

OVNI: Integrated Software/Hardware Solution for Real-Time Simulation of Large Power Systems

José R. Martí, Luis R. Linares, Jorge A. Hollman, Fernando A. Moreira
The University of British Columbia, Vancouver, Canada

Abstract—This paper presents the solution approach and implementation strategy followed in the development of UBC's OVNI ("Object Virtual Network Integrator") power system simulator. OVNI's aim is to simulate full-size power system networks for real-time and online dynamic studies. To cope with the dimensions of the problem, partitioning techniques are used, both in the system solution and in the hardware implementation. The software solution is based on the MATE ("Multi-Area Thévenin Equivalent") network partitioning framework, while the hardware implementation is realized using a cluster of interconnected PC computers.

Keywords: *Real-time OVNI simulator, the MATE algorithm, network partitioning for large systems, diakoptics, EMTP solution, PC-clusters, distributed computing.*

I. INTRODUCTION

It was Gabriel Kron in the 1950's who introduced the concept of Diakoptics (from the Greek "dia" for "very" and "kopto" for "tearing") for the solution of large power system networks (see, for example [1]). In his comprehensive review of Diakoptics in [2] in 1974, Happ already visualized its potential in multiple-computer hardware configurations. In [2], Happ proposes the concept of a "cluster of computers" with each computer in the cluster solving a separate part of the whole network.

Except for some notorious examples mostly due to Happ (see list of references quoted in [2]), Diakoptics did not fulfil its potential of becoming a universal tool for the study of large and complex power systems. Two factors determined Diakoptics' fate in the following fifty years. The main one was the success of Sparsity Techniques pioneered by Tinney and Walker [3] and advanced by many others. The other aspect was the difficulty of visualizing and formulating the Diakoptics solution in a clear and straight forward manner.

In 1975, Ho, Ruehli, and Brennan published their seminal paper [5] on Modified Nodal Analysis (MNA). MNA separates the network solution into nodal equations and branch equations. Their main motivation was to combine current-dependent elements, which are much easier defined in terms of branch equations, with the nodal equations for the rest of the network, thus maintaining the efficiency of the nodal solution using sparsity techniques. Even though the authors did not relate their work to Diakoptics, the resulting MNA equations are the same as the Diakoptics equations, with the augmented branch equations corresponding to Diakoptics' tearing links.

The main aim of Kron's Diakoptics was to achieve increased computational speeds. With the development of Sparsity techniques, however, the fact is that on normal off-line single-computer solutions there is little advantage of

Diakoptics over Sparsity in terms of solution times, and while regular or modified nodal analysis with sparsity are straight forward to understand and to implement, Diakoptics, in its original form, requires obscure manipulations of "Contour Theory of Networks". To achieve computational efficiency, both Diakoptics and Sparsity take advantage of the very sparse nature of the admittance matrix (many off-diagonal zeroes) of power networks (and most other electrical networks). Diakoptics exploits this sparsity by splitting the network into dense subsystems connected by a few links. Sparsity techniques simply avoid storing and operating with the zeroes of the full system matrix.

It has been, however, in the context of real-time simulators [6], and particularly in the case of multi-computer solutions ([8]), that the advantages of Diakoptics over Sparsity have become strongly manifested. While both Diakoptics and Sparsity take advantage of the sparsity of the system, Diakoptics maintains the identity of the component subsystems, something that is very difficult to achieve in solutions based on matrix reduction with Sparsity techniques. Maintaining the individuality of the subsystems permits, for example, the pre-calculation (before the simulation loop) of the inverse solution matrices for those subsystems which topology does not change [6], while only some of the subsystems require recalculation. This aspect alone gives Diakoptics a considerable advantage over Sparsity Techniques. Other (not less important) advantages of maintaining the individuality of the subsystems, such as latency exploitation, will be discussed later in this paper.

UBC's OVNI real-time simulator is built around the MATE ("Multi-Area Thévenin Equivalent") concept which extends the main ideas of Diakoptics in a conceptually clear and computationally efficient solution framework. MATE goes beyond Kron's Diakoptics and Ho, et al.'s MNA in that it recognizes that the subsystems split by the branch links can be represented by Thévenin Equivalents. This becomes a key point when combining together, in a simultaneous non-iterative solution, subsystems that have been solved with different solution techniques (for example with different-size Δt 's). The hardware implementation of MATE is realized in a multi-machine PC-Cluster environment.

II. MATE (MULTI-AREA THÉVENIN EQUIVALENT)

The MATE concept [7] establishes a software/hardware solution framework around which UBC's OVNI Real-Time Simulator is designed. MATE embodies the concepts of Multinode Thévenin Equivalents [4], Diakoptics [1], and Ho, et al.'s Modified Nodal Analysis [5] in a simple and general formulation. The main idea in MATE, as in Diakoptics, is

that a large network can be split into smaller parts (subsystems) by tearing apart some of its branches. Ho, et al.'s modified nodal analysis is used to combine, on the same system matrix, nodal equations for the individual subsystems with branch equations for the branches ("links") connecting the subsystems. Multinode Thévenin Equivalents are used to reconcile the independent solutions (with possibly different solution techniques) from the individual subsystems into a full-system simultaneous solution.

The concept of a Thévenin Equivalent can be thought of as an "order reduction" process in which only a small subset of nodes from an "internal" large linear system are needed to reconcile the solutions with the other subsystems. In the traditional multinode thévenin, if only N visible nodes from an internal linear system are needed to connect to an external subsystem, then the internal linear system can be represented by a thévenin source vector $[E_{TH}]_{N \times 1}$ and a thévenin impedance matrix $[Z_{TH}]_{N \times N}$ with regards to its interactions with the external system. After the solution of this reduced system, voltages and currents in the full internal linear system can be updated by injecting current sources equal to the currents that flow in the links connecting the internal and external systems. The same idea is followed in MATE, except that MATE allows for an arbitrary number of $[E_{TH}]$, $[Z_{TH}]$ subsystems.

MATE FORMULATION

Even though the concepts in MATE apply to a general circuit solution, the case of an EMTP ("Electromagnetic Transients Program") discrete-time solution [4] is used here to explain its formulation.

Consider the simple example network shown in Fig. 1. This network consists of two subsystems [A] and [B] connected by two links $\alpha 1$ and $\alpha 2$. Reference, for each subsystem, is the subsystem's local ground. External current sources can be injected from ground into any node. (Note that a current source flowing between two nodes k and m can be modelled as two current sources: one from ground into node m plus one from node k into ground.) In an EMTP solution, the current sources would include the history terms of the discrete-time circuit component models and the branches would represent the equivalent conductances.

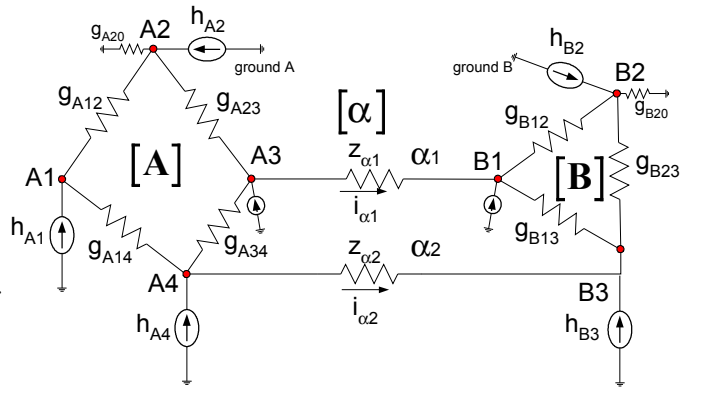


Figure 1: Sample network to explain MATE.

The notation used is illustrated next with a particular example:

- $[A]$ = admittance matrix of subnetwork [A]
- g_{A23} = admittance of branch between nodes 2 and 3 in subnetwork [A]
- G_{A23} = element (2,3) of admittance matrix [A]
- h_{B2} = current source injected from ground into node 2 of subnetwork [B]
- V_{A3} = voltage of node 3 to ground in subnetwork [A]
- $i_{\alpha 1}$ = current in link branch $\alpha 1$
- $z_{\alpha 1}$ = impedance of branch $\alpha 1$

(Link branches to other subnetworks, say [A] to [C], would have subscript β , and so on)

If the links α in Fig. 1 did not exist, we would have two independent subnetworks, [A] and [B], with corresponding nodal equations,

$$[A][v_A] = [h_A] \quad (1)$$

$$[B][v_B] = [h_B] \quad (2)$$

We can put these two equations in a "common container" as follows.

$$\begin{bmatrix} A & 0 \\ 0 & B \end{bmatrix} \begin{bmatrix} v_A \\ v_B \end{bmatrix} = \begin{bmatrix} h_A \\ h_B \end{bmatrix} \quad (3)$$

Equation (3) is the same as (1) and (2), but the bigger container will allow us to add to the matrix the links be-

	A1	A2	A3	A4	B1	B2	B3	$\alpha 1$	$\alpha 2$		
A1	G_{A11}	G_{A12}	0	G_{A14}				0	0	v_{A1}	h_{A1}
A2	G_{A12}	G_{A22}	G_{A23}	0				0	0	v_{A2}	h_{A2}
A3	0	G_{A23}	G_{A33}	G_{A34}				1	0	v_{A3}	h_{A3}
A4	G_{A14}	0	G_{A34}	G_{A44}				0	1	v_{A4}	h_{A4}
B1					G_{B11}	G_{B12}	G_{B13}	-1	0	v_{B1}	h_{B1}
B2					G_{B12}	G_{B22}	G_{B23}	0	0	v_{B2}	h_{B2}
B3					G_{B13}	G_{B23}	G_{B33}	0	-1	v_{B3}	h_{B3}
$\alpha 1$	0	0	1	0	-1	0	0	$-z_{\alpha 1}$	0	$i_{\alpha 1}$	0
$\alpha 2$	0	0	0	1	0	0	-1	0	$-z_{\alpha 2}$	$i_{\alpha 2}$	0

$=$ (4)

tween subnetworks. Expanding the submatrices and introducing the links, we get the MATE matrix of (4).

The MATE matrix of (4) is built as follows. $G_{A11} = g_{A12} + g_{A14}$, $G_{A12} = -g_{A12}$, etc. The empty areas in the matrix correspond to zero elements. The link branches are “hooked up” as follows. Since current $i_{\alpha 1}$ leaves node A3, a “1” is put in position A3 of column $\alpha 1$ and a “1” is put in position A3 of row $\alpha 1$. Since current $\alpha 1$ enters into node B1, a “-1” is put in the corresponding positions of column $\alpha 2$ and row $\alpha 2$. Similarly for link current $i_{\alpha 2}$. If, in a more general case, branches $\alpha 1$ and $\alpha 2$ were coupled through a z_{12} impedance, this impedance would occupy the off-diagonal positions of the α subblock.

After introducing the link equations, the MATE matrix (4) becomes the hybrid system of modified nodal analysis [5] containing nodal equations for the subsystems’ node voltages and branch equations for the link currents flowing between subsystems. Equation (4) also corresponds to the so-called *fundamental equation of diakoptics* ([9], p. 137), except that (4) is written directly from nodal analysis and branch currents while diakoptics requires decomposition into *primitive networks* and *connectivity* matrices.

We will now deviate from Diakoptics and from modified nodal analysis in the way (4) is solved. Our main aim will be to manipulate the submatrices in (4) in a form that preserves the individuality of the subsystems and reduces them to Thévenin equivalents.

We first rewrite (4) using partitioned matrices,

$$(5) \quad \begin{array}{c|ccc|c|c} & A & B & \alpha & & \\ \hline A & A & 0 & p & v_A & h_A \\ B & 0 & B & q & v_B & h_B \\ \alpha & p^t & q^t & -z & i_\alpha & 0 \end{array} =$$

Here,

$$(6) \quad p = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ +1 & 0 \\ 0 & +1 \end{bmatrix}, q = \begin{bmatrix} -1 & 0 \\ 0 & 0 \\ 0 & -1 \end{bmatrix}$$

are current injection matrices derived from the link branches’ hooks (“+1” if current comes out, “-1” if current goes in).

We now operate on (5), one partitioned row at a time.

We apply A^{-1} to row A:

$$(7a) \quad A^{-1} \begin{array}{c|ccc|c|c} & A & B & \alpha & & \\ \hline A & A & 0 & p & v_A & h_A \\ B & 0 & B & q & v_B & h_B \\ \alpha & p^t & q^t & -z & i_\alpha & 0 \end{array} = A^{-1} \begin{array}{c|ccc|c|c} & A & B & \alpha & & \\ \hline A & 1 & 0 & a & v_A & e_A \\ B & 0 & B & q & v_B & h_B \\ \alpha & p^t & q^t & -z & i_\alpha & 0 \end{array}$$

$$(7b) \quad \begin{array}{c|ccc|c|c} & A & B & \alpha & & \\ \hline A & 1 & 0 & a & v_A & e_A \\ B & 0 & B & q & v_B & h_B \\ \alpha & p^t & q^t & -z & i_\alpha & 0 \end{array} = A^{-1} \begin{array}{c|ccc|c|c} & A & B & \alpha & & \\ \hline A & 1 & 0 & a & v_A & e_A \\ B & 0 & B & q & v_B & h_B \\ \alpha & p^t & q^t & -z & i_\alpha & 0 \end{array}$$

$$(7c) \quad \begin{array}{c|ccc|c|c} & A & B & \alpha & & \\ \hline A & 1 & 0 & a & v_A & e_A \\ B & 0 & B & q & v_B & h_B \\ \alpha & p^t & q^t & -z & i_\alpha & 0 \end{array} = \begin{array}{c|ccc|c|c} & A & B & \alpha & & \\ \hline A & 1 & 0 & a & v_A & e_A \\ B & 0 & B & q & v_B & h_B \\ \alpha & p^t & q^t & -z & i_\alpha & 0 \end{array}$$

We apply B^{-1} to row B:

$$(8a) \quad B^{-1} \begin{array}{c|ccc|c|c} & A & B & \alpha & & \\ \hline A & 1 & 0 & a & v_A & e_A \\ B & 0 & B & q & v_B & h_B \\ \alpha & p^t & q^t & -z & i_\alpha & 0 \end{array} = B^{-1} \begin{array}{c|ccc|c|c} & A & B & \alpha & & \\ \hline A & 1 & 0 & a & v_A & e_A \\ B & 0 & B & q & v_B & h_B \\ \alpha & p^t & q^t & -z & i_\alpha & 0 \end{array}$$

$$(8b) \quad \begin{array}{c|ccc|c|c} & A & B & \alpha & & \\ \hline A & 1 & 0 & a & v_A & e_A \\ B & 0 & B & q & v_B & h_B \\ \alpha & p^t & q^t & -z & i_\alpha & 0 \end{array} = \begin{array}{c|ccc|c|c} & A & B & \alpha & & \\ \hline A & 1 & 0 & a & v_A & e_A \\ B & 0 & B & q & v_B & h_B \\ \alpha & p^t & q^t & -z & i_\alpha & 0 \end{array}$$

$$(8c) \quad \begin{array}{c|ccc|c|c} & A & B & \alpha & & \\ \hline A & 1 & 0 & a & v_A & e_A \\ B & 0 & B & q & v_B & h_B \\ \alpha & p^t & q^t & -z & i_\alpha & 0 \end{array} = \begin{array}{c|ccc|c|c} & A & B & \alpha & & \\ \hline A & 1 & 0 & a & v_A & e_A \\ B & 0 & B & q & v_B & h_B \\ \alpha & p^t & q^t & -z & i_\alpha & 0 \end{array}$$

The last row in (5) represents the link branches,

$$(9a) \quad \begin{array}{c|ccc|c|c} & A & B & \alpha & & \\ \hline A & 1 & 0 & a & v_A & e_A \\ B & 0 & B & q & v_B & h_B \\ \alpha & p^t & q^t & -z & i_\alpha & 0 \end{array} = \begin{array}{c|ccc|c|c} & A & B & \alpha & & \\ \hline A & 1 & 0 & a & v_A & e_A \\ B & 0 & B & q & v_B & h_B \\ \alpha & p^t & q^t & -z & i_\alpha & 0 \end{array}$$

But, $[v_A]$ and $[v_B]$ are determined in (7c) and (8c) in terms of $[i_\alpha]$, $[e_A]$ and $[e_B]$. Substituting in (9a),

$$(9b) \quad \begin{array}{c|ccc|c|c} & A & B & \alpha & & \\ \hline A & 1 & 0 & a & v_A & e_A \\ B & 0 & B & q & v_B & h_B \\ \alpha & p^t & q^t & -z & i_\alpha & 0 \end{array} = \begin{array}{c|ccc|c|c} & A & B & \alpha & & \\ \hline A & 1 & 0 & a & v_A & e_A \\ B & 0 & B & q & v_B & h_B \\ \alpha & p^t & q^t & -z & i_\alpha & 0 \end{array}$$

and augmenting with $[v_A]$ and $[v_B]$ to recover the original set of variables,

$$(9c) \quad \begin{array}{c|ccc|c|c} & A & B & \alpha & & \\ \hline A & 1 & 0 & a & v_A & e_A \\ B & 0 & B & q & v_B & h_B \\ \alpha & p^t & q^t & -z & i_\alpha & 0 \end{array} = \begin{array}{c|ccc|c|c} & A & B & \alpha & & \\ \hline A & 1 & 0 & a & v_A & e_A \\ B & 0 & B & q & v_B & h_B \\ \alpha & p^t & q^t & -z & i_\alpha & 0 \end{array}$$

Putting the results of (7c), (8c), and (9c) together, we obtain

$$(10) \quad \begin{array}{c|ccc|c|c} & A & B & \alpha & & \\ \hline A & 1 & 0 & a & v_A & e_A \\ B & 0 & 1 & b & v_B & e_B \\ \alpha & 0 & 0 & Z_\alpha & i_\alpha & e_\alpha \end{array}$$

where,

	A1	A2	A3	A4	B1	B2	B3	$\alpha 1$	$\alpha 2$			
A1	1							a_{13}	a_{14}	v_{A1}		e_{A1}
A2		1						a_{23}	a_{24}	v_{A2}		e_{A2}
A3			1					a_{33}	a_{34}	v_{A3}		e_{A3}
A4				1				a_{43}	a_{44}	v_{A4}		e_{A4}
B1					1			$-b_{11}$	$-b_{13}$	v_{B1}	$=$	e_{B1}
B2						1		$-b_{21}$	$-b_{23}$	v_{B2}		e_{B2}
B3							1	$-b_{31}$	$-b_{33}$	v_{B3}		e_{B3}
$\alpha 1$	0	0	0	0	0	0	0	Z_{11}	Z_{12}	$i_{\alpha 1}$		$e_{\alpha 1}$
$\alpha 2$	0	0	0	0	0	0	0	Z_{12}	Z_{22}	$i_{\alpha 2}$		$e_{\alpha 2}$

(11)

$$\begin{aligned}
a &= A^{-1}p & e_A &= A^{-1}h_A \\
b &= B^{-1}q & e_B &= B^{-1}h_B \\
Z_\alpha &= p^t a + q^t b + z & e_\alpha &= p^t e_A + q^t e_B
\end{aligned} \tag{12}$$

In expanded form, (10) looks as shown in (11).

Note the special form of columns α_1 and α_2 in (11). If the entries of $[A^{-1}]$ and $[B^{-1}]$ are as shown in (13),

$$[A^{-1}] = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix}, \quad [B^{-1}] = \begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{bmatrix} \tag{13}$$

then we can see that the effect of the current injection matrices $[p]$ and $[q]$ of (6) is to pick up the “correct” columns of $[A^{-1}]$ and $[B^{-1}]$.

The equivalent links matrix $[Z_\alpha]$ in (10), (11) is a new impedance matrix that combines the impedances of the link branches $[z]$ in (4), (5) with the “hooks” $[a]$ and $[b]$ in (10), (11) to which these branches are connected.

Equation (11) can be represented by the MATE Equivalent Circuit of Fig. 2. That is, after the transformations of Eqs. (7), (8), and (9), the original system network of Fig. 1 becomes the equivalent network of Fig. 2. Notice that the network in Fig. 2 has two parts. The upper part corresponds to the reduced-order subsystem formed by the nodes with link branches. The bottom part is not needed for the solution of the top part and represents the interactions with the rest of the internal subnetworks. The reduced system at the thévenin linking nodes, has the dimensions of the links subsystem $[Z_\alpha][i_\alpha] = [e_\alpha]$ at the bottom right corner of (11) and is formally solved from the equation,

$$\begin{bmatrix} Z_{11} & Z_{12} \\ Z_{12} & Z_{22} \end{bmatrix} \begin{bmatrix} i_{\alpha 1} \\ i_{\alpha 2} \end{bmatrix} = \begin{bmatrix} e_{\alpha 1} \\ e_{\alpha 2} \end{bmatrix} \tag{14}$$

The $[e_A]$ sources and the $[a]$ impedances inside the rectangle in Fig. 2 are the Thévenin Equivalent source vector and impedance matrix for subsystem $[A]$ in Fig. 1, as seen from the link nodes. Similarly, the $[e_B]$ sources and the $[b]$

impedances inside the rectangle in Fig. 2 are the Thévenin Equivalent sources for subsystem $[B]$ in Fig. 1. The $[e_A]$ sources and the $[a]$ impedances for subsystem $[A]$ are obtained assuming $[A]$ is isolated from the rest of the network (Eqs. (7)), and similarly for the $[e_B]$ sources and the $[b]$ impedances (Eqs. (8)). The interaction between subnetworks $[A]$ and $[B]$ occurs when injecting the link currents $[i_\alpha]$ into the corresponding nodes. This situation is very similar to the modelling of transmission lines in the EMTP [4] where the subsystems connected to the sending and receiving ends of a line can be solved independently from each other, except that now we have the extra step of injecting the link currents into the subsystems. What this means is that software/hardware solutions that have been developed for real-time transient simulators (e.g., [6], [8]) based on the decoupling effect of transmission lines can be directly extended to the general MATE case just by adding an extra computational node to solve the links subsystem of (14).

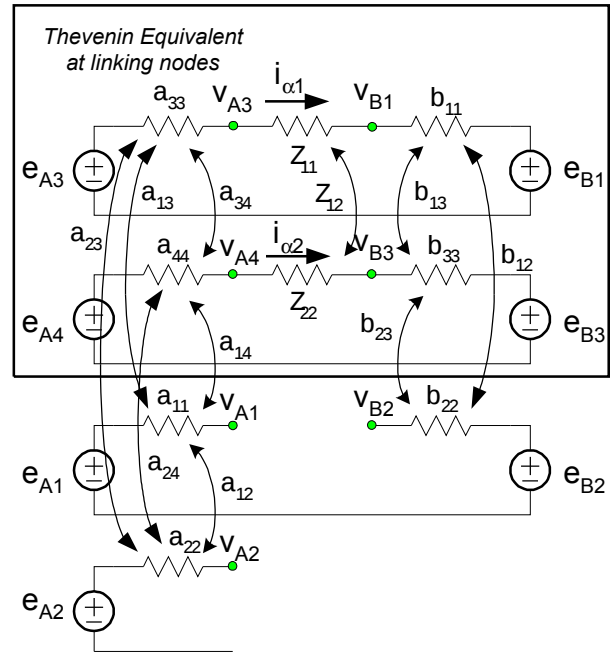


Figure 2: MATE Equivalent Circuit for the network in Fig. 1. (Only the reduced system A3, A4, B1, B3 is needed to solve for the link currents.)

APPLICATION EXAMPLES

Switching Operations

MATE links concept is particularly advantageous to represent switching operations in branches connecting subsystems. For example, in traditional EMTP modelling, when an ideal switch closes, the two nodes connected by the switch collapse into a single node. Similarly, when the switch opens two nodes are created out of one. These topological changes slow down the solution in two fronts: i) The change in node subscripts, and ii) The need to retriangularize the entire system.

Suppose in MATE that subnetworks [A] and [B] in Fig. 1 remain topologically constant during the time loop simulation (i.e., only the history sources change from time step to time step), and that all switching operations occur in the link branches. Suppose branch α_1 is an ideal switch. If the switch is closed, element z_{α_1} in the links subsystem of (4) will be zero, with no consequences to MATE's solution process. If the switch is open, the condition will be $i_{\alpha_1} = 0$, with the topological effect of eliminating row and column α_1 from (4). Even though there is now a topological change, its effect is limited to the links subsystem and submatrices $[A^{-1}]$ and $[B^{-1}]$ do not need to be retriangularized.

Reference [10] describes the development of a model for a 24-valve HVDC converter for the OVNI simulator using MATE solution concepts. A comparison of timings between this model and a standard EMTP solution (using the Microtran program [4]) resulted in a ratio of 40 times faster for the MATE model. This test was run on a single PC computer and OVNI's timing advantage was due to avoiding retriangularization of the solution matrices each time there is a change in a valve's conducting state.

Latency Exploitation

In EMTP type of solutions, the same discretization step size Δt is used for all parts of the network. This is often wasteful of computational resources because the common Δt size is determined by the solution needs of the "fastest" subsystem in the network, while this Δt may be unnecessarily small for other subsystems. A typical example is the case of HVDC converters and FACTS devices where the firing angle needs of the valves may require a small Δt resolution, while the rest of the network can be solved with a larger Δt .

Another common situation is the modelling of transmission lines. An important new line model, the z-Line Model ([11]), solves the problem of frequency dependent transformation matrices by subdividing, in length, the lossy part of the model. This model requires internal Δt 's equal to the travelling time of the total line length divided by the number of segments into which the losses are distributed. The full advantages of this model are realized when the Δt for the internal line solution can be much smaller than the Δt of the external system solution. The capability of solving a net-

work with different Δt 's in different subparts of it is termed "latency exploitation".

As discussed in [12], MATE provides an ideal framework for latency exploitation. Suppose, for example, that in the network of Fig. 1, subsystem [A] is "fast" and subsystem [B] is "slow". Then the independent solution of [A] could be performed with a small Δt , while the independent solution of [B] is performed with a large Δt . It is only for the link currents' solution and for the updating of the subsystems after the link currents are calculated that care must be exercised. While the fast subsystem is being solved at the faster pace dictated by its small Δt , the slow subsystem remains dormant ("latent") for the length of its large Δt , except for the updating of its Thévenin equivalent circuit at the link nodes. The updating of the Thévenin equivalent of the slow subsystem is done by an interpolation process between two consecutive large Δt steps of its Thévenin Equivalent and does not require the solution of the internal details of the slow subsystem during the small Δt steps of the fast subsystem solution. An upcoming paper by F. A. Moreira and J. R. Martí will present a more detailed explanation of this process.

Other Applications

MATE's Thévenin equivalent circuit of Fig. 2 provides a "common system of coordinates" for the resolution of interactions among different subsystems. Each subsystem can be first solved independently from the others with its own solution technique. The interactions among subsystems can then be resolved at the level of their Thévenin equivalents. Once the link currents are calculated they can then be injected as current sources into the particular subsystems to take into account (according to the subsystems' own internal rules) the effect of its neighbours.

In addition to the examples cited above, other mixed-solution environments could include: a) Subsystems solved with different integration techniques. For example, if all important information in subsystem [B] in Fig. 1 is 60-Hz, and it can be adequately solved using a quasi-steady-state phasor solution (as traditionally done in machine stability studies), while subsystem [A] is, for example, an HVDC converter solved using a discrete-time EMTP solution. b) Another common application where MATE results in considerable savings is, for example, when subnetwork [A] is linear and subnetwork [B] is nonlinear. In this case since $[A^{-1}]$ does not change, the solution of [A] requires minimum updating work, and the most computational work is concentrated in subsystem [B]. Further savings can be achieved if [B] is modelled as a "finite nonlinearity" (e.g., as piecewise linear), in which case all possible states of [B] can be pre-calculated and pre-stored.

III. PC-CLUSTER HARDWARE REALIZATION

The PC-cluster hardware architecture for the OVNI simulator has been conceived in two different environments: a "hard real-time environment" and a "soft real-time environ-

ment”, see Fig. 3. This organization of hardware resources aims to match the concepts of MATE described in this paper and of Latency [12], in the most flexible and efficient way. On a first-level description, the hard real-time environment is conceived as a high-performance synchronous array of solver nodes dedicated to the task of solving MATE’s decoupled subsystems. It is also at this level that interaction with external analog devices, such as protective relays or controllers takes place. This level of hardware realization of OVNI using a PC-cluster architecture has been fully implemented in our real-time laboratory at UBC and is described in detail in [8]. The soft real-time environment is currently under development in our real-time laboratory at UBC and is aimed at fulfilling the full potential of the MATE concept by extending the capabilities of the hardware to interface asynchronous events with the “hard” solution core. This level provides a general mechanism to solve the links subsystem of MATE and deliver the updating link currents to the individual subsystems. Examples of asynchronous events that are better considered at this “soft latency” level include protection system events and interarea control events in large interconnected power systems.

The selection and design of the communication hardware in each environment was mainly determined by the amount of information that needs to be interchanged between solver nodes at each solution time step, as well as by the inherent time constraints of the environments.

HARD REAL-TIME ENVIRONMENT

In the hard real-time environment, the interconnection between solver nodes takes place through an inexpensive proprietary card design developed at UBC’s Real-Time Laboratory to fit our particular simulation strategy [8]. Under the hard real-time environment, a master unit is in charge of preprocessing the input data, performing the user interface functions, and distributing the solution among the PCs of the cluster. The distribution of cases is achieved through an ethernet network with the TCP/IP protocol using a master-slave architecture. In order to avoid extra investment in hardware, the ethernet network is also used to perform remote booting of each solver node in the cluster. The solver nodes act as TCP/IP servers, while the master unit acts as the TCP/IP client. Each solver node (slave computers) of the cluster calculates its part of the network solution while performing the interaction with its neighbours through the developed I/O interface card. A unique shared synchronization source assures a perfect hard-real-time behaviour for the cluster. The solver nodes only require the computational power (CPU), memory (RAM) and standard communication ports provided in any off-the-shelf mother board.

If interaction with analog equipment is required, such as in the case of relay testing, it is easily accomplished through D/A cards connected to the corresponding slave nodes.

One key element of this environment is the use of double port memory blocks. These blocks are in charge of storing and transferring data between the cluster subsystems. The advantage of DPM lays on its independent access to read or write to any location of the block, simplifying the design.

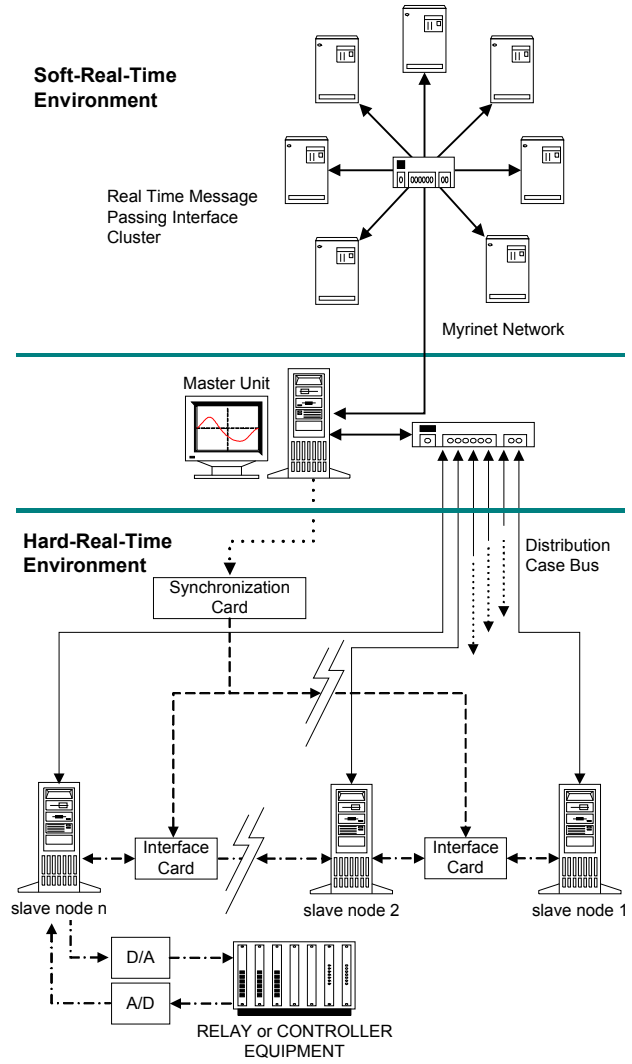


Figure 3: PC-Cluster hardware architecture.

The implemented software writes and reads to/from the DPM in different memory pages with external and shared synchronization, avoiding simultaneous access conflicts in the DPM.

SOFT REAL-TIME ENVIRONMENT

Taking advantage of the fact that in the soft real-time environment the timing constraints for communication and data interchange are less stringent than in the hard real-time environment, a more flexible communication architecture between solver nodes is proposed based on a radial solution, such as the Myrinet Network. As indicated, this concept is currently under development at UBC’s Real-Time Laboratory. A similar shared synchronization scheme to the one implemented in our hard-real-time environment is being adapted to the general case to assure real-time performance of all the nodes in the environment. The connection between environments is being implemented through a Master unit that takes advantage of the latency exploitation in the MATE circuit solution for the interaction of the fast and slow areas. Only when both environments are interacting is data from the hard real-time simulator interchanged with the

soft real-time simulator. For the rest of the time each environment computes the solution separately, with the global time as the only synchronization reference. A standard Message Passing Interface provides flexibility and compatibility across several computing platforms. Using a fine-grain global clock based on a master-slave scheme, the synchronization among host clocks can be achieved. The global clock for the soft real-time environment is integrated with the individual CPU schedulers on each node. On each node, the scheduler provides bounded-jitter time-based scheduling of computing tasks and communicating channel tasks. At initialization, the global clock module is installed and establishes a synchronized time frame. Once a global time has been established, the individual scheduler is installed. The time-based individual scheduler in the soft real-time environment uses the global time for all dispatching decisions.

The soft real-time environment can also be used to run analysis application tools on top of the real-time simulator. In this case one or more nodes of the soft-real-time environment can be assigned to process data generated in real-time by the hard and soft-real-time environments.

IV. CONCLUSIONS

The large computational power available today at low prices opens a number of opportunities in what can be achieved in system simulation. At the same time, as the goals become more ambitious (e.g., real time simulation), new solution approaches are needed to harness this power. OVNI (for Object Virtual Network Integrator) is an ambitious project that aims at attaining real-time solution speeds for large power system networks using desktop computers. OVNI proposes a hardware-mapped software solution strategy based on the MATE (Multi-Area Thévenin Equivalent) concept for network partitioning. MATE splits the network into subsystems which are first solved independently and then their solutions combined at the level of their Thévenin equivalents. By choosing the subsystems to be dense, except for a few interconnecting links, MATE achieves the advantages of sparsity without sacrificing the identity of the subsystems. By maintaining the identity of the subsystems and solving each subsystem separately, each subsystem solution can be optimized by using the solution technique that best suits that subsystem characteristics, for example, different integration step sizes, different integration rules, phasor versus EMTP solution, etc. An exact simultaneous solution to evaluate the link currents among subsystems is then performed at the level of the subsystems' Thévenin equivalents. Since a given subsystem "sees" the others only through this multi-Thévenin equivalent, the updating of the internal details within the subsystems is up to the solution needs of the particular subsystem.

A hardware architecture that mimics the MATE algorithm in its partitioning of the full system into solution nodes has been implemented in the form of a hard real-time environment using a cluster of PC computers. The extension of this cluster to better serve asynchronous events in a soft real-time environment is currently under development. Both soft-

ware and hardware in OVNI have been designed for easy maintenance and upgradability. The software has been written in C++ using high-level object-oriented design techniques to allow for the addition of new modules or the upgrading of existing ones with a minimum of modifications to the base code. The hardware of the PC-Cluster design uses standard off-the-shelf components and allows for easy expansion with additional units and easy replacement of existing ones with a minimum of system retuning.

V. REFERENCES

- [1] Gabriel Kron, "Tensorial Analysis of Integrated Transmission Systems, Part III - The Primitive Division," AIEE Transactions, Vol. 71, Pt. 3, pp. 814-821, 1952.
- [2] Harvey H. Happ, "Diakoptics - The Solution of System Problems by Tearing," Proceedings of the IEEE, Vol. 62, No. 7, pp. 930-940, July 1974.
- [3] W. F. Tinney, J. W. Walker, "Direct Solution of Sparse Network Equations by Optimally Ordered Triangular Factorization," Proceedings IEEE, Vol. 55, pp. 1801-1809, November 1967.
- [4] Hermann W. Dommel, *EMTP Theory Book*, 2nd ed., Microtran Power System Analysis Corporation, Vancouver, British Columbia, Canada, pp. 12-3 to 12-9, 1992-1996.
- [5] C. W. Ho, A. E. Ruehli, P. A. Brennan, "The Modified Nodal Approach to Network Analysis," IEEE Transactions Circuits and Systems, CAS-22, pp. 504-509, June 1975.
- [6] J. R. Martí, L. R. Linares, "Real-Time EMTP-based transients simulation," IEEE Transactions on Power Systems, Vol. 9, No. 3, pp. 1309-1317, August 1994.
- [7] J. R. Martí, L. R. Linares, J. Calviño, H. W. Dommel, J. Lin, "OVNI: An object approach to real-time power system simulators," Proceedings of the 1998 International Conference on Power System Technology, Powercon'98, Beijing, China, August 18-21, 1998, pp. 977-981.
- [8] J. R. Martí, J. A. Hollman, J. Calviño-Fraga, "Implementation of a Real-Time Distributed Network Simulator with PC-Clusters," PARELEC'2000 IEEE Computer Society Proceedings, Trois-Rivières, Québec, August 27-30, 2000, 5 journal pages.
- [9] A. Brameller, M. N. John, M. R. Scott, *Practical Diakoptics for Electrical Networks*, Chapman & Hall Ltd., 1969.
- [10] S. Acevedo, L. R. Linares, J. R. Martí, Y. Fujimoto, "Efficient HVDC converter model for real time transients simulation," IEEE Transactions on Power Systems, Vol. 14, No. 1, pp. 166-171, February 1999.
- [11] F. Castellanos, J. R. Martí, "Full frequency-dependent phase-domain transmission line model," IEEE Transactions on Power Systems, Vol. 12, No. 3, pp. 1331-1339, August 1997.
- [12] F. A. Moreira, J. A. Hollman, L. R. Linares, J. R. Martí, "Network Decoupling by Latency Exploitation and Distributed Hardware Architecture," Transactions of International Conference on Power Systems Transients, IPST'2001, Rio de Janeiro, Brazil, June 24-28, 2001.