

# Implementation of accelerated models for EMT tools

B. Clerc, C. Martin, S. Denetière

**Abstract**— For many applications (long EHV AC cables installations, HVDC interconnections...), EMT (ElectroMagnetic Transients) simulation studies require the use of frequency dependent component and grids models. Time domain simulations of such complex models are quite challenging for EMT tools. Actually these models usually require a large number of arithmetic operations to be performed at every time-step in time-domain. This constraint is especially difficult to address for real-time simulation.

First this paper describes solutions to accelerate full frequency dependent lines and cables models. These solutions have been designed to speed up the simulations without any reduction in accuracy. The main idea is to limit as much as possible to number of arithmetic operations without any simplifications. This is a general approach that is suitable for offline and real-time tools. Some EMT studies performed on the French Transmission grid and including wideband cables models are presented to prove the effectiveness of the proposed solution.

Second, the paper presents the implementation of an accelerated State Space Model algorithm based on the use of a sparse matrices package. The State Space representation is very useful to describe the frequency dependency of large passive electrical networks in time domain simulations and therefore is very efficient for multi-terminals network reduction without losses of frequency dependency and coupling effect at boundaries. But the sizes of State Space matrices are quite important leading to complex matrices manipulations and long execution times. A new State space model that takes benefit from the sparsity of state space matrices has been implemented to address this issue. This code has been interfaced with the off line EMTP tool (using the DLL interface) and with the HYPERSIM Real-Time Simulator (using the HYPERSIM User Code Model component).

The models presented in this paper are now used for EMT studies on the French transmission grid.

**Keywords:** Software development, Simulation tool, Real Time Simulation, Electromagnetic transient studies, Power Electronics Controller Replica, Solution Methods and Algorithms.

## I. INTRODUCTION

Many power electronics based equipment will be installed on the French transmission grid in the few coming years. SVCs are set up on the network to overcome voltages instabilities and HVDC links are designed to develop underground or submarine interconnections. These new devices are characterized by numerous power components but also by the use of complex algorithms in their control system. To analyze the dynamic performances of these devices connected to the grid, Rte has built up its own real-time simulation laboratory including a real time simulator based on the HYPERSIM software and hosting HVDC & FACTS control system replicas. This facility gives tremendous opportunities to study the integration of HVDC & FACTS equipment on the grid.

In this context, Rte has set up a collaboration agreement on the development of HYPERSIM with Hydro-Québec in 2012. This collaboration enables sharing development efforts and expertise.

Real time simulations including power electronics control system in the loop generally requires accurate frequency dependent models for all of the different components of the grid. In Europe and especially in France, HVDC links include long underground or submarine DC cables. To analyze transients on these equipment frequency dependent cables models shall be used. Moreover, Rte is setting up several long underground AC cables on its grid and these facilities can have a significant impact on close electronic power systems (for example, the long Merlatière – Recouvance 225 kV AC underground cable near the Merlatière SVC). In these conditions, Rte decided to implement a frequency dependent line and cable model (also known as the wide Band model or universal line/cable model) in the HYPERSIM software.

This model is based on the work presented in [5] in which wide Band theory is described. Some solutions are also proposed to speed up execution times by reducing numerical computations without any simplifications.

The present paper provides explanations about adaptations of this wide band model to make it suitable for real time simulations in HYPERSIM.

One of the major issues of transient analysis with EMT tools is the simulation time for large scale networks. Large scale network modeling is often required for an accurate representation especially to analyze low-frequency transients.

The 400 kV French grid has been completely modeled in EMTP as described in [6]. For some studies, such as long

---

B. Clerc, C. Martin and S. Denetière are with RTE - Réseau de Transport d'Electricité, Tour Cœur Défense, 92932 PARIS La Défense, France (e-mail of corresponding author: bertrand.clerc@rte-france.com).

HVAC cable integration studies, additional part of the transmission grid including lower voltage levels (225 kV, 90 kV and 63 kV) have been added to the 400 kV grid model..

Frequency Dependent Network Equivalents (FDNE) are used in [1] with the component called “state space equations” in EMTP-RV. This method keeps a good frequency representation of the network and provides acceptable computation time for time-domain simulations. It also makes possible to provide EMTP models of a grid without revealing detailed and confidential information about the network.

This paper presents a solution to increase performance of actual state space component of EMTP using the sparse characteristics of state space matrices obtained by FDNE generation.

Comparisons are made between a detailed network model in time domain, the built-in state space block component and the improved one. This paper details savings in CPU time for a set of state space matrices directly coming from the FDNE process.

The method to interface the EMTP code with the Real-Time Simulator HYPERSIM is finally introduced.

## II. WIDEBAND LINES AND CABLES MODELS

### A. Overview

The HYPERSIM real time simulation tool provides users with several frequency dependent line models. These models are based on the FD line model [7] which is only valid for overhead lines. The main limitation of this model is the use of a constant transformation matrix. This limitation prevents users from modeling underground cables.

### B. Main principles of the Wide Band theory

Considering a line or a cable composed of N conductors with a length equal to L and two ports, one at each end, the general model of this line or cable to take into account the waves' propagation is shown in Figure 1.

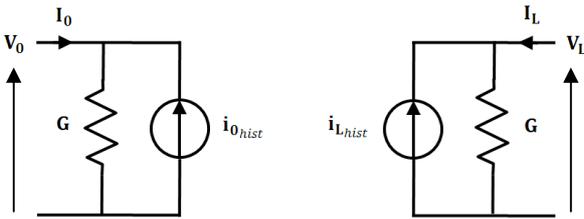


Fig. 1. General line model for wave propagation

Relations between both ends of the line are, in frequency domain:

$$I_0 - Y_C V_0 = -H(I_L + Y_C V_L) \quad (1)$$

$$I_L - Y_C V_L = -H(I_0 + Y_C V_0) \quad (2)$$

$Y_C$  is the characteristic admittance matrix defined by:

$$Y_C = \sqrt{YZ^{-1}} \quad (3)$$

$H$  is the propagation matrix defined by:

$$H = e^{-\sqrt{YZ}L} \quad (4)$$

$Z$  and  $Y$  are respectively the series and shunt coupled impedance matrices of the line or the cable which can be computed at different frequencies from the geometrical and electrical data. From these matrices, rational functions for  $Y_C$  and  $H$  matrices can be obtained from a vector fitting process [3] as described in equation (5).

$$Y_C = G_0 + \sum_{i=1}^{N_y} \frac{G_i}{s - q_i} \quad (5)$$

Where  $N_y$  is the number of  $q_i$  poles,  $G_0$  a constant matrix and  $G_i$  the residues matrices for each pole.

$$H = \sum_{k=1}^{N_m} e^{-s\tau_k} \sum_{i=1}^{N_p(k)} \frac{R_{k,i}}{s - p_{k,i}} \quad (6)$$

Where  $N_m$  is the number of modes,  $\tau_k$  the propagation delays of each mode,  $N_p(k)$  the number of  $p_{k,i}$  poles for each mode and  $R_{k,i}$  the residues matrices for the corresponding poles.

After converting equations (1), (2), (5) and (6) into time-domain and applying trapezoidal integration method, we final get the following expression for the history terms of the current sources:

$$i_{0_{hist}} = \sum_{i=1}^{N_{yR}} (\hat{q}_i + 1) \hat{G}_i \Psi_{i_0} + 2 \sum_{i=1}^{\frac{N_{yC}}{2}} y_{R_{i_0}} - \sum_{i=1}^N \left( \sum_{k=1}^{M_{iR}} x_{i,k_0} + 2 \sum_{k=1}^{\frac{M_{iC}}{2}} x_{R_{i,k_0}} \right) \quad (7)$$

The first part of this expression is related to the characteristic admittance matrices with the use of the sets of internal variables named  $\psi_i$  and  $y_{R_i}$  defined by (for the 0 end)

$$\Psi_i = \hat{q}_i \Psi'_i + v'_0 \quad (8)$$

$$y_{R_i} = \alpha y'_{R_i} - \beta y'_{i_i} + \tilde{C} v'_0 \quad (9)$$

$$y_{i_i} = \alpha y'_{i_i} + \beta y'_{R_i} + \tilde{E} v'_0 \quad (10)$$

Primed variables as  $\psi'_i$  correspond to values' variables ( $\psi_i$  in our case) during the previous time step. By extension, double primed variables are equals to the value of this variable two time steps before.

The second part is related to the propagation matrices with the use of the sets of internal variables named  $x_{i,k}$  and  $x_{R_{i,k}}$  defined by (where  $x_{R_{i,k}}$  is the real part of  $x_{i,k}$  for complex poles):

$$x_{i,k_0} = \hat{p}_{i,k} x'_{i,k_0} + \hat{R}_{i,k} [\mathbf{i}_{f_{wL}}(t - \tau_i) + \mathbf{i}'_{f_{wL}}(t - \tau_i)] \quad (11)$$

The definitions of the different constants ( $\hat{q}_i, \hat{G}_i, \dots$ ) can be found in [5].

### C. Reduction of computations number

HYPERSIM is a real time simulation tool. Thus computations have to be carried out within a time step which is usually between 15 $\mu$ s to 100  $\mu$ s. One of the main issue to develop a model for real time simulation is to limit as much as possible the computational operations and to avoid simplification that could be detrimental for the model validity.

Solutions proposed in [5] have been implemented in HYPERSIM. The first improvement results from the fact that the most part of poles are real. By separating real poles treatment from complex poles one, it is possible to avoid useless computations (with imaginary parts equal to zero) by using real internal variables ( $\psi_i$  for  $Y_C$  and  $x_{i,k}$  for H).

Secondly, complex poles resulting from the vector fitting process appear by pairs. As proposed in [5], it is possible to combine their corresponding internal variables to obtain a new internal variable containing all the required information in the real part ( $y_{R_i}$  for  $Y_C$  and  $x_{R_{i,k}}$  for H).

Finally, [5] has proposed a change of variable after the admittance matrix discretization process to save computations to obtain a new variable which can be easily updated at each time step.

Implementation of this wide band model is done with a special care to avoid useless calculations, such arithmetic operations involving two constant parameters (pre calculations are made in a preparatory functions). Moreover, intermediate variables are used within the execution function's loops when computations can be mutualised.

### D. Algorithm adaptation for a Real Time use

The general outline of the wide band model is to compute the values, at each time step, both ends' current sources (history terms  $i_{0_{hist}}$  and  $i_{L_{hist}}$  in (7)) from the voltages just calculated during the nodal resolution and the updated internal variables. These calculated historical values are used at the beginning of the next time step, before voltages calculations. As we can see on Figure 1, both ends of the line are decoupled: corresponding computations can be done in separated tasks. Nevertheless, it induces a constraint in the fact that lines or cables with propagation delays lower than the time step value cannot be run with this model. As a matter of the fact, updating internal variables of one end needs information from the other end (see expression (11)). To compute the values of the 0 end's  $x_{i,k}$  variables, which are linked to the propagation matrices, the values of travelling currents coming from the L end are required:

$$i_{fwL} = I_L + Y_c V_L \quad (12)$$

If one of the propagation delays  $\tau_i$  is lower than the simulation time step, the computation of  $x_{i,k_0}$  requires the values of  $V_L$ ,  $V_L'$  and  $V_L''$  to interpolate the values of  $i_{fwL}$  and  $i'_{fwL}$ . But  $V_L$ , as  $V_0$ , have not been calculated yet.

So the two decoupled tasks have to exchange information to calculate the historic terms and to solve the nodal equations at the connection point. Modeling large grids leads to numerous computations for the nodal resolution. In real time simulation, it is necessary to reduce the number of operations

in these tasks. That's why the wide band model has been implemented in HYPERSIM with three separated tasks: two tasks operating line/cable ends and the nodal solution of their connected network and one middle task dedicated to the propagation system. The role of this third task is to store the needed values of  $i_{fwL}$  and  $i_{fw_0}$  to calculate the  $x_{i,k}$ . The size of the storage buffer depends on the highest propagation delay  $\tau_i$ . The drawback of this structure is that propagation delays shall be greater than twice the time step. Actually, as tasks are parallelized and as no data can be exchanged during time step's computations, one time step is necessary to transmit last voltage value from one end to the middle task and another one time step is needed to transmit history terms from the middle task to the second end's task. So computations for internal variables related to the propagation matrix are performed by the middle task before being sent to ends' task for the final computations of  $i_{hist}$ . In contrast, the internal variables linked to the characteristic admittance matrix depend on  $V_L'$  or  $V_0'$  : in a first approach their computations have to be operated by ends' tasks. But these variables' definitions can be developed and separated in two parts. Expression (8) for the 0 end becomes:

$$\Psi_i = \hat{q}_i^2 \Psi_i'' + \hat{q}_i V_0'' + v_0' \quad (13)$$

The two first members of this expression can be computed by the middle task.

As complex poles for characteristic admittance matrix are handled by expressions (9) and (10), developing these expressions leads to a new definition for the part of history terms linked to characteristic admittance matrix:

$$iY_{Chist} = iY_C V_{hist} + iY_C \Psi_{hist} \quad (14)$$

With

$$iY_C \Psi_{hist} = \sum_i \hat{q}_i (\hat{q}_i + 1) \hat{G}_i \Psi_i' + 2 \sum_i \alpha y_{R_i}' - \beta y_{I_i}' \quad (15)$$

And

$$iY_C V_{hist} = \left( \sum_i (\hat{q}_i + 1) \hat{G}_i + 2 \sum_i \tilde{C} \right) v_0' = G_{cv} * v_0' \quad (16)$$

If (16) has to be computed by the ends' tasks because it depends on the voltage calculated at the previous time step, (15) can be handled by the middle task as well as the different internal variables  $\Psi_i$ ,  $y_{R_i}$  and  $y_{I_i}$ . As a matter of fact,  $iY_C \Psi_{hist}$  depends on  $\Psi_i'$ ,  $y_{R_i}'$ ,  $y_{I_i}'$  which depend on  $V''$ , the value of the end's voltage calculated two time steps before. As  $iY_C V_{hist}$  only depends on the previous values of the voltages vector which has to be multiplied by a constant matrix ( $G_{cv}$ ), ends' tasks are easily calculated. Operations are summarized on the next diagram:

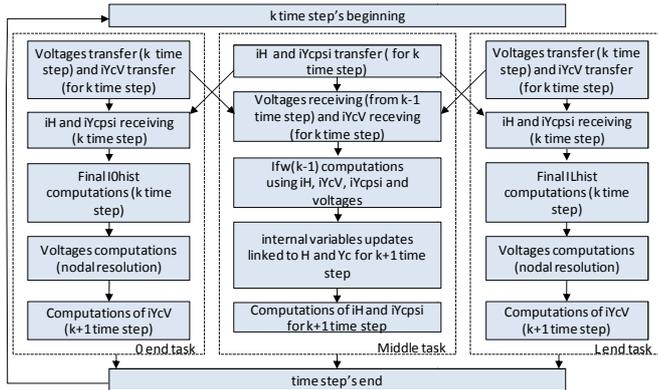


Fig. 2. HYPERSIM wide band algorithm's diagram

In this diagram,  $I_H$  is the part of  $i_{hist}$  related to the propagation matrices using  $x_{i,k}$  and  $x_{R_{i,k}}$  sets of internal variables.

### E. Validation with EMTP

Although lot a modifications and variables changes have been made to the initial expressions, model equations are strictly identical to those which have been coded in EMTP-RV. To ensure that no mistake has been introduced in the code, it is possible to compare model's behaviours in each tool.

A 15 km 3-phase single core cable energized by a very simple network composed of a 225 kV voltage source with a RL impedance is modeled in HYPERSIM. The fitting of this 6-conductors cable leads to the following characteristics (frequencies range between 0.01 Hz and 1 MHz with 10 points per decade):

- 11 real poles in the characteristic admittance matrix,
- 34 poles, including 6 complex conjugate poles in the propagation matrix distributed in 4 modes,
- Shortest propagation delay equal to 94  $\mu$ s.

Sheaths are connected to earth at both ends of the cable.

The following figure shows the components in the HYPERSIM GUI:

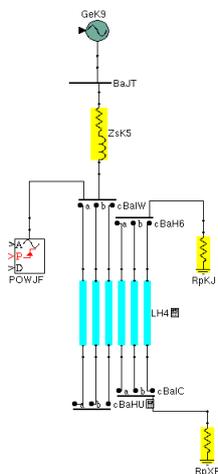


Fig. 3. A 3-phase single ore cable energized in HYPERSIM

The same system is assembled in EMTP-RV. The cable is energized without any load at  $t=0$ . Voltages at cable terminals obtained in HYPERSIM and EMTP are compared in Figure 4:

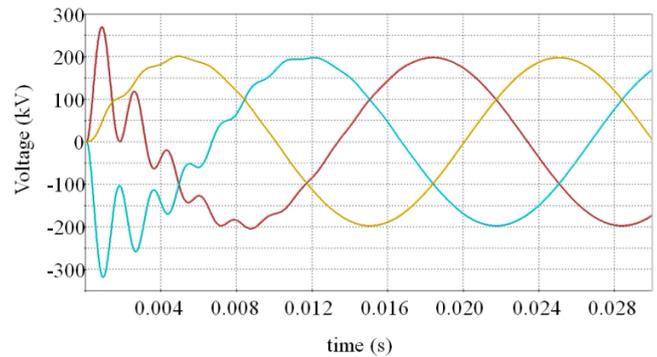


Fig. 4. Voltages at cable terminals during energization without any load

Curves are exactly superimposed (minimal relative difference lower than  $1e^{-10}$ ). A special care has been taken about computations accuracy to ensure that resulting differences between EMTP-RV and HYPERSIM implementations were only due to architecture differences and numerical operations arrangements.

### F. Execution times in HYPERSIM

A typical time step used in real time simulation is 50  $\mu$ s. HYPERSIM is provided with a tool called HYPERVIEW giving the possibility to get some information about execution times of operations of each CPUs used by the simulation. The previous case can be tuned to separate the 3 tasks and isolate the middle task which embeds the main algorithm for this wide band model. Different versions of the case are run, for several sets of parameters for the cable, in Real Time, on a SGI UV100 SUSE Linux Enterprise Server (E7-8837 Intel Xeon CPUs at 2.67 Ghz with a memory cache size of 24 Mo):

TABLE I  
EXECUTION TIMES TABLE

| Cases  | Modes number | Poles number in H | Poles number in Yc | Execution time |
|--------|--------------|-------------------|--------------------|----------------|
| case 1 | 1            | 1                 | 1                  | 1 $\mu$ s      |
| case 2 | 4            | 36                | 14                 | 8,5 $\mu$ s    |
| case 3 | 4            | 80                | 20                 | 17,5 $\mu$ s   |

The resulting execution speed is suitable for real time simulations with HYPERSIM.

### III. STATE SPACE MODEL

#### A. Overview

A built-in state space block component is provided in EMTP-RV. It is especially useful to accelerate time domain simulations of large passive electrical networks by converting them into state space equations with FDNE (Frequency Dependent Network Equivalent) modeling. The process of FDNE generation [1] is as follows. The impedance matrix of the network (17) is measured in the frequency domain with the "Frequency scan" option of EMTP-RV. The « Input impedance, 1-phase » component is connected to each external bus of the area to be reduced in order to calculate  $Z_{ij}(s)$ .

$$Z(s) = \begin{bmatrix} Z_{aa}(s) & Z_{ab}(s) & Z_{ac}(s) \\ Z_{ba}(s) & Z_{bb}(s) & Z_{bc}(s) \\ Z_{ca}(s) & Z_{cb}(s) & Z_{cc}(s) \end{bmatrix} \quad (17)$$

Then, the frequency domain responses are fitted with a rational function-based model and are finally converted into a state space model (18).

$$Y(s) = C(sI - A)^{-1}B + D + sE \quad (18)$$

To perform the fitting, Rte uses an external tool developed by SINTEF, as part of the project "Electromagnetic transients in future power systems". The FDNE approach significantly reduces calculation time with negligible loss of accuracy. Indeed, the number of electrical nodes of the final component is limited to the bus of interest, internal nodes being eliminated. However, to enhance the whole FDNE process, an accelerated state space model has been developed. This new module, described below, takes advantage of state space matrices sparsity to significantly reduce the number of arithmetic operations performed at model initialization and at every time-step in time-domain.

#### B. Sparse matrix theory

Sparse matrix data structures and algorithms are taken from [2]. Most of elements in a sparse matrix are zero. They are many ways of storing a sparse matrix, the compressed-column form is useful in most sparse matrix algorithms. As an example, the matrix (19a) is transformed into three arrays (19b), (19c) and (19d)

$$M = \begin{bmatrix} 0 & c & 0 \\ a & 0 & 0 \\ b & 0 & d \end{bmatrix} \quad (19a)$$

$$\text{int } p[] = \{0, \quad 2, \quad 3, \quad 4\} \quad (19b)$$

$$\text{int } i[] = \{1, \quad 2, \quad 0, \quad 2\} \quad (19c)$$

$$\text{double } x[] = \{a, \quad b, \quad c, \quad d\} \quad (19d)$$

Row indices of entries in column  $j$  are stored in  $i[p[j]]$  through  $i[p[j+1]-1]$  and the corresponding numerical values are stored in the same locations in  $x$ .

The only algorithm used for calculation at every time step is

matrix-vector multiplication (section III-E-4). The pseudocode for computing  $y = Mx + y$  is

```
for j = 0 to n - 1 do
  for each i for which  $m_{ij} \neq 0$  do
     $y_i = y_i + m_{ij}x_j$ 
```

Algorithm complexity for matrix-vector multiplication can be compared between dense matrix algorithm and the sparse one. If  $T_d$  represents the number of floating-point operations (multiplication and addition in that case) for computing  $y = Mx + y$  for the dense matrix algorithm and  $T_s$  for the sparse one, we have:

$$T_d(M) = 2 * (\text{total number of entries of } M) \quad (20a)$$

$$T_s(M) = 2 * (\text{number of nonzero entries of } M) \quad (20b)$$

Therefore, calculation time for computing  $y = Mx + y$  is directly linked to  $M$  matrix sparsity.

#### C. Sparsity of FDNE generation matrices

The tool developed by SINTEF provides real state space matrices with a specific structure. The generation process allows enforcing  $D$  and  $E$  matrices to be zero.  $A$  is block diagonal, with 1x1 or 2x2 blocks (in case of a complex conjugate pair of poles during rational approximation of impedance matrix (17)). Then  $B$  matrix structure is as follows:

$$B = \begin{bmatrix} B_1 \\ \vdots \\ B_n \end{bmatrix} \quad (21)$$

$n$  is the number of external pins, and  $B_i$  contains only zero entries except on its  $i^{\text{th}}$  column (that can contain values 0, 1 or 2).  $C$  is a dense matrix.

Analyzing pattern of each matrix is substantial because we have to ensure that some matrices participating in steady state resolution (section III-D) or time domain equations (section III-E) are sparse so as to justify the use of sparse matrix algorithms.

#### D. Steady state initialization

To start simulation in steady state, admittance matrix of the component (18) has to be calculated. Let  $\tilde{A} = sI - A$ , the most time-consuming calculation is  $\tilde{A}$  inversion. This matrix size grows with the number of poles chosen during Vector Fitting.  $\tilde{A}$  is sparse, it has the same nonzero pattern as  $A$ . The inversion is accelerated using complex sparse algorithms, a KLU factorization of  $\tilde{A}$  (decomposition of  $\tilde{A}$  into a permutation matrix  $K$ , a lower triangular matrix  $L$  and an upper triangular matrix  $U$ ) combined with sparse linear systems solving. This algorithm is especially efficient when a frequency scan is performed because (18) is solved for every frequency of the specified interval. Performance comparisons of the initialization phases are made in section III-F between the built-in state space block component and the improved one.

## E. State space equations in time domain

### 1) General equations

Time domain version of equation (18) is (22a) and (22b):

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \quad (22a)$$

$$\mathbf{i} = \mathbf{C}\mathbf{x} + \mathbf{D}\mathbf{u} + \mathbf{E}\dot{\mathbf{u}} \quad (22b)$$

$\mathbf{i}$  is the vector of currents entering the device,  $\mathbf{x}$  the state vector and  $\mathbf{u}$  is the vector of voltages at connectivity pins. Applying an integration method (Trapezoidal or Backward Euler) with a time-step  $T$ , we get the general equation of the current (23) for a linear and passive component (Norton equivalent equation):

$$\mathbf{i}(t) = \mathbf{G}_{eq}\mathbf{u}(t) + \mathbf{i}_{hist}(t - T) \quad (23)$$

### 2) FDNE generation case

When state space matrices are coming from FDNE generation, ensuring  $\mathbf{D}$  and  $\mathbf{E}$  to be zero (section III-C), and trapezoidal integration method is used, the conductance matrix  $\mathbf{G}_{eq}$  and historic currents expressions  $\mathbf{i}_{hist}$  are given below

$$\mathbf{G}_{eq} = \mathbf{C}\boldsymbol{\varphi} \frac{T}{2} \mathbf{B} \quad (24a)$$

$$\mathbf{i}_{hist}(t) = \mathbf{C}\boldsymbol{\varphi}\boldsymbol{\beta}\mathbf{x}(t) + \mathbf{C}\boldsymbol{\varphi} \frac{T}{2} \mathbf{B}\mathbf{u}(t) \quad (24b)$$

With

$$\boldsymbol{\varphi} = \left( \mathbf{I} - \frac{T}{2} \mathbf{A} \right)^{-1} \quad (24c)$$

$$\boldsymbol{\beta} = \left( \mathbf{I} + \frac{T}{2} \mathbf{A} \right)^{-1} \quad (24d)$$

$$\mathbf{x}(t) = \boldsymbol{\varphi}\boldsymbol{\beta}\mathbf{x}(t - T) + \boldsymbol{\varphi} \frac{T}{2} \mathbf{B}(\mathbf{u}(t) + \mathbf{u}(t - T)) \quad (24e)$$

### 3) Time domain initialization

Every constant matrix is calculated at preparatory phase. Such as steady state initialization (section III-C), sparse algorithms are really efficient for  $\boldsymbol{\varphi}$  and  $\boldsymbol{\beta}$  calculation because sparse matrices inversions are required.

### 4) Time domain solution

The vector of historic current,  $\mathbf{i}_{hist}$ , and the state vector  $\mathbf{x}$  are recalculated at every time step using voltages, currents and state vector of the previous time step. Ensuring time domain optimized operations is essential especially for real time simulation (section III-G). Equations (24b) to (24e) show that historic currents and state vector resolution are a sum of matrix-vector multiplications. Ignoring multiplications by constants  $\frac{T}{2}$  or  $\frac{2}{T}$ , matrices that are used history terms calculation are  $\boldsymbol{\varphi}\boldsymbol{\beta}$ ,  $\boldsymbol{\varphi}\mathbf{B}$ ,  $\mathbf{C}\boldsymbol{\varphi}\boldsymbol{\beta}$ ,  $\mathbf{C}\boldsymbol{\varphi}\mathbf{B}$ . Those matrices have a particular nonzero structure. Indeed nonzero pattern analysis done in section III-B allows to determine sparsity of those matrices. As inversion of a block diagonal matrix is also a block diagonal matrix with the same nonzero structure and

multiplication of two block diagonal matrices of the same nonzero pattern gives a block diagonal matrix of the same nonzero pattern (only if numerical cancellation is ignored), we can conclude that  $\boldsymbol{\varphi}$ ,  $\boldsymbol{\beta}$ ,  $\boldsymbol{\varphi}\boldsymbol{\beta}$ , and  $\mathbf{A}$  have the same nonzero pattern.

Hence  $\boldsymbol{\varphi}\boldsymbol{\beta}$  is sparse, and so is  $\boldsymbol{\varphi}\mathbf{B}$  matrix because  $\mathbf{B}$  is sparse. Finally matrices that are involved in  $\mathbf{x}$  calculation are sparse and its determination, equation (24e), is faster with sparse algorithms. Nevertheless, as  $\mathbf{C}$  matrix is dense,  $\mathbf{C}\boldsymbol{\varphi}\boldsymbol{\beta}$  and  $\mathbf{C}\boldsymbol{\varphi}\mathbf{B}$  are too and equation (24b) is not accelerated with sparse algorithm.

$\boldsymbol{\varphi}$ ,  $\boldsymbol{\beta}$ ,  $\boldsymbol{\varphi}\boldsymbol{\beta}$ ,  $\boldsymbol{\varphi}\mathbf{B}$

## F. Model's performance

### 1) Time domain comparisons

Performance tests have been performed by fitting the frequency impedance of the 400kV French grid which include  $X$  substations,  $Y$  lines,  $Z$  loads and  $V$  3-phases nodes. The network is fitted into a linear and passive component "State-space" with the Norton equivalent current source that represents active components of the detailed network.

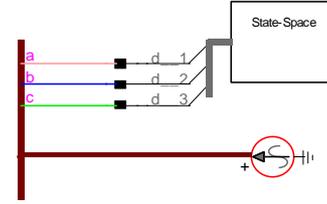


Fig. 5. State space model equivalent of the 400kV French grid

To produce a network equivalent, the number of poles during FDNE generation has to be high enough. As explained in [1], a reliable rational modeling may require 150 pole-residue terms.

Results rounded up or down to the nearest first decimal, between state space models and detailed network, are presented below. (1 s duration, 10 $\mu$ s time step: 100,000 time steps). CPU times have been evaluated with EMTP-RV version 2.6 for an Intel Core i7- 3630QM 2.40GHz processor.

TABLE II  
CPU TIME COMPARISON FOR TIME DOMAIN RESOLUTION

| Poles number | Detailed network | Built in state space model | Accelerated state space model |
|--------------|------------------|----------------------------|-------------------------------|
| 50           | 433.4s           | 3.2s                       | 1.5s                          |
| 150          | 433.4s           | 17s                        | 1.8s                          |
| 300          | 433.4s           | 111.4s                     | 2.5s                          |

The difference between the state space components would be even larger if we increase the number of external nodes. Indeed, state space matrices sizes depend not only on poles number but also on connectivity pins number.

### 2) Initialization comparisons

The same network equivalent is used to compare initialization times that include preparation, reading data and steady-state initialization.

TABLE III  
CPU TIME COMPARISON FOR INITIALIZATION STEP

| Poles number | Detailed network | Built in state space model | Accelerated state space model |
|--------------|------------------|----------------------------|-------------------------------|
| 50           | 33.7s            | 0.1s                       | 0.1s                          |
| 150          | 33.7s            | 2.5s                       | 0.4s                          |
| 300          | 33.7s            | 36.4s                      | 1.4s                          |

As explained in section III-D, steady-state initialization requires a complex sparse matrix inversion. As algorithms for the built in state space model do not handle matrix sparsity, for a large number of poles, initialization phase can be slower than the detailed network.

### G. HYPERSIM integration

The code developed in EMTP has been written in C++ language, the same language as the real time simulator HYPERSIM's GUI, in order to facilitate interfacing model with this software. The HYPERSIM objects' palette provides a powerful component named User Code Model giving the possibility to users to develop their own coded models. Within this code, it is possible to use functions from external libraries (.so files). EMTP C++ source files can be compiled in UNIX/LINUX environments to build a dynamic .so library and then used in UCM components. As HYPERSIM execution functions are generated in C by the simulator, additional wrappers functions are necessary to use EMTP C++ classes inside. Tests have been made, in real time, with the State Space code interfaced in HYPERSIM with this method. As an example, for a very simple network composed of a voltage source in series with a RL and connected to an UCM State Space with 40 poles, the execution time is about 6.5  $\mu$ s for a simulation running on a Linux Server composed of 12 X5690 Intel Xeon CPUs at 3.47 Ghz with a memory cache size of 12 Mo.

## IV. CONCLUSIONS

Accelerated models for EMTP-RV and HYPERSIM have been proposed in this paper. These developments were necessary to perform Real Time simulations including underground or submarine cables as well as large scale network models. These models bring also an improvement of accuracy which was required to analyze transients in Real Time simulations including Hardware in the Loop.

The performances of the accelerated models presented in this paper have never been achieved before in offline and real-time simulation tools. They are presently used in industrial application cases and more specifically HVDC interconnections and offshore wind farms connections.

## V. REFERENCES

- [1] Y. Vernay, B. Gustavsen, "Application of Frequency-Dependent Network Equivalents for EMTP Simulation of Transformer Inrush Current in Large Networks", International Conference on Power System Transients (IPST) conference, Vancouver, 2013.
- [2] Timothy A. Davis, "Direct Methods for Sparse Linear Systems (Fundamentals of Algorithms)" Society for Industrial and Applied Mathematics (SIAM) book.
- [3] [www.cise.ufl.edu/~davis](http://www.cise.ufl.edu/~davis) [www.cise.ufl.edu/research/sparse](http://www.cise.ufl.edu/research/sparse)
- [4] B. Gustavsen, A. Semlyen, "Rational approximation of frequency domain responses by vector fitting", Power Delivery, IEEE Transactions on, vol. 14, no.3, pp.1052-1061, Jul 1999.
- [5] O. Ramos, L. Naredo, J. Mahseredjian, C. Dufour, A. Gutierrez, I. Kocar, "A Wideband Line/Cable Model for Real-Time Simulations of Power System Transients", Power Delivery, IEEE Transactions on, vol. 27, no. 4, pp. 2211-2218, 2012.
- [6] S. Denetiere, A. Parisot, E. Milin and A. Dalmau Pons, "Resonance and insertion studies with EMTP: Working with large scale network models" presented at IPST2011 conference, Delft, The Netherlands, June 2011
- [7] J.R. Marti, "Accurate Modeling of Frequency-Dependent Transmission Lines in Electromagnetic Transient Simulations", IEEE Trans. On Power Apparatus and Systems, vol. PAS-101, pp. 147-157, 1982.