Implementation of a unified modelling between EMT tools for Network Studies

B. Bruned, C. Martin, S. Dennetière, Y. Vernay

***Abstract*—EMT offline and real-time simulation tools are often based on similar modeling approaches. Nevertheless some simplifications are implemented in real-time simulation tools to satisfy the real-time constraint. The common practice is to study a transient phenomenon with an offline tool and compare the analysis with real-time simulation connected to physical controls. In order to have a unified approach in the models for both EMT studies, two solutions are presented for control systems and electrical grid models.**

First, to enable inclusion of widely used Simulink libraries, a model interface feature has been implemented within two EMT software for offline and real-time studies. These Simulink-EMT interfaces achieve a unified modeling control approach ensuring the consistency on network studies with different EMT environments. Second, sharing among tools large networks via Frequency Dependent Network Equivalents (FDNE) is another approach to solve the modelling discrepancy. This solution based on state space formulation has been developed for real-time simulation. Test cases for real HVDC projects are presented to illustrate the benefits of proposed solutions.

***Keywords*: EMTP, Software development, Simulation techniques, Real Time Simulation, Control modelling, Network Equivalent**

I. Introduction

AT RTE (Réseau de Transport d'Electricité), the French TSO, several EMT tools are used for dedicated needs. Indeed, installations of electronic power equipment and converter substations for HVDC links require real-time simulation for maintenance activities such as replaying faults. This is done in the SMARte laboratory where control replicas of equipment directly connected to simulators are able to run scenarios on real-time [1]. Otherwise offline studies, which gain in precision without the time constraint, are dedicated to process deeper analysis to state the impact of these devices when they are connected to the network and anticipate issues. Two tools are presently used at RTE: EMTP (offline) [2] and HYPERSIM (real-time) [3]. These tools are mentioned as illustrative examples in this paper but the proposed approach is general. Both tools are based on similar resolution. However, result consistency is not guaranteed as simplifications are proceeded for real-time simulation. So transposing a complete study to a real-time environment is a complex and long task that requires experienced simulation engineers.

Therefore, model portability needs to be implemented for insuring unified modelling. Interfaces from standards are just getting to emerge such as the Common Information Model (CIM) [4]. This data exchange format enables data exchange between simulation programs. Current development work presented in [5] allows to import the whole 400kV and 225kV French grid into EMTP. However, only network static data are exported. Indeed, CIM enhancement for dynamic models [6] has not been yet implemented. Otherwise, the Functional Mockup Interface (FMI) offers an open and standardized method for model exchange and co-simulation between tools [7]. For the moment, this interface is not fully adapted to generate models that are optimized for real-time simulation. Nevertheless, it proposes a promising solution.

This paper presents two solutions to achieve model and data portability between offline and real-time tools. One is for control systems, the other is for electrical grid models. The first solution is based on Matlab-Simulink. Simulink provides a code generation method to export models [8] [9]. Specific model import has already been developed for each tool to integrate the generated code. In [10], these interfaces have already been used for getting a common Multilevel Voltage Source Converters modelling between EMT tools. We will focus more on the implementation techniques and make a cross-validation with the three tools (ie EMTP-RV, HYPERSIM and Matlab-Simulink).

The second solution is based on Frequency Dependent Network Equivalent model (FDNE). It can ensure consistency in network models especially for large network equivalents [11]. Accelerated in [12] with sparse algorithms, the State Space model has been implemented with specialized solution to meet real-time constraints. This paper presents solutions to optimize real-time performances and real test cases.

II. Implementation of a Simulink-EMT tools interface

A. Overview

Simulink is commonly used to build complex control models of power electronics equipment. Implementing an import method for EMT can unify control modelling among tools. A Simulink interface has been developed in both environments (offline and real-time). It uses exporting method based on code generation proposed by Simulink. The code generation can be customized through TLC (Target Language Compiler) scripts. Several options can be set such as the language of the code or the compilation setup. Therefore, each EMT tool has already their own external code interface. So customized TLC should re-arrange the model exporting code to fit these interfaces as a method to implement a Simulink-EMT interface. Both

B. Bruned, C. Martin, S. Dennetière, Y. Vernay are with RTE – Réseau de Transport d'Electricité. Tour Coeur Défense, 92932 PARIS La Défense, France (email of the corresponding author: boris.bruned@rte-france.com)

Paper submitted to the International Conference on Power Systems Transients (IPST2017) in Seoul, Republic of Korea June 26-29, 2017

techniques are described below.

For real-time simulation, code generation is proceeded on a host on which the Graphical User Interface (GUI) is executed. Then code is sent to a computer target to be compiled and executed during simulation. To incorporate external code, the User Coded Model (UCM) interface is used. A .def file allows the user to declare his models. During the network analysis on the host, the UCM code is generated from the definition file. Then it is linked and compiled with the simulation code on the target. The TLC script used for real-time simulation generates the UCM definition file (*model.def*) and refers to the Simulink model source along this file (*model.c, model.h…*).

For offline simulation, the solution is based on a Windows standalone application. Unlike the real-time simulation tool the offline simulation tool is not based on code generation. The simulation is run locally like the GUI. The simulation software reads the network data stored in an ASCII file and then executes the simulation. User defined code is integrated and compiled within a DLL which communicates at each time step with the main executable as a native model during the simulation [2]. Taking into account this interface, the TLC script used for offline simulation generates the DLL code. Visual Studio C++ compilers are used through Simulink code generation to build a locally ready-to-execute DLL. At the end of the code generation, only remains the DLL executable file (*model.dll*).

The Fig. 1 gives an overview of the whole automatic import process. The user action is required twice. First, he has to select the custom TLC script (EMTtool.tlc) and run the code generation within Simulink. The custom TLC generates the appropriate code interface. Second, the model is imported in the EMT tool by selecting the UCM file (real-time simulation) or the DLL file (offline simulation).

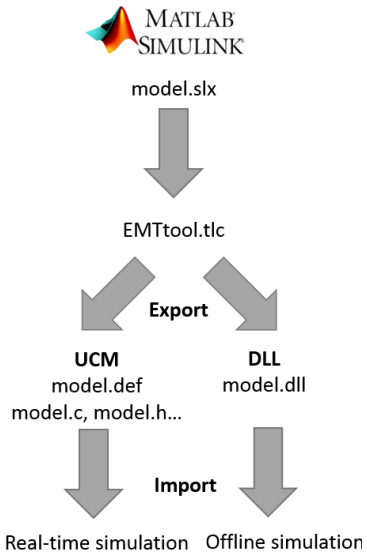

Fig. 1. Overview of the automatic import process from Simulink to EMT tools.

B. Main functionalities and limitations

A core of main functionalities shall be supported by the Simulink interface to benefit from the model export. However, some limitations remain from the code generation process.

A crucial topic is the multi-instance of the imported model. Any development of a Simulink-EMT interfacing shall be multi-instance oriented. A simple example to test multi-instance is presented in Fig. 2. This circuit uses two delays (UCM1 and UCM2) that have been generated from Simulink with same time delay (0.01s). A step signal is applied as input of both delays with a magnitude of 1 for UCM1 and 2 for UCM2. When there is no multi-instance, the same amplitude is observed as output of both delays because they share the same memory buffer. Both external code interfaces (UCM and DLL) solve this issue by allocating a buffer for each instance.

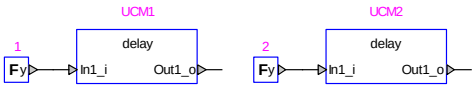

Fig. 2. Two delays test to check multi-instance.

Tunable parameter is a key functionality which shall be supported by both tools. It allows parameter value changes prior or during a simulation without having to re-process the whole import described in Fig. 1. However, the simulation time step cannot be tunable. This is the main limitation from the Simulink code generation method. As commercial grades EMT tools usually do not use variable time step solvers, the time step should not be tunable when the simulation is running. However, the possibility to change the value before the simulation is not possible as the value is hard coded in the generated code. A simple solution should be to generate the code N times for N different time steps. It is time consuming for the user and it increases the size of the generated code. Extrapolation methods can also approximate the simulation value when the Simulink time step is different from the EMT tool. It is the current solution used offline. Recently, a new method based on a time step value matching has been implemented and it is only suitable for an un-compiled code (real-time UCM solution). Only two code generations are required to first identify time step values and then make it tunable.

Portability among tools is supported through these interfaces sharing the same modelling. However, users may require model confidentiality while sharing an exported model from Simulink without revealing the whole logic. This is merely achieved with the offline interface where the code is directly compiled. Transmitting only the DLL insures model protection. The source code generated is hidden to the third party. For the real-time method, the model source (*model.c…)* shall be compiled as a static library (*modele.a*) to hide the whole model logic. The .def file contains only an external description of the model. However, this cannot be done with the local Windows compiler which is not compatible with the target computer that runs on Linux. To solve this issue a cross-compiler may be used. Nevertheless, only embedded architectures are supported within Simulink for proceeding cross-compilation locally. So for real-time mode, the static library has to be compiled after the Simulink code generation directly into the target prior the model import.

The previous functionality is a starting point to address the utility of using the Embedded Coder option [8] for the Simulink code generation. Simulink Coder [9] is a pre-requisite for code generation to export Simulink model. Embedded Coder

enhances this functionality proposing an optimized version of the code generation. Therefore, the whole integration process for real time simulation is facilitated from the code generation to the target execution. The Simulink-EMT interface (Fig. 1) is working with both solutions. However, as seen previously, real-time EMT targets are not supported among the easy-to-integrate architectures. Moreover, the gain in performance are not significant for the EMT real-time use. Actually, performance tests have been run on a complex generic control system of VSC MMC converter [13]. The Simulink model has been imported through the Simulink Coder and Embedded Coder solution and run on real time target (Linux Server composed of 48 E7-8837 Intel Xeon CPUs at 2.67 GHz). Small performance gains of 1.8% are observed between the two methods (8.07µs for Embedded Coder, 8.22µs for Simulink Coder). So for this specific EMT case, the Embedded Coder solution does not provide a clear benefit.

C. Validation example: POW switching command

In order to validate the accuracy of the Simulink-EMT interface, validation has been done on a simple case. It illustrates how to achieve a common modelling among three different tools (Simulink, EMTP and HYPERSIM). The test case is based on a Point-On-Wave (POW) circuit breaker command used to switch on and off shunt capacitors. This system is widely used in high voltage application and it is more specifically implemented in some Static Var Compensators (SVC) installed on the French grid to control Mechanical Switched Capacitor branch and avoid high transients during MSC switching. This POW control system has been modeled in Simulink and exported in EMTP, HYPERSIM and connected to the Simulink toolbox SimPowerSystem (SPS).

The test case is made up of a 225kV Thevenin equivalent voltage source connected to a 150MVAR shunt capacitor through the controlled circuit breaker. The synchronized command acts on each phase individually (cmd_A, cmd_B and cmd_C) and a delay of 60 ms is added to model the mechanical delay of circuit breaker. In a real case different mechanical delays shall be considered for the closing and opening sequences of the circuit breaker. The POW command uses the zero-crossing of phase A voltage (V_A) to synchronize the closing of the circuit breaker depending of the charge level of the capacitor. An algorithm to estimate the capacitor discharge is implemented in the POW command based on the last opening times of the breaker, the capacitance values and some internal design parameters of the system.

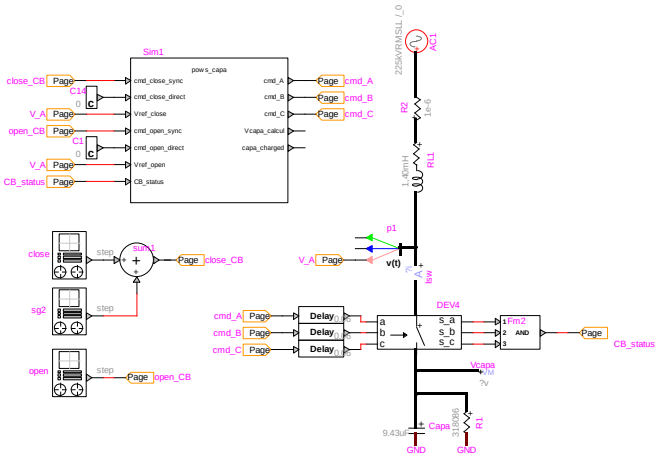

Fig. 3. Overview of test case for POW control.

In this example, the circuit breaker is first closed at 100ms with the capacitor fully discharged. At t=600ms, the circuit breaker is opened and re-closed partially discharged at t=2500ms. The losses of the capacitor are over-estimated in this example to meet the time simulation length. The TABLE I shows the time command for each phase (same times observed for the 3 tools).

TABLE I
INDIVIDUAL COMMAND TIMES GENERATED FOR CAPACITOR SWITCHING

Phase times	First closing	Opening	Second closing
t_A	115.15ms	615.15ms	2523.15ms
t_B	121.7ms	620ms	2529.85ms
t_C	118.35ms	621.7ms	2526.5ms

The phase A switch current and capacitor voltage of each tool are superposed in Fig. 4 below for validation. The synchronized command has effectively limited the current and voltage transients. The superposition of the three tools results is quite good (ε maximum of relative differences among tools). The same integration method with fixed type step and the same method to solve switch discontinuities have been used in the three simulation tools.

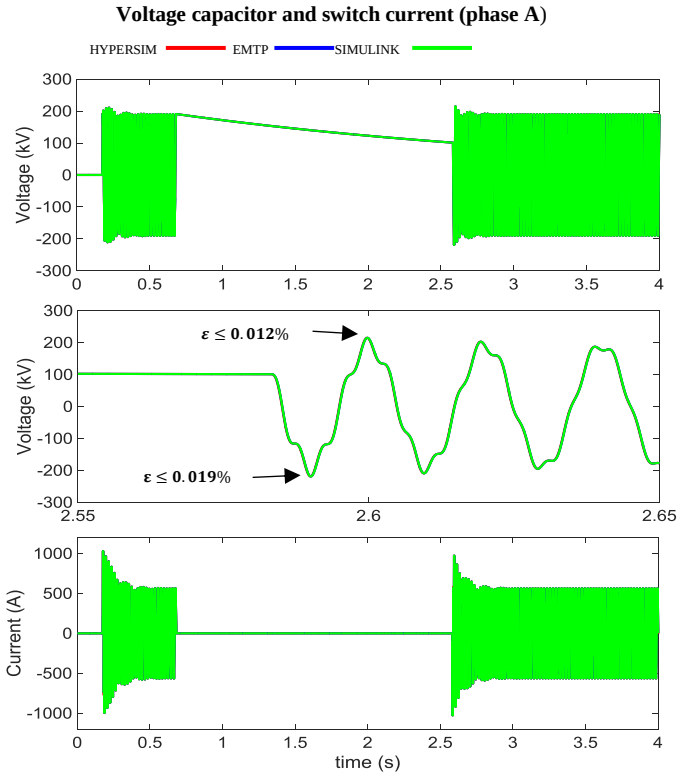

Fig. 4. Cross-validation results from the synchronized command of a capacitor.

III. REAL-TIME IMPLEMENTATION OF A STATE SPACE MODEL

A. Overview

Frequency Dependent Network Equivalents (FDNE) allow reduction of large power network keeping a good precision toward the original model. It improves effectively simulation performances. Also it can be a mean to exchange network data through its FDNE representation without revealing all details. The work in [11] has demonstrated the efficiency of such method to proceed offline EMT study. The concern here is to implement a real time State Space model which loads the FDNE data. A first try has been attempted in [12] for one example. However, the integration for other examples was not generic. Moreover, a native component can improve the performance avoiding multi-function calls which come from the DLL and UCM interface. The implementation focus is on the real time calculation. This is why all sparse algorithms described in [12] are not necessary useful while dealing with time constraint.

B. FDNE generation

The generation of a FDNE comes initially from a network frequency impedance measurement getting the matrices of the State Space representation as follow:

$$\begin{cases} \dot{x} = Ax + Bv \\ i = Cx + Dv + E\dot{v} \end{cases} \quad (1)$$

Where **i** is the vector of current, **x** the state vector and **v** is the voltage vector. Data inputs are the matrices **A**, **B**, **C**, **D** and **E**. The input file is generated from an internal tool developed by SINTEF using vector fitting techniques as described in [14]. As presented in [11] the following steps have to be processed:

- First, a frequency scan is run on EMTP to measure the network impedance in the frequency domain. For each frequency the impedance matrix is obtained.

$$Z(s) = \begin{bmatrix} Z_{aa}(s) & Z_{ab}(s) & Z_{ac}(s) \\ Z_{ba}(s) & Z_{bb}(s) & Z_{bc}(s) \\ Z_{ca}(s) & Z_{cb}(s) & Z_{cc}(s) \end{bmatrix} \quad (2)$$

- Then, a vector fitting is proceeded on the impedance matrices in the application [14].

$$Y(s) \cong \sum_{i=1}^{n} \frac{R_i}{s - a_i} + D + \mathrm{sE} \quad (3)$$

An equation rearrangement allows to get the state space matrices.

$$Y(s) = \mathrm{C(sI - A)^{-1}B + D + sE} \quad (4)$$

- Passivity enforcement may be necessary to run after the vector fitting insuring that a passive component is obtained [15].

C. State Space equations in time domain

The time domain equations (1) are discretized using the EMT tool integration method. Setting T as the time step and E as empty, we get for the trapezoidal rule of integration:

$$i(t) = G_{eq}v(t) + i_{hist}(t - T) \quad (5)$$

With G_{eq} the conductance matrix and i_{hist} the historical current.

$$G_{eq} = C\varphi\frac{T}{2}B + D \quad (6)$$

$$i_{hist}(t) = C\varphi\beta x(t) + C\varphi\frac{T}{2}Bv(t) \quad (7)$$

$$x(t) = \varphi\beta x(t - T) + \varphi\tfrac{T}{2}Bv(t - T) + \varphi\tfrac{T}{2}Bv(t) \quad (8)$$

$$\varphi = \left(1 - \frac{T}{2}A\right)^{-1} \quad (9)$$

$$\beta = 1 + \frac{T}{2}A \quad (10)$$

D. Real-time implementation

To figure out the model performance, calculations during the time step execution have to be isolated. The conductance matrix is calculated during the simulation preparation (no real-time constraint) when the code is generated. In real time simulation, the state space model is limited to the solution of equations (7) (8). Those calculations shall be done in a time step. The state space parameters have already been proceeded during the preparation (like $C\varphi\beta$, $C\varphi\frac{T}{2}B$ ...) . Only five matrix-vector multiplications have to be calculated. As noticed in [12] among them, four involve a sparse matrix: $\varphi\beta x(t - T)$, $\varphi\frac{T}{2}Bv(t - T)$, $\varphi\frac{T}{2}Bv(t)$, $C\varphi\beta x(t)$.

So a sparse matrix representation might be used to accelerate the computation. However, as no matrix inversion is done in real-time, there is no need to take a complex representation like in [12]. The triplet form of sparse matrix with the simple algorithm for matrix-vector multiplication is sufficient. Let's

take below a simple example to illustrate the triplet form for sparse matrices.

$$M = \begin{bmatrix} 0 & c & 0 \\ a & 0 & 0 \\ b & 0 & d \end{bmatrix} \tag{11a}$$

$$int\ i\ [] = \{0, \quad 1, \quad 2, \quad 2\} \tag{11b}$$

$$int\ j\ [] = \{1, \quad 0, \quad 0, \quad 2\} \tag{11c}$$

$$int\ p\ [] = \{c, \quad a, \quad b, \quad d\} \tag{11d}$$

Here**, i** and **j** are respectively the row and column indices for the non-null elements of M which are contained in **x**. The pseudo code for the matrix-vector multiplication $y = Mx + y$ is:

$$\begin{aligned} &\text{for } n \in i \text{ do} \\ &\quad \text{for } m \in j \text{ and } p_{nm} \neq 0 \text{ do} \\ &\quad\quad y_i = y_i + p_{nm} x_j \end{aligned} \tag{12}$$

E. Validation and Performance

A cross-validation has been done with real-time and offline tool. The test case is a FDNE reduction of a small 225kV network as presented in Fig. 5.

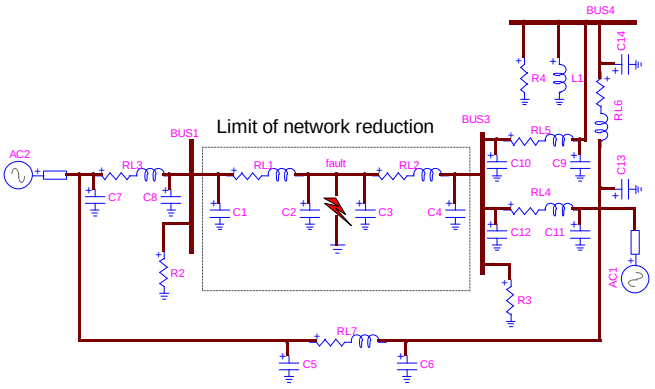

Fig. 5. Simple network for FDNE reduction test case (HYPERSIM view).

The network equivalent has two terminals BUS1 and BUS3. Only the two central PI-lines will be kept. A 10 Ohm single phase fault (phase A) is applied between these PI-lines at t=30ms and is eliminated at t=60ms. This scenario is used for both networks (reduced one and complete one). The network reduction has been done within EMTP to get the 50 poles FDNE data file. The Fig. 6 below shows the reduced network.

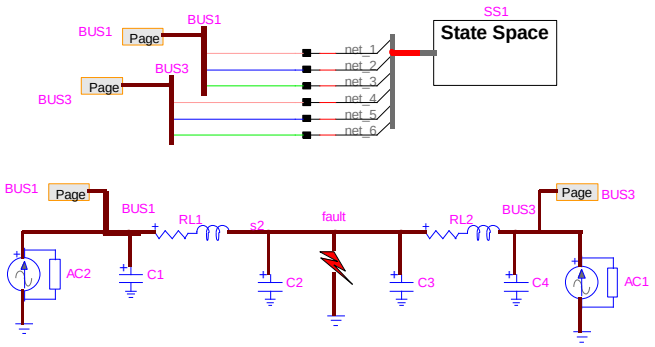

Fig. 6. Reduced Network (HYPERSIM view).

Voltage sources have been replaced by a Norton equivalent to cope with the FDNE modelling which is considered as an impedance. Simulation results of the fault current and voltage node are superposed for the complete network (HYP2) and the reduced one (HYP1 and EMT1).

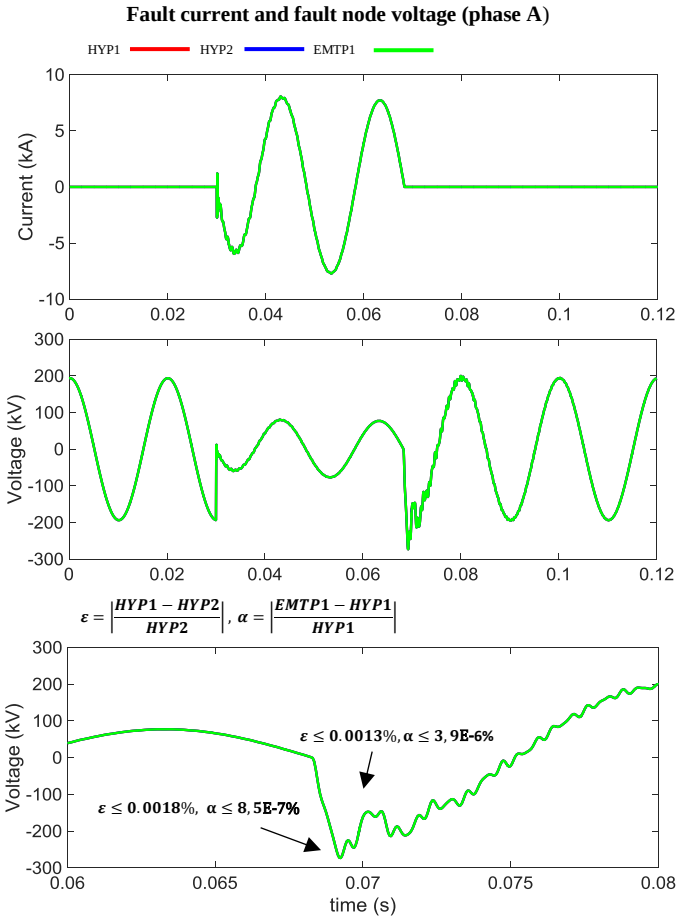

Fig. 7. Results superposition for reduced and real network.

The superposition of the results is quite good. It validates the State Space implementation in real time toward offline model (ε relative difference) and also it demonstrates the FDNE relevance where precision is equally the same as the real network (α relative difference). In a second step, the FDNE data file has been replaced by a 50 poles network equivalent of the whole 225kV French grid with two terminals (two substations located in French Brittany region, Tregueux and Rance). The results of both tools are well superposed too in Fig. 8.

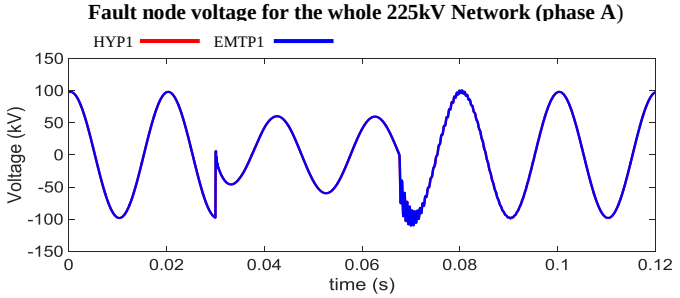

Fig. 8. Results for the French network 225kV FDNE.

Real-time performances have been studied to analyze the impact of the sparse matrix representation. The simulation has

been run on a Linux target computer composed of 48 E7-8837 Intel Xeon CPUs at 2.67 GHz for the previous case (Fig. 6 reduced model) with a time step of 50µs. In TABLE II, the number of poles has been modified to state the impact on performance.

TABLE II
CPU TIME COMPARISON FOR TIME DOMAIN RESOLUTION

Number of poles	State Space model	State Space with sparse algorithm
25	42.5µs	4.1µs
50	168µs	7.4µs
100	2290µs	11.8µs
150	7679µs	16.1µs

We can notice that without the sparse algorithm the State Space model fails to run on real-time. The complexity of a State Space relies on the number of poles and the number of terminals whatever the reduced network is complex or not. The number of poles is related to the frequency range and precision of the vector fitting. In this case, as seen previously, satisfactory results in terms of accuracy are achieved with only 50 poles.

Another practical application of the FDNE is the long cable modelling. Frequency dependent cable models used in EMT tools usually take into account propagation effect as the Wideband model [12]. However, propagation modeling limits the simulation time step value which shall not exceed the minimum of propagation delays. When each individual section of a cable is modeled (to get an accurate representation of the overall cable system with each grounding point), the required simulation time step can be very small. This constraint can be quite problematic for real-time simulation. The FDNE is a good option to solve this issue keeping the model precision. This is illustrated by the HVDC Savoy-Piedmont interconnection project between France and Italy [16]. The 58 sections of the 2*190km long DC cables have been modeled with individual WideBand models in the offline tool EMTP-RV as illustrated in Fig. 9. The length of the sections imposes a simulation time step of 10µs that cannot be achieved in real-time. The VSC converter stations are represented with detailed MMC models.

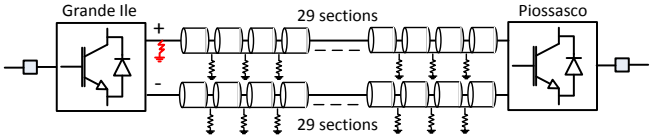

Fig. 9. Overview of the Savoy-Piedmont test case in EMTP (reference case).

The Simulink interface described in section II. is used in offline and real-time tools in order to have the same control VSC system model in both simulation tools. The HVDC link is simulated in real-time with the DC cables represented with a 180 poles 4x4-port FDNE equivalent. A 20µs time step is set for the real-time simulation. In Fig. 10 Stub lines are used to decouple the solution of each VSC converter with the FDNE device in order to run the case on 3 CPUs (1 per VSC converter 1 for the FDNE).

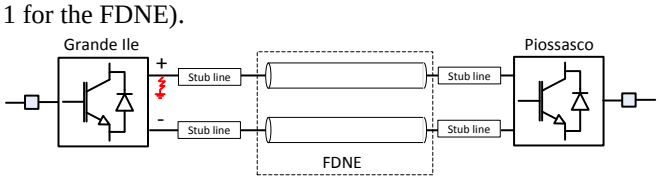

Fig. 10. Overview of the Savoy-Piedmont test case in HYPERSIM

The real-time simulation results (HYP) are compared against the HVDC link modeled with the 58 WB cables sections (EMTP). Pole to ground voltage on negative pole is presented in Fig. 11. Slight differences are noticeable due to the FDNE reduction.

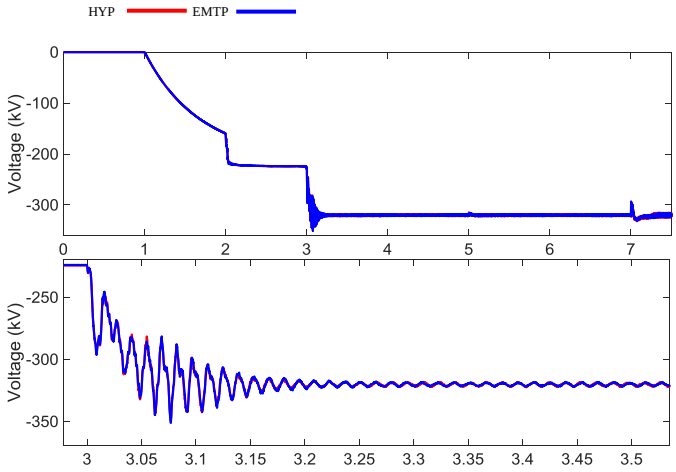

Fig. 11. Pole to ground voltage during starting sequence.

A pole to ground fault is simulated at t=8s on the positive pole. Pole to ground voltage of the healthy pole is presented in Fig. 12. Protection systems (converter block and AC circuit breaker trip) and surge arresters in converter stations are modeled as described in [17].

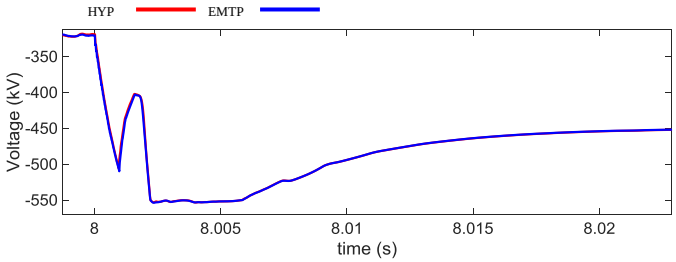

Fig. 12. Pole to ground voltage during pole to ground fault.

This result superposition validates the relevance of the FDNE method for reducing the detailed cable model for this specific case. It is not a generic approach and it shall be used with care. The physical propagation delays are not accurately represented with this approach. In this case, a good precision is kept for transient analysis and the HVDC link model is runnable on real-time with a higher 20µs time step.

IV. CONCLUSIONS

This paper has demonstrated that a unified modelling is possible for EMT tools for control and power equipment modeling. A Simulink-EMT interface is essential for running control systems in offline in real-time tools with the same accuracy. It meets a real requirement for the HDVC interconnection projects where real time controls model can be used in conjunction with the physical control replicas. Furthermore, the State Space model for real time simulation has been presented. Above all, it allows complex networks exchange between tools without having to redefine it completely.

These works are a step forward in a convergence process between EMT offline and real-time tools.

V. Appendix

Data of the test case presented in Fig. 5:

- RL load on BUS3 R=480 Ohm, L=3.06H
- Coupling PI-Line between AC1 and AC2 voltage source R=17.5 Ohm, L= 351 mH, C=189 µF.
- Other PI-Lines R=1.75 Ohm, L= 35.1 mH, C=189 µF.
- BUS1 and BUS3 resistance R=1 MOhm.
- The two Thevenin equivalent voltage sources (AC1 and AC2) are initialized from Load Flow. Constraints are a swing bus on AC1 which set the voltage as 240kV RMS
- (225 kV is the reference voltage). A PQ constraint is set for AC2 voltage source P=-200 MW and Q=-10 MVar. The source RL values are R=4 Ohm, L=70 mH.
- Single phase fault applied at t=30 ms and eliminated at t=60 ms
- Fault resistance value R=10 Ohm.

Then, for the reduced network (Fig. 6) the two Norton equivalents have also been initialized from Load Flow. Two voltage constraints (swing bus) have been set on BUS1 and BUS3 taking the whole network load flow values (Fig. 5).

VI. References

[1] S. Dennetière, O. Saad, A. El-Akoum, X. Legrand, A. Xémard, J. Mahseredjian, H. Motoyama, "Involvement of electric utilities in the development of EMT simulation tools", Cigre Paris Session, France, 2014.

[2] J. Mahseredjian, S. Dennetière, L. Dubé, B. Khodabakhchian and L. Gérin-Lajoie: "On a new approach for the simulation of transients in power systems", Electric Power Systems Research, Volume 77, Issue 11, September 2007, pp. 1514-1520.

[3] V. Q. Do, J.-C. Soumagne, G. Sybille, G. Turmel, P. Giroux, G.Cloutier, S. Poulin. "Hypersim, an Integrated Real-Time Simulator for Power Networks and Control Systems", ICDS'99, Vasteras, Sweden, May 25-28, 1999.

[4] E. Lambert, X. Yang, and X. Legrand, "Is CIM suitable for deriving a portable data format for simulation tools?", IEEE PES General Meeting, July 2011, pp. 1–5.

[5] S Dennetière, Y. Vernay, C. Martin, D. Petesch, H. Saad, "AC grids and HVDC systems modelling coherency between EMT and phasor domain tools", Cigre Paris Session , France, 2016.

[6] D. Becker, T.L Saxton, and M. Goodrich, "CIM Standard for Dynamic Model Exchange", IEEE PES General Meeting, July 2010, pp. 1–3.

[7] T. Blochwitz *et al.*, "The Functional Mockup Interface for Tool independent Exchange of Simulation Models", Proceedings of the 8th International Modelica Conference, Mar. 2011.

[8] Mathworks, Automatic Code Generation - Embedded Coder [online]. Available: http://www.mathworks.com/products/embedded-coder.

[9] Mathworks, Automatic Code Generation - Simulink Coder [online]. Available: http://www.mathworks.com/products/simulink-coder.

[10] P. Le-Huy, S. Casoria, O. Saad, "Unified Modeling and Simulation Approach for Modular Multilevel Voltage Source Converters", International Conference on Power System Transients (IPST) conference, Vancouver, Canada, 2015.

[11] 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, Canada, 2013.

[12] B. Clerc, C. Martin, S.Dennetière "Implementation of accelerated models for EMT tools", International Conference on Power System Transients (IPST) conference, Cavtat, Croatia, 2015.

[13] H. Saad *et al.*, "Dynamic Averaged and Simplified Models for MMC-Based HVDC Transmission Systems", IEEE Trans. Power Delivery, vol. 28, no. 3, pp. 1723–1730, July 2013.

[14] B. Gustavsen and A. Semlyen, "Rational approximation of frequency domain responses by vector fitting", IEEE Trans. Power Delivery, vol. 14, no. 3, pp. 1052-1061, July 1999.

[15] B. Gustavsen and A. Semlyen, "Enforcing passivity for admittance matrices approximated by rational functions", IEEE Trans. Power Systems, vol. 16, pp. 97-104, Feb. 2001.

[16] Piedmont – Savoy HVDC Link [online]. Available: http://www.think-grid.org/piedmont-savoy-hvdc-link.

[17] S. Dennetière, H. Saad, B. Clerc, "Setup and performances of the real-time simulation platform connected to the INELFE control system", International Conference on Power System Transients (IPST) conference, Cavtat, Croatia, 2015.