point for a *connected* network are given in (5). Equation (5) states that the net mesh current between conductors *a* and *b*, and *b* and *c* is zero. Equation (5) was organized in matrix form as the second equation row in (1), where the $\mathbf{D}_i$ connection matrices have +1, 0, and -1 as coefficients. An structure plot of the coefficient matrix in (1) (for p=20) is shown as an example in Fig. 11, which is solved using (3) and (4).

$$\left[\begin{array}{cccc:c} \mathbf{R}_{mesh1}^{k+1} & & & & \mathbf{D}_1 \\ & \mathbf{R}_{mesh2}^{k+1} & & & \mathbf{D}_2 \\ & & \ddots & & \vdots \\ & & & \mathbf{R}_{mesh_p}^{k+1} & \mathbf{D}_p \\ \hdashline \mathbf{D}_1^{\mathrm{T}} & \mathbf{D}_2^{\mathrm{T}} & \cdots & \mathbf{D}_p^{\mathrm{T}} & \mathbf{0} \end{array}\right]\left[\begin{array}{c} \mathbf{i}_{mesh1}^{k+1} \\ \mathbf{i}_{mesh2}^{k+1} \\ \vdots \\ \mathbf{i}_{mesh_p}^{k+1} \\ \hdashline \mathbf{u}^{k+1} \end{array}\right]=\left[\begin{array}{c} \mathbf{e}_{mesh1}^{k+1} \\ \mathbf{e}_{mesh2}^{k+1} \\ \vdots \\ \mathbf{e}_{mesh_p}^{k+1} \\ \hdashline \mathbf{0} \end{array}\right] \tag{1}$$

$$\mathbf{D}_p(i,j) \rightarrow \begin{cases} =1, & \text{if the } i^{\text{th}} \text{ mesh current in subsystem } p \text{ enters the } \textit{positive} \\ & \text{terminal of unknown boundary voltage source } j \\ =-1, & \text{if the } i^{\text{th}} \text{ mesh current in subsystem } p \text{ enters the } \textit{negative} \\ & \text{terminal of unknown boundary voltage source } j \\ =0, & \text{otherwise} \end{cases} \tag{2}$$

$\mathbf{e}_{mesh_p}^{k+1}$ = EMF vector of subsystem p

$\mathbf{i}_{mesh_p}^{k+1}$ = mesh current vector of subsystem p

$\mathbf{R}_{mesh_p}^{k+1}$ = mesh resistance matrix of subsystem p

$\mathbf{D}_p$ = tensor linking the mesh currents of subsystem p to its boundary voltages

$\mathbf{u}^{k+1}$ = vector of system wide unknown boundary voltage sources

$\mathbf{0}$ = zero vector or matrix

p = number of subsystems or partitions

$$\begin{bmatrix} \mathbf{i}_{mesh1}^{k+1} \\ \mathbf{i}_{mesh2}^{k+1} \\ \vdots \\ \mathbf{i}_{mesh_p}^{k+1} \end{bmatrix} = \begin{bmatrix} \left(\mathbf{R}_{mesh1}^{k+1}\right)^{-1} \mathbf{e}_{mesh1}^{k+1} \\ \left(\mathbf{R}_{mesh2}^{k+1}\right)^{-1} \mathbf{e}_{mesh2}^{k+1} \\ \vdots \\ \left(\mathbf{R}_{mesh_p}^{k+1}\right)^{-1} \mathbf{e}_{mesh_p}^{k+1} \end{bmatrix} - \begin{bmatrix} \left(\mathbf{R}_{mesh1}^{k+1}\right)^{-1} \mathbf{D}_1 \\ \left(\mathbf{R}_{mesh2}^{k+1}\right)^{-1} \mathbf{D}_2 \\ \vdots \\ \left(\mathbf{R}_{mesh_p}^{k+1}\right)^{-1} \mathbf{D}_p \end{bmatrix} \mathbf{u}^{k+1} \tag{3}$$

$$\mathbf{u}^{k+1} = \underbrace{\left(\sum_{i=1}^{p} \left(\mathbf{D}_i^{\mathrm{T}} \left(\mathbf{R}_{mesh_i}^{k+1} \right)^{-1} \mathbf{D}_i \right) \right)^{-1}}_{\mathbf{\Lambda}} \left(\sum_{i=1}^{p} \left(\mathbf{D}_i^{\mathrm{T}} \underbrace{\left(\mathbf{R}_{mesh_i}^{k+1} \right)^{-1} \mathbf{e}_{mesh_i}^{k+1}}_{\text{from Substep } a} \right) \right) \tag{4}$$

$$\text{Boundary equations} \rightarrow \begin{cases} i_{ab1} - i_{ab2} - i_{ca1} + i_{ca2} = 0 \\ i_{bc1} - i_{bc2} - i_{ca1} + i_{ca2} = 0 \end{cases} \tag{5}$$

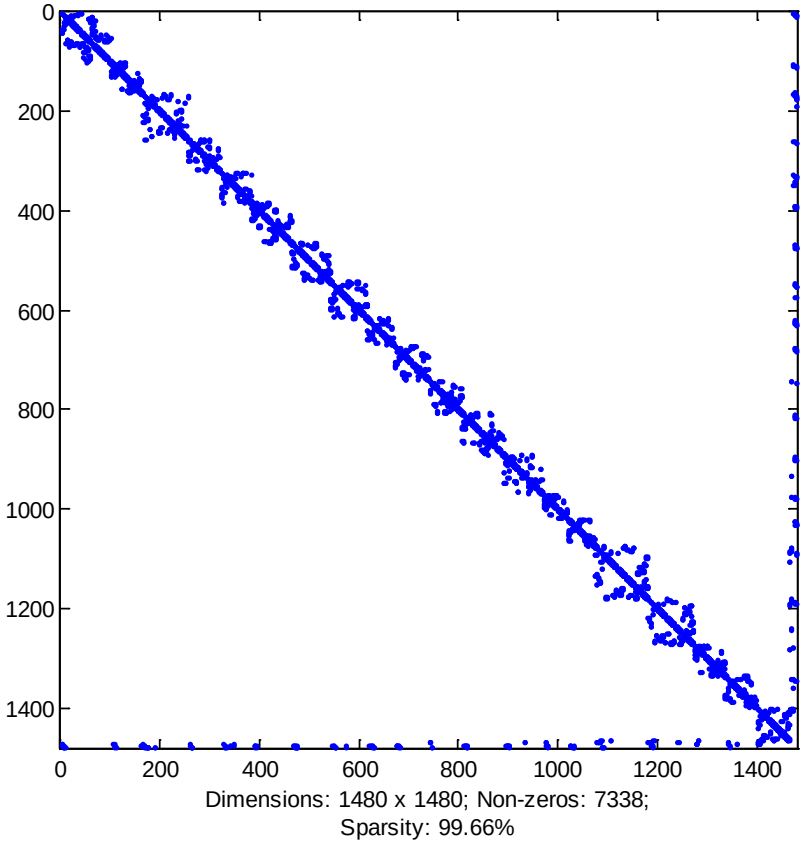

Fig. 6. Structure plot of the coefficient matrix in (1) for $p = 20$ partitions.

The multicore computer used in this work to solve (3) and (4) is described in Table III.

TABLE III
MULTICORE COMPUTER USED FOR CASE STUDY

Brand & Model	HP HDX 18 laptop
Memory (RAM)	4GB (DDR3)
Operating System	Windows Vista Home Edition (64-bit) with Service Pack 2
Processor	Intel Q9000 quad-core 2.0GHz

C. Simulation

The simulation stage consists of developing a multithreaded strategy to implement. A *thread* is an independent path of code execution, or informally: an agent that can execute code concurrently *and* independently of other agents. A swim lane diagram [23] is presented in Fig. 7 to illustrate the work performed by each thread at each time step of a multithreaded simulation program developed in *C#* 3.0.

Each simulation time step *k* is divided into substeps *a* through *h*. In all substeps, except for *Substep d* and *Substep f*, the threads work concurrently and do not require information from each other. Detailed descriptions of each substep follows.

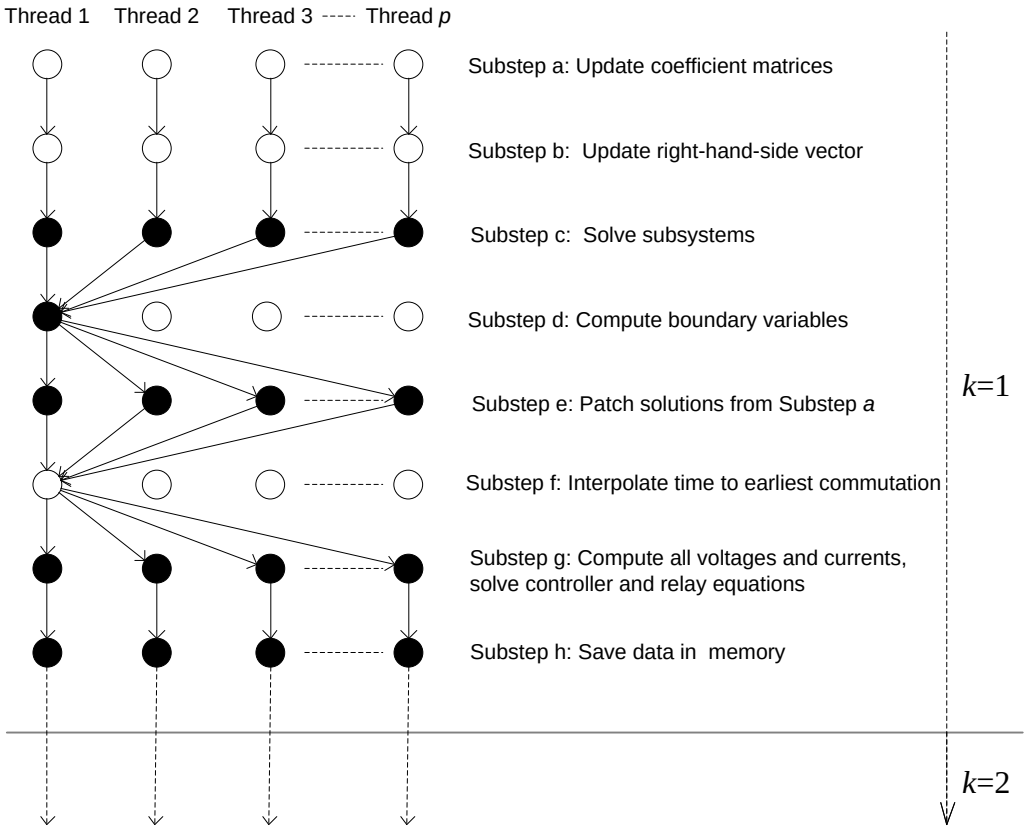

Fig. 7. Swim lane diagram for threads 1-*p*. It is possible to have more than one thread per core.

Substep a: Each thread updates the coefficient matrix $\mathbf{R}_{mesh}$ corresponding to its subsystem. This is necessary if a power converters switch or a protective device changed state. Further, when a switch changes state, matrix $\mathbf{\Lambda}$ in (4) is re-factored by thread 1 in *Substep d*.

Substep b: In each subsystem, all historical and independent sources are updated. Updating all sources requires updating $\mathbf{e}_{mesh}$ in each subsystem.

Substep c: Each thread solves the first term on the right-hand side of (3) using a Cholesky factorization.

Substep d: Using all solutions from *Substep c*, thread 1 computes the boundary condition vector **u** in (4), while threads 2-*p* wait. This is the major bottle-neck of the partitioning algorithm presented, and is common to other diakoptics-based partitioning strategies as well [13]. In fact, the time spent on this substep limits the speed-up.

Substep e: After thread 1 computes **u**, each thread patches the result from *Substep a*, which corresponds to the second term on the right-hand side of (3). The physical interpretation of this substep is that after each thread solves its *own* electrical network, the boundary voltages in **u** impress voltages at the disconnection points, which results in an overall solution by superposition.

Substep f: When simulating power electronic devices, at each k it must determined whether a switch has commutated (e.g., whether a diode current went negative, or its voltage surpassed the turn-on voltage). The exact time at which the first switch commutates is called the zero-crossing instant t_z. In this substep, all switches are polled [24] to find t_z. If a switch commutated, the solution of *Substep e* is interpolated back to t_z [25] before continuing to *Substep g*. The corresponding coefficient matrices are updated in *Substep a* at the next k. If no switch commutated during this substep, the solution from *Substep e* is not altered

Substep g: Using the solution from *Substep e* (or *Substep f* if a switch commutated), the voltages and currents for each subsystem are computed (instantaneous, RMS, power flows, etc.).

Substep h: The measurements made in *Substep g* for all protective devices, as well as additional measurements specified by the user, are saved to memory arrays.

This section presented the simulation methodology in three stages: discretization, partitioning, and simulation. The discretization stage exemplified how each power apparatus was discretized at the *branch* level. The partitioning stage explained how to find and tear the disconnection points of a network, and derived the boundary equations. The simulation stage presented details at the implementation level. The next section presents speed-up results from partitioning and simulating the SPS model shown in Fig. 13.

IV. Case Study

Three variations of a case study are presented to assess the speed gain of the proposed partitioning approach. In all variations nine three-phase line-to-line faults were applied in intervals of 0.1 secs. starting at $t = 0.1$. The three variations of this case study are:

Variation 1: The SPS was simulated *without* power converters. All induction motors were connected directly to the 450 VAC mains (no switches)

Variation 2: The SPS was simulated with all rectifiers loaded with static DC loads instead of inverters and motors (19 rectifiers x 6 diodes = 114 switches)

Variation 3: The SPS was simulated with all power electronic converters included (19 rectifiers x 6 diodes + 19 inverters x 6 IGBTs = 228 switches).

The purpose of having three variations was to observe the speed gain as both the number of partitions and number of switches varied. The simulation speed gain K_{speed} was using (6), where $t_{\text{unpartitioned}}$ is unpartitioned simulation run-time in seconds, and $t_{\text{partitioned}}$ is any partitioned simulation run-time in the same units. The number of partitions was varied as follows: $p = \{1, 2, 5, 10, 20, 30, 40, 50, 60\}$.

$$K_{speed} = \frac{t_{\text{unpartitioned}}}{t_{\text{partitioned}}} \tag{6}$$

The run-time and speed gain results for all variations are shown in Table IV, Table V, and Table VI. For variation 1 in Table IV, the run-time reduced from 18 minutes to approximately 1 minute (p=30, K_{speed}=11). For variation 2 in Table V, the run-time reduced from 3 hours and 35 minutes to approximately 5 minutes (p=30, K_{speed}=40). For variation3 in Table VI, the run-time reduced from approximately nine days to approximately 7 minutes (p=30, K_{speed}=1,800). In all cases, p=30 results in the least run-time (maximum speed gain). The run-times (and gains) between p=30 and p=40 were not well defined (e.g., when p=36, the simulation at times took less time than when p=39; at times it took longer). The speed gains for p=1, 2, 5, 10, 20 , 30, 40, 50, and 60, however, were consistent in all variations.

TABLE IV
RUN-TIME AND SPEED GAIN FOR VARIATION 1

Number of Partitions	Run-time			Speed Gain*
	in Hours	in Minutes	in Seconds	
1	-	19	1,130	1
2	-	6	360	3
5	-	3	190	6
10	-	2	140	8
20	-	2	120	9
30	-	2	100	11
40	-	2	140	8
50	-	3	180	6
60	-	4	210	5

*with respect to 1,130 secs.

TABLE V
RUN-TIME AND SPEED GAIN FOR VARIATION 2

Number of Partitions	Run-time			Speed Gain*
	in Hours	in Minutes	in Seconds	
1	4	216	12,940	1
2	1	63	3,760	3
5	0	19	1,160	11
10	0	11	650	20
20	0	7	400	32
30	0	5	320	40
40	0	6	360	36
50	0	10	570	23
60	0	13	760	17

*with respect to 12,940 secs.

TABLE VI
RUN-TIME AND SPEED GAIN FOR VARIATION 3

Number of Partitions	Run-time			Speed Gain*
	in Hours	in Minutes	in Seconds	
1	210	12,600	756,000	1
2	18	1,050	63,000	12
5	2	97	5,820	130
10	1	50	3,028	250
20	0	10	611	1,237
30	0	7	420	1,800
40	0	7	435	1,738
50	0	8	496	1,524
60	0	18	1,056	716

*with respect to 756,000 secs.

The speed gains listed in Table IV, Table V, and Table VI are plotted as bar charts in Fig. 8, Fig. 9, and Fig. 10, respectively. From Fig. 8, the rate of change of K_{speed} appears quassi-linear. From Fig. 9 and Fig. 10, the rate of change of K_{speed} has a non-linear trend. This can be observed by the abrupt speed gain change between p=1 and p=10. In Fig. 10, beyond p=10, the speed gain reached $O(10^3)$, which is an impressive improvement over waiting nine days for the unpartitioned simulation.

It was observed that the number of switches significantly influenced the speed gain. As the number of switches increased from none (variation 1) to 114 and 228, for variations 2 and 3, respectively, the numerator in (6) increased abruptly. The maximum value of K_{speed} in Fig. 8 is not as high as in Fig. 9 or Fig. 10 precisely due to this reason. Without switches, the mesh resistance matrices of all subsystems was constant (except when applying the faults), which benefits unpartitioned simulations. The more switches were included in the power system model, the higher was the speed gain. This concluding remark is bounded by the switching algorithm routine used by the program that was developed, where the entire coefficient matrix of the unpartitioned simulation (p=1) was re-factored every time a switch commutated.

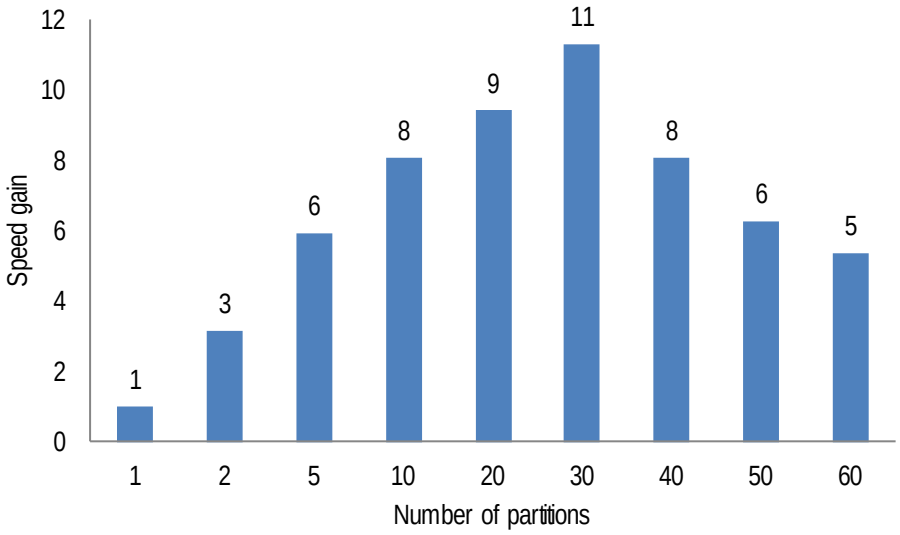

Fig. 8. Speed gain for Variation 1: motors connected directly to the AC mains (no power converters)

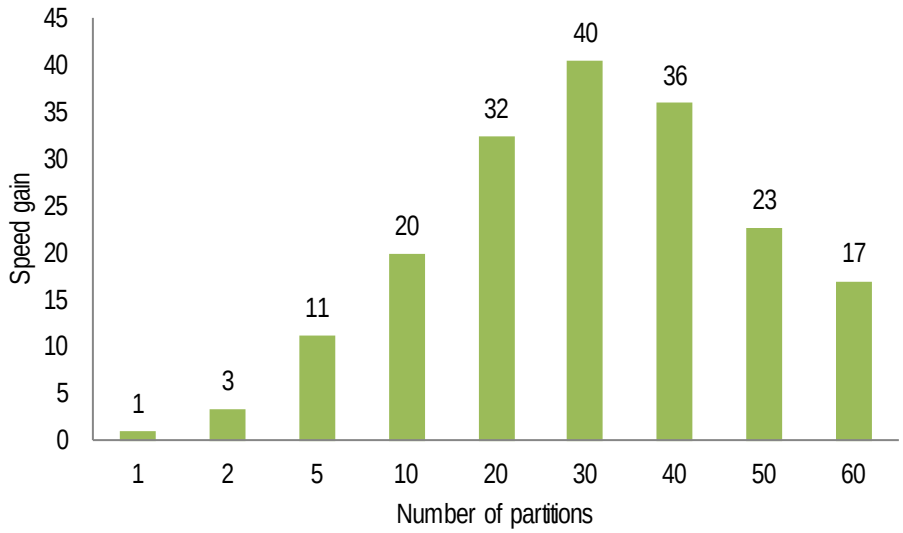

Fig. 9. Speed gain for Variation 2: motor drive rectifiers were loaded with static DC loads instead of inverters and motors (114 switches)

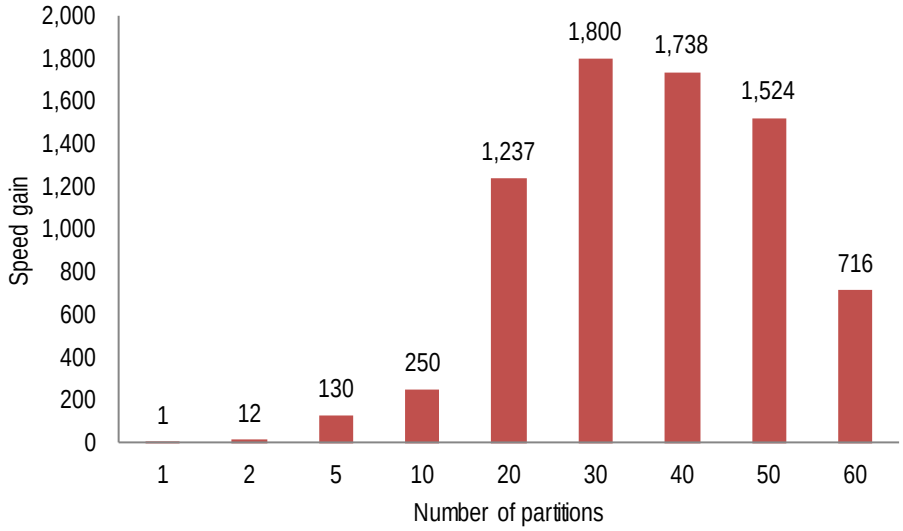

Fig. 10. Speed gain for Variation 3: all motor drive rectifiers and inverters were included (228 switches)

The number of boundary variables (in **u**) increased by two for every disconnection point as illustrated in Fig. 4 (e.g., 62 boundary variables indicates 31 disconnection points). A chart of the average subsystem order (rank) and number of boundary variables is shown in Fig. 11. From Fig. 8, Fig. 9, and Fig. 10, the maximum value of K_{speed} was obtained when p=30, which is when the rank of **u** was close to one-third of the average subsystem rank (i.e., 18/49 = 0.367). This empirical result has been observed for several case studies of the power system model in Fig. 13.

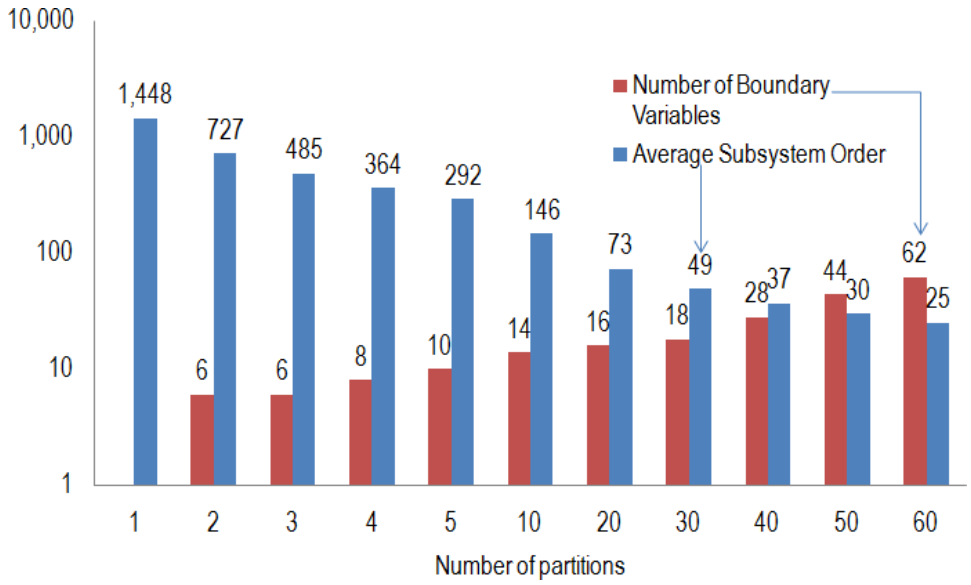

Fig. 11. Number of boundary variables and average subsystem order.

A screen shot of *Microsoft Windows'* task manager was taken moments after starting the simulation for *Variation 3* when p=20, and is shown in Fig. 12. The CPU and memory usage of the computer described in Table III is shown in Fig. 12. All four cores are almost equally used, but not fully. It is not possible to maximize core utilization without giving up computer responsiveness (i.e., keyboard and mouse input). In most cases, a computer running SPS simulations is simultaneously used for other purposes as well.

From the bottom of Fig. 12, a CPU utilization of 87% was observed. This was expected as the graph partitions are not perfectly balanced, which causes cores to idle. Also, 90 processes all together were simultaneously running. This number represents the total number of background applications running concurrently with the simulation, which increases run-time. The speed gains observed in *Windows* vary according to how many (and how often) other processes require processor time; this makes *Windows* a non-deterministic simulation environment. However, the low-cost and ubiquity of multicore computers outweigh the non-determinism in *Windows*. This, in turn, makes *Windows* an attractive operating system for (offline) time domain simulations of power systems.

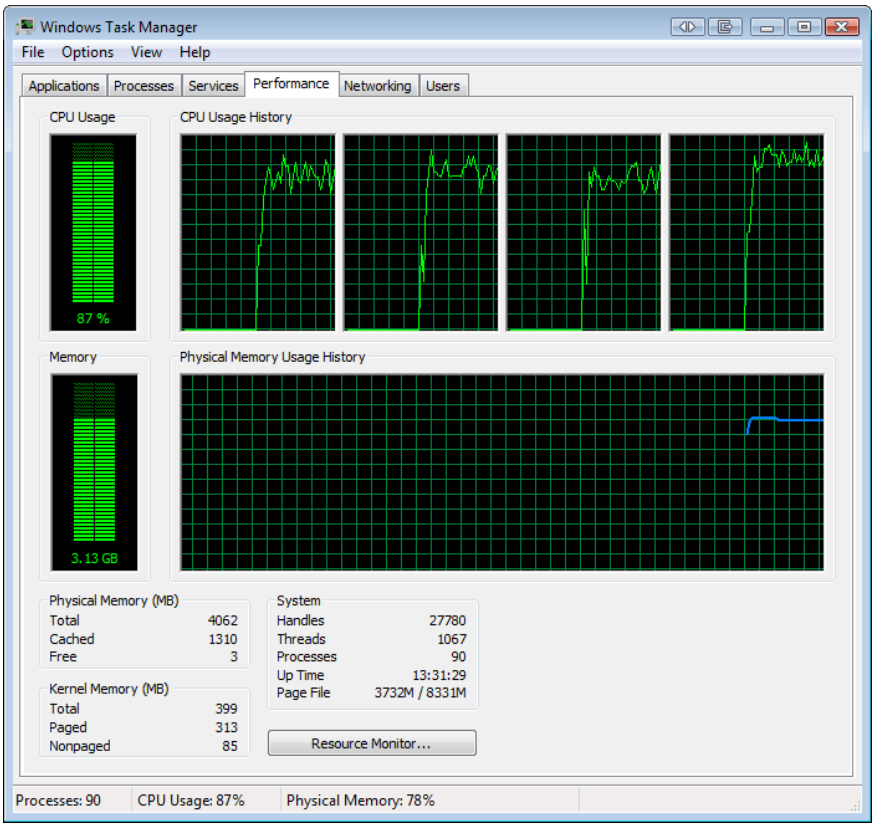

Fig. 12. Screenshot of the *Windows Task Manager* moments after starting the simulation for variation 3 (p=20)

V. Summary

Section I introduced the problem of simulating large-scale SPSs. To reduce simulation run-time, reference was made to two common partitioning techniques: Bergeron's travelling-wave model, and the use Δt delays. The advantages and disadvantages of these two techniques were discussed.

Section II introduced the SPS model used in this work. Two tables were presented to enumerate the power apparatus and node count that comprise the SPS model. Details were given on how the power apparatus were modeled, including references where appropriate. The one-line diagram of the SPS model was introduced Fig. 13.

Section III presented the simulation methodology in three stages: discretization, partitioning, and simulation. The discretization stage explained how power apparatus were discretized for computer simulation, which is also the process of converting a differential network into a purely resistive one. The partitioning stage explained how graph theory software hMeTiS was used to identify the power apparatus that belong in each subsystem. The simulation stage discussed implementation details on the work performed by each thread.

Section IV described a case study, its variations, and presented speed gain results. The results indicated that the speed gain is a function of both the number of partitions and the number of switches included in the model.

VI. Conclusions

The speed gain is a function of–at least–two factors: the number of partitions (p) and the number of power electronic switches. As the number of switches in the system increases, so does the number of solutions per time step resulting from frequent interpolations (i.e., *Substep f*). This requires additional synchronizations at each time step and slows down the simulation. (So does *Substep d*.) The number of switches in a SPS model is determined by the number of power converters that a user considers. To reduce run-time, the number of power converts can be reduced at the expense of realm. The gains obtained are bounded by the program's switching routine. In this paper, the entire mesh resistance matrix of the unpartitioned system and, if partitioned, of each subsystem was re-factored every time a switch commutated.

It is unbearable to simulate large-scale SPSs with so many switches unless a partitioning strategy is implemented. Time-varying systems of small order (e.g., less than rank 100) may be simulated in reasonable time without having to partition the system. As the order of the system increases, the time-varying nature of SPSs makes simulations excessively long because the numerator in (6) increases abruptly. For instance, variation 3 took 9 days (210 hours) to simulate as unpartitioned. This fact highlights the problem that partitioning addresses: to be able to observe transient phenomena in large-scale power system models, a partitioning strategy should be implemented. From Table VI, the run-time of 9 days reduced to 7 minutes, which is 1,800 times faster.

Substep d is the major bottleneck of the partitioning approach. At each k, subsystems synchronize themselves at this substep and wait for the master thread to return the solution of the boundary variables (i.e., the vector $\mathbf{u}$). When threads synchronize, threads 2-p incur dead time and reduce the simulation's efficiency. Incidentally, *Substep f* influences *Substep d* . When switches commutate, matrix $\mathbf{\Lambda}$ must be re-factored by thread 1, which additionally increases the time spent on *Substep d*.

The efficiency of the partitioning approach was demonstrated with the run-time reductions listed in Table IV, Table V, and Table VI, which formally exploits multicore processors. The CPU usage shown by Fig. 12 shows that by having multiple threads, the processor can be utilized to a much higher potential. In unpartitioned power system simulations, *Windows* moves the single thread responsible of solving the power system equations among the cores. In partitioned simulations, when using multiple threads, the threads are *distributed* across all cores, the work is divided, and run-time reductions experienced.

The partitioning method presented is applicable to next generation integrated ship power systems (IPSs [26],[27]) as well, regardless of voltage level, AC frequency, or power architecture. At places where disconnection points can be identified, mesh tearing is valid. (Mesh tearing is also valid in DC zonal distribution.) For example, in IPSs three-phase disconnection points exist on the ring bus, in-front of transformers, downstream DC-AC inverters, etc. In the case of a DC disconnection point, mesh tearing considers one mesh at per disconnection point (between conductors a and b). In the SPS model used, the motor drives

were partitioned at the DC-links. Partitioning DC-links (as done here), or partitioning DC lines in notional IPS designs is the same situation conceptually.

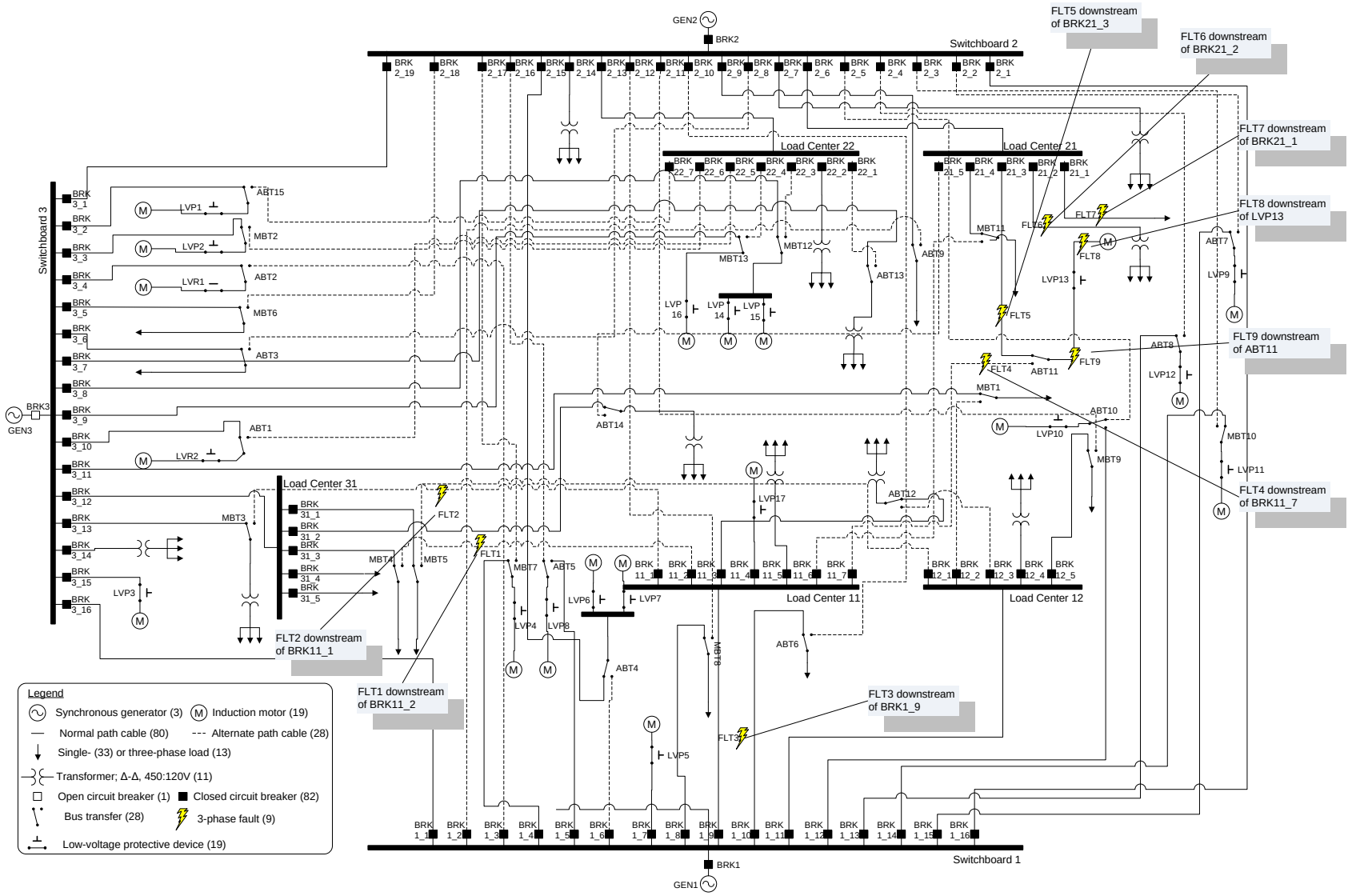

Fig. 13. One-line diagram of the ungrounded power system used to implement the partitioning approach

VII. References

[1] C. Dufour, J.-N. Paquin, V. Lapointe, J. Bélanger, and L. Schoen, "PC-cluster-based real-time simulation of an 8-synchronous machine network with HVDC link using RT-LAB and TestDrive," in *Proc. 2007 International Conference on Power Systems Transients*.

[2] P. W. Sauer and M. A. Pai. (1998). *Power System Dynamics and Stability.* Upper Saddle River, N.J.: Prentice Hall.

[3] T. Noda and S. Sasaki, "Algorithms for Distributed Computation of Electromagnetic Transients toward PC Cluster Based Real-Time Simulations," in *Proc. 2003 International Conference on Power System Transients*.

[4] R. A. Rohrer, "Circuit partitioning simplified," *IEEE Trans. Circuits and Systems,* vol. 35, pp. 2-5, 1988.

[5] J. Langston, S. Suryanarayanan, M. Steurer, M. Andrus, S. Woodruff, and P. F. Ribeiro, "Experiences with the Simulation of a Notional All-Electric Ship Integrated Power Systems on a Large-Scale High-Speed Electromagnetic Transients Simulator," in *Proc. 2007 Electric Ship Technologies Symposium,* pp. 1-5.

[6] J. R. Marti, L. R. Linares, J. Calviño, H. W. Dommel, and J. Lin, "OVNI: An Object Approach to Real-Time Power System Simulators," in *Proc. 1998 International Conference on Power System Technology (Powercon'98).*

[7] K. Miu, V. Ajjarapu, K. Butler-Purry, D. Niebur, C. Nwankpa, N. Shultz, and A. Stankovic, "Testing of shipboard power systems: a case for remote testing and measurement," in *Proc. 2005 IEEE Electric Ship Technologies Symposium*.

[8] M. Kleinberg, K. Miu, and C. Nwankpa, "Distributed multi-phase distribution power flow: modeling, solution algorithm, and simulation results," *Trans of the Soc for Modeling & Simul Intl,* vol. 84, pp. 403-412, 2008.

[9] K. L. Butler-Purry, G. R. Damle, N. D. R. Sarma, F. Uriarte, and D. Grant, "Test Bed for Studying Real-Time Simulation and Control for Shipboard Power Systems," in *Proc. 2007 Electric Ship Technologies Symposium (ESTS 2007).*

[10] C. Zhang, V. K. Vijapurapu, A. K. Srivastava, N. N. Schulz, J. Bastos, and R. Wierckx, "Hardware-in-the-loop simulation of distance relay using RTDS," in *Proc. 2007 Summer Computer Simulation Conference*

[11] R. Crosbie, "Low-cost, high-speed, real-time simulation for electric ship power systems," in *Proc. 2005 IEEE Electric Ship Technologies Symposium,* pp. 46-47.

[12] 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.

[13] F. M. Uriarte and K. L. Butler-Purry, "Multicore simulation of an AC-radial shipboard power system," in *Proc. 2010 Power Engineering Society General Meeting*.

[14] 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.

[15] 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 Delivery,* vol. 12, pp. 505-514, Jan 1997.

[16] N. Watson and J. Arrillaga. (2003). *Power Systems Electromagnetic Transients Simulation.* (1st ed.) London: IEE.

[17] K. Strunz. (2002). *Numerical Methods for Real Time Simulation of Electromagnetics in AC/DC Network Systems.* Düsseldorf: VDI Verlag.

[18] J. Arrillaga and N. R. Watson. (2001). *Computer Modeling of Electrical Power Systems.* (2nd ed.) Christchurch, New Zealand: John Wiley & Sons, LTD.

[19] P. Zhang, J.R.Marti, and H. W. Dommel, "Network Partitioning for Real-Time Power System Simulation," in *Proc. 2005 International Conference on Power System Transients*, pp. 1-6.

[20] G. Karypis, R. Aggarwal, V. Kumar, and S. Shekhar, "Multilevel hypergraph partitioning: applications in VLSI domain," in *Proc. 1997 Design and Automation Conference*, pp. 526-529.

[21] A. Brameller, M. N. John, and M. R. Scott. (1969). *Practical Diakoptics for Electrical Networks.* London: Chapman & Hall.

[22] Z. Quming, S. Kai, K. Mohanram, and D. C. Sorensen, "Large Power Grid Analysis Using Domain Decomposition," in *Proc. 2006 Design, Automation and Test in Europe, 2006. DATE '06. Proceedings*, pp. 1-6.

[23] K. W. Chan, R. C. Dai, and C. H. Cheung, "A Coarse Grain Parallel Solution Method for Solving Large Sets of Power System Network Equations," in *Proc. 2002 International Conference on Power System Technology (PowerCon '02)*, pp. 2640-2644.

[24] P. Kuffel, K. Kent, and G. Irwin, "The Implementation and Effectiveness of Linear Interpolation Within Digital Simulation," in *Proc. 1995 International Conference on Power System Transients*, pp. 499-504.

[25] K. Strunz, "Flexible numerical integration for efficient representation of switching in real time electromagnetic transients simulation," *IEEE Trans. Power Delivery*, vol. 19, pp. 1276-1283, Jul. 2004.

[26] X. Feng, T. Zourntos, K. Butler-Purry, and S. Mashayekh, "Dynamic load management for NG IPS ships," in *Proc. 2010 Power Engineering Society General Meeting*.

[27] P. Crapse, J. Wang, Y.-J. Shin, and R. Dougal, "Analysis and comparison of electric ship integrated power system architectures via harmonic metrics," in *Proc. 2008 Elec Mach Tech Symp*.

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 electromagnetic transient simulation of shipboard power systems. His research interests are in parallel computing, real-time simulation, and modeling and simulation of power systems. The author is currently with the Center for Electromechanics of the University of Texas at Austin.