

A Backward-Forward Interpolation Technique for a Precise Modelling of Power Electronics in HYPERSIM

Van-Qué Do¹, David McCallum¹, Pierre Giroux¹, Bruno De Kelper²

1) Institut de Recherche d'Hydro-Québec (IREQ)
1800 blvd. Lionel Boulet
Varenes (Québec), Canada, J3X 1S1
do.van_que@ireq.ca

2) École de Technologie Supérieure
1100 rue Notre-Dame ouest
Montréal (Québec) Canada, H3C 1K3
bruno@ele.etsmtl.ca

Abstract - Hypersim is a real-time numerical simulator which enables the simulation of power systems and their controls. Many of today FACTS controls generate firing pulses for power electronic switches. Simulation of these devices requires the refactorization of the admittance matrix due to the change of switch status. This paper describes the formulation of the node equation, the reordering technique to reduce the refactorization effort and the backward-forward interpolation technique to simulate correctly the switching phenomena. The algorithm requires only two interpolations but reduces considerably unwanted oscillation created by the jittering between the switching moment and the simulation time-steps. Tests are done on an AC-DC network and a PWM circuit. Results show good improvement with a minor increase in computation time.

Keywords - power electronic, jitter effect, linear interpolation, sparse matrix reordering, refactorization.

1. INTRODUCTION

HYPERSIM is a real-time digital simulator which enables the simulation of power system networks together with their controls. Details of the simulator have been described in [1]. Real-time is achieved by using the time-synchronized parallel processing in which each participating processor performs some tasks representing parts of a network. All processors run at a same constant time-step.

Power system studies on the real-time simulators involve more and more power electronic devices such as thyristor, GTO, IGBT, which are turned on and off by control systems. In the past, Hypersim had already the facility to simulate these devices [2] but considered that switchings happen exactly at time-steps. Unfortunately, they generally occur between two time-steps. This creates an error on the switching moment (translated into firing or extinguishing angle) which varies from time-step to time-step. The final result is a non-characteristic oscillation because the measured value can almost never reach the desired setting.

This problem is well known in the digital simulation of electronic switching devices. Previously published papers have suggested various correction algorithms. Gole and al [6] used a technique to specifically overcome the thyristor extinguishing problem and to improve the measurement of

firing angle. De Kelper and al. [7] use a more precise technique but it requires recalculation of the admittance matrix at the switching point, although this is simplified somewhat to gain the calculation speed.

Real-time simulation needs an efficient algorithm to solve the nodal equation and take into account the fact that the switchings generally happen between 2 time-steps. To achieve this, the algorithm described in this paper uses an ordering technique to minimize operations in the refactorization and a backward-forward interpolation to accurately simulate the timed switching. The algorithm has been implemented in Hypersim to work in real-time. Tests shown good results with a little increase in calculation effort.

2. SOLUTION OF NODE EQUATION

Equation formulation

Hypersim simulates large power systems in real-time by taking the advantage of the natural delay caused by transmission lines. Substations can therefore be simulated in parallel. At the end of each time-step, simulation results are exchanged to be used in the next time-step.

Transmission lines are modelled as equivalents at each end which are composed of current sources and parallel resistors. Each sub-station is therefore reduced to a network composed of RLC elements (resistor, inductor, capacitor) and sources.

Hypersim uses the same nodal approach as in EMTP to simulate power system networks. Using the trapezoidal integration technique, each inductor or capacitor is equivalent to a resistor in parallel to a current source reflecting its history (past values) of current or voltage. Voltage sources such as generators are converted into current sources while current sources are kept as is. Switches are represented as resistors with very small value when they are closed and very large value when opened.

In overall, each sub-station is composed solely of resistors and current sources. The purpose of the simulation is to solve for node voltages using the following equation:

$$\mathbf{YV} = \mathbf{I} \quad (1)$$

where \mathbf{Y} is the admittance matrix including all resistors in the substation, \mathbf{V} is the node voltage vector to be found and \mathbf{I} represents all current sources including real sources as

well as history of L and C elements.

In power systems, the admittance matrix \mathbf{Y} is normally very sparse. This should be taken into account in order to minimize the calculation effort in the solution for node voltages.

LU solution

With the LU (lower-upper) conversion of \mathbf{Y} , equation (1) becomes

$$\mathbf{Y}_L \mathbf{Y}_U \mathbf{V} = \mathbf{I} \quad (2)$$

or

$$\mathbf{Y}_U \mathbf{V} = \mathbf{Z}_L \mathbf{I} \quad (3)$$

where \mathbf{Y}_U contains only non-zero terms on and above the diagonal. $\mathbf{Z}_L = \mathbf{Y}_L^{-1}$ and has non-zero terms on and below the diagonal. To facilitate the backward substitution, the LU conversion is normalized such that \mathbf{Y}_U has 1.0 on its diagonal.

To obtain (3) from (1), one starts with

$$\mathbf{YV} = \mathbf{UI} \quad (4)$$

where \mathbf{U} is unity matrix, then for each row starting from the top, we nullify terms on the left of the diagonal. To nullify the k th element of the i th row:

$$\text{new} \mathbf{r}_i(\mathbf{Y}) = \mathbf{r}_i(\mathbf{Y}) - Y_{ik} \mathbf{r}_k(\mathbf{Y}) \quad (5)$$

$$\text{new} \mathbf{r}_i(\mathbf{U}) = \mathbf{r}_i(\mathbf{U}) - Y_{ik} \mathbf{r}_k(\mathbf{U}) \quad (6)$$

where $\mathbf{r}_i(\mathbf{Y})$ mean row i of \mathbf{Y} and *new* stands for the new iteration. By applying (5) and (6) to eliminate all terms in \mathbf{Y} on the left of its diagonal, \mathbf{Y} will be transformed to \mathbf{Y}_U and \mathbf{U} to \mathbf{Z}_L . Finally, \mathbf{V} is found using the backward substitution, starting from the bottom row.

3. REORDERING OF ADMITTANCE MATRIX

As the matrix \mathbf{Y} changes at every switching and at every change in segment of a non-linear element, it needs to be refactorized (LU conversion must be redone). This is a time-consuming operation which has to be optimized. The reordering described here is done prior to the simulation in order to minimize the time spent for the refactorization.

Reordering for switches and non-linear elements

In (5) and (6), $k < i$ as shown in Fig. 1, it means that the cancellation of terms on the left of the diagonal at a given row depends only on rows above it. Therefore, if one element in \mathbf{Y} is changed (either by a switch operation or by a change in segment of a non-linear element), all rows below it should be recalculated. This suggests to put all rows corresponding to nodes where switches and non-linear elements are connected at the bottom of \mathbf{Y} in order to minimize the calculations needed for updating of \mathbf{Y}_U and \mathbf{Z}_L in the refactorization process.

Therefore, \mathbf{Y} is first reordered in two sections: rows with fixed elements on top and rows with variable elements (switches and non-linear elements) at the bottom.

Reordering to reduce fill-in elements

Another reason for reordering \mathbf{Y} is that during the calculation of (5), some zero elements on $\mathbf{r}_i(\mathbf{Y})$ can become non-zero elements due to a non-zero element found on row k as shown in Fig.1, this is called fill-in element. The more there are fill-in elements on the left of the diagonal of \mathbf{Y} , the more additional calculation is needed to eliminate them. The more there are fill-in elements on the right, the more effort is needed to update them.

The transformation of \mathbf{U} into \mathbf{Z}_L by (6) shows also that terms on row i of \mathbf{U} become non-zero if $Y_{ik} \neq 0$. In order to reduce the computation time, \mathbf{Z}_L must contains the fewest possible non-zero terms.

Therefore \mathbf{Y} should be ordered in such a way as to minimize the fill-in elements in \mathbf{Y}_U and the non-zero terms in \mathbf{Z}_L . The reordering scheme used in Hypersim is similar in principle to Gomez's scheme [4, 5] but with some additions:

- Reorder rows of \mathbf{Y} from top to bottom in the increasing order of number of non-zero terms,
- For rows with the same number of non-zero terms, in the upper half of the matrix, order them in the decreasing order of the distance between the diagonal and the first non-zero term; in the lower half, order them in the increasing order of this distance.

Step b) is the main difference between the Hypersim technique and the one described in [4] and [5]. It has the effect of delaying fill-in values and consequently reduces further fill-in values to rows below them. For example, in Fig. 1, non-zero term Y_{km} creates fill-in values for rows starting from $m+1$ and below while Y_{kn} , ($n > m$), creates fill-in values for rows from $n+1$ which is less than the previous case. The step b) helps to further decrease the number of non-zero terms in \mathbf{Y}_U and \mathbf{Z}_L . This is illustrated in Table 1 showing the percentage of non-zero elements after the refactorization for two cases: substation of 50 and 100 nodes, and for three situations: no reordering, reordering without step b) and reordering with step b).

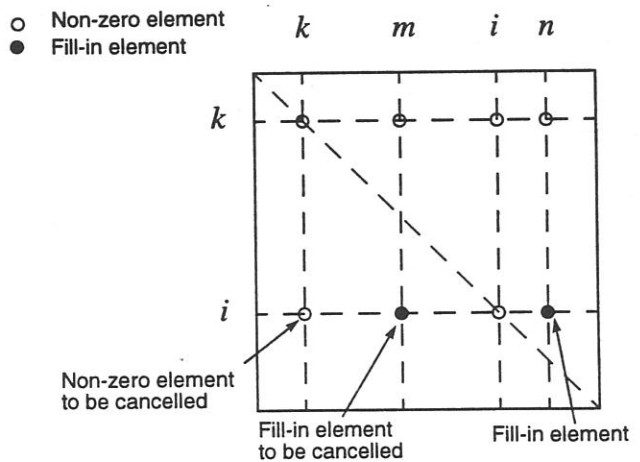


Fig. 1 Calculation of \mathbf{Y}_U by cancelling non-zero terms on the left of the diagonal of \mathbf{Y} .

Table 1: Percentage of non-zero elements in \mathbf{Y}_U and \mathbf{Z}_L after the refactorization

nb. of nodes		No reordering	Reord. wo step b	Reord. with step b
50	\mathbf{Y}_U	26.01	15.11	14.81
	\mathbf{Z}_L	43.84	32.63	31.37
100	\mathbf{Y}_U	35.46	24.55	23.60
	\mathbf{Z}_L	48.35	43.17	42.50

Steps a) and b) are done separately for fixed and variables elements to ensure that variable elements stay at the bottom rows.

4. BACKWARD-FORWARD INTERPOLATION

Principle

The principle of the backward-forward interpolation technique used in Hypersim is illustrated in Fig. 2. In this figure the switching (point 4) happens between the present time-step (point 5) and the previous one (point 3). In the other hand, the simulation must solve for node voltages at every time-steps and not between them. Therefore, voltages are first calculated for the point between time-steps to respect the switching instance and then interpolated to get values at the time-step.

The admittance matrix \mathbf{Y} has generally elements whose values depend on the time increment T between two consecutive time-steps. If T changes, a complete refactorization is needed. During the refactorization of (1) to get (3), if T is not changed, the upper rows of \mathbf{Y}_U and \mathbf{Z}_L correspond to fixed elements and therefore need not to be recalculated. To maintain this advantage, we will keep T constant in the following procedure:

- To get solution at the switching point (point 4), one need the history at T before it (point 2). History at point 2 are found by a backward interpolation of history of points 1 and 3. The interpolated history is then used to build the

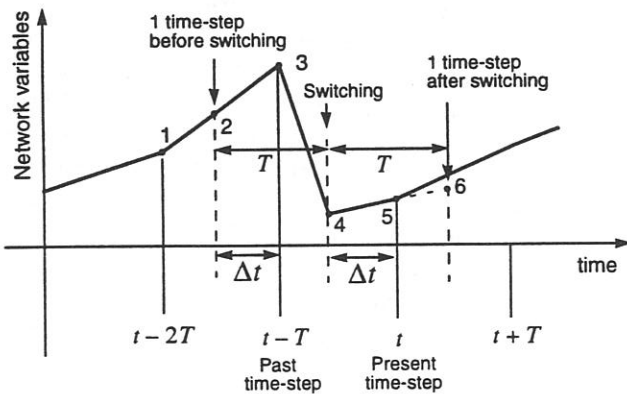


Fig. 2 Backward-forward interpolation principle

current vector \mathbf{I} .

- At point 4, the switching happens. To get the solution immediately after the switching, rows of \mathbf{Y}_U and \mathbf{Z}_L starting from nodes corresponding to switched element down to the bottom need to be recalculated because of the changing in admittance of the switched element.
- With history found in a) and newly refactorized matrices \mathbf{Y}_U and \mathbf{Z}_L in b), equation (3) can then be solved to get nodes voltages at the switching point 4.
- To get the solution for the present time-step (point 5), a forward interpolation is performed: Using the same \mathbf{Y}_U and \mathbf{Z}_L already refactorized in step b) and history at the switching point, go one time-step forward and solve for point 6. Finally, the solution of point 5 is then found by the linear interpolation of point 4 and 6.

Overall, the backward-forward interpolation technique involves one partial refactorization in step b) and two interpolations in steps a) and d). Comparing to the non interpolated method, only interpolation effort has been added and this is not a major burden.

Algorithm

Equation (3) rewritten in more complete form is

$$\mathbf{Y}_{U,n} \mathbf{V}_n = \mathbf{Z}_{L,n} \mathbf{I}_n (\mathbf{V}_{src,n}, \mathbf{I}_{src,n}, \mathbf{H}_k) \quad (7)$$

where subscript n and k indicate the point number in Fig. 2 (1 to 6), k is the point of the previous step with respect to n , current vector \mathbf{I} is a function of history \mathbf{H} , of voltage sources \mathbf{V}_{src} and of current sources \mathbf{I}_{src} .

The backward linear-interpolation calculates history at point 2 as follows:

$$\mathbf{H}_2 = \mathbf{H}_3 - D(\mathbf{H}_3 - \mathbf{H}_1), \quad (8)$$

$$D = \Delta t / T \quad (9)$$

T is the time increment between two time-step and Δt is the delay between the switching point and the present time-step (between 4 and 5). For a forced-commutation, the switching moment is determined by the control system. For a natural commutation such as the extinguishing of a thyristor which happens when current crosses its holding level, an extrapolation using two previous values of the current determines the switching-off moment. The effort to do this extrapolation is not significant because it is applied only on current waveforms rather than on matrix's elements.

From (7) and (8), voltages at point 4 are given by

$$\mathbf{Y}_{U,4} \mathbf{V}_4 = \mathbf{Z}_{L,4} \mathbf{I}_4 (\mathbf{V}_{src,4}, \mathbf{I}_{src,4}, \mathbf{H}_2) \quad (10)$$

$\mathbf{Y}_{U,4}$ and $\mathbf{Z}_{L,4}$ are partially refactorized (only rows starting from node corresponding to switched elements are recalculated) to take into account changes in admittances caused by the switching occurred at point 4. Solution at point 4 determines also history at that point which is then used to solve for point 6:

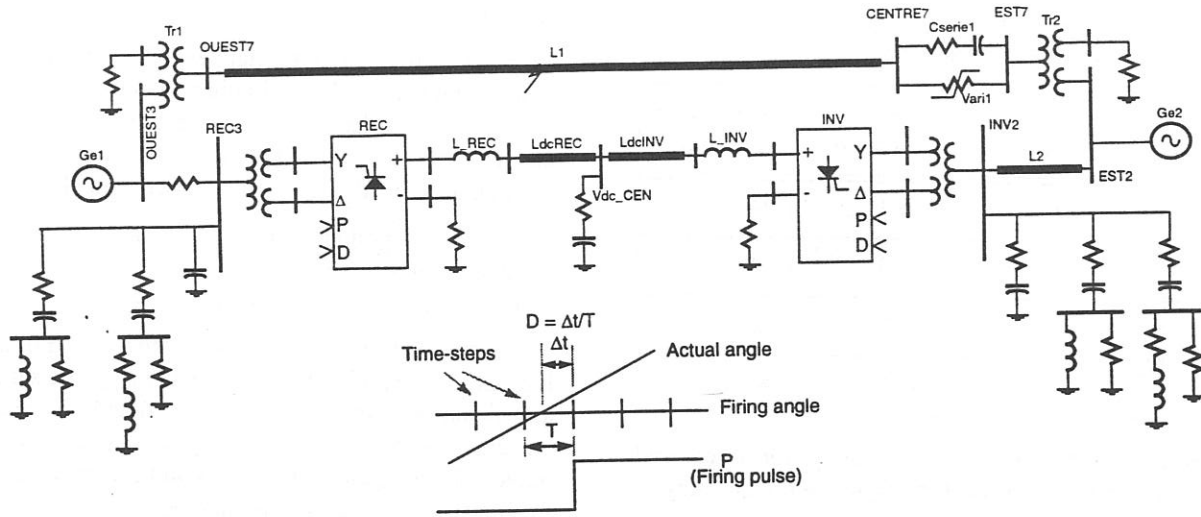


Fig. 3 AC-DC interconnected network used to test the interpolation technique. Inset diagram shows also the principle of generating the firing pulse P and the delay signal D .

$$Y_{U,4} V_6 = Z_{L,4} I_6(V_{src,6}, I_{src,6}, H_4) \quad (11)$$

Here, the same $Y_{U,4}$ and $Z_{L,4}$ as in (10) are used, they need not to be refactorized because there is no switching from 4 to 6.

Finally, the node voltages and history for the present time-step (point 5) are found by a second linear-interpolation:

$$V_5 = V_6 - D(V_6 - V_4) \quad (12)$$

$$H_5 = H_6 - D(H_6 - H_4) \quad (13)$$

In (10) and (11), V_{src} and I_{src} at point 5 (the present time-step) are used as approximated values for points 4 and 6 because they are normally slow comparing to switching phenomena. Our experiences have shown that the additional interpolation of V_{src} and I_{src} have only minor effect and it is therefore not necessary to spend this extra effort.

5. APPLICATIONS

AC-DC interconnected network

The network used to test the interpolated thyristor model is shown in Fig. 3 which is an AC-DC interconnected network.

The AC network is composed of two equivalent networks interconnected by an AC line with series compensation protected by ZnO surge arrester. AC filters are also connected at both terminals.

The DC network includes a rectifier and an inverter interconnected by a DC line and a DC cable (frequency dependent model). The converter controllers used are of generic type [3]. Controllers generate the necessary firing pulses P for thyristor as well as the delay signal D to be used in the interpolation process. When external controls are used, the rectifier REC and the inverter INV are simply two twelve-pulses bridges which are triggered by external

connectors P and D . These signals can be provided by user's control units built either with the integrated Hypersim control blocks or with Simulink.

Steady-state waveforms obtained from this network are shown in Fig. 4. The network is simulated with a step of $50\mu s$ (about 1 electrical degree in a 60 Hz system). Without the interpolation, a non-characteristic oscillation is found on all waveforms. With the backward-forward interpolation, the oscillations become imperceptible.

Without the interpolation, the firing pulses P are always delayed while with interpolation, this delay is compensated. This explains the differences of about one degree in firing angles between two cases.

PWM circuit

PWM applications are normally more demanding for a good modelling of switching devices because of the high modulating frequency. The PWM circuit used to test the efficiency of the interpolation algorithm is shown in Fig. 5 which is a DC-DC converter. The GTO firing is controlled by the PWM module, when the GTO conducts, the Diode is biased in reverse and is opened. When the GTO is extinguished, the Diode starts to conduct providing a path for current I_{Zs} . Consequently, I_{Zs} will flow continuously.

The PWM module compares a triangular waveform (carrier) to a constant signal of 0.7 (modulated signal) to produce a pulse train of 70% duty cycle. Without interpolation, the duty cycle is not constant because the turn-on and off times are not exactly at the simulation time-steps. The PWM bloc outputs an additional signal D indicating the delay from the switching instant to the actual time-step. Hypersim uses this output to perform the necessary interpolation.

The theoretical DC value of voltage at *Bus1* is $0.7V_1$. The DC component of I_{Zs} is therefore

$$I_{dc,Zs} = \frac{(0.7V_1 - V_2)}{1\text{ohm}} \quad (14)$$

which should be 10A with the given parameters. The instantaneous values of I_{Zs} are shown in Fig. 5c and d. The irregularity of the upper and lower peaks on these two waveforms are due to the asynchronism between the data

sampling frequency and the PWM carrier's frequency. The DC components of I_{Zs} are calculated using a window-averaging and results are shown in Fig. 5b. One can see that without the interpolation, there is an important oscillation

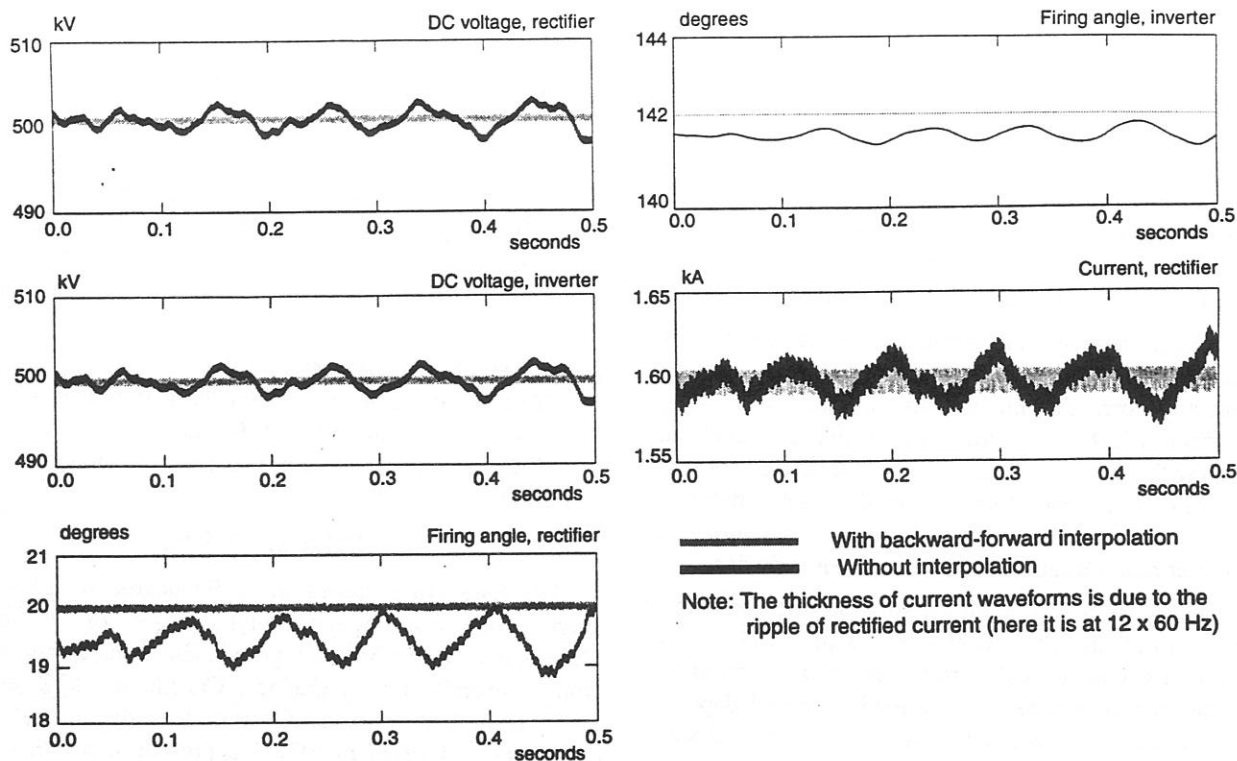


Fig. 4 Comparison of waveforms obtained from the network shown in Fig. 3 simulated at 50μs time-step.

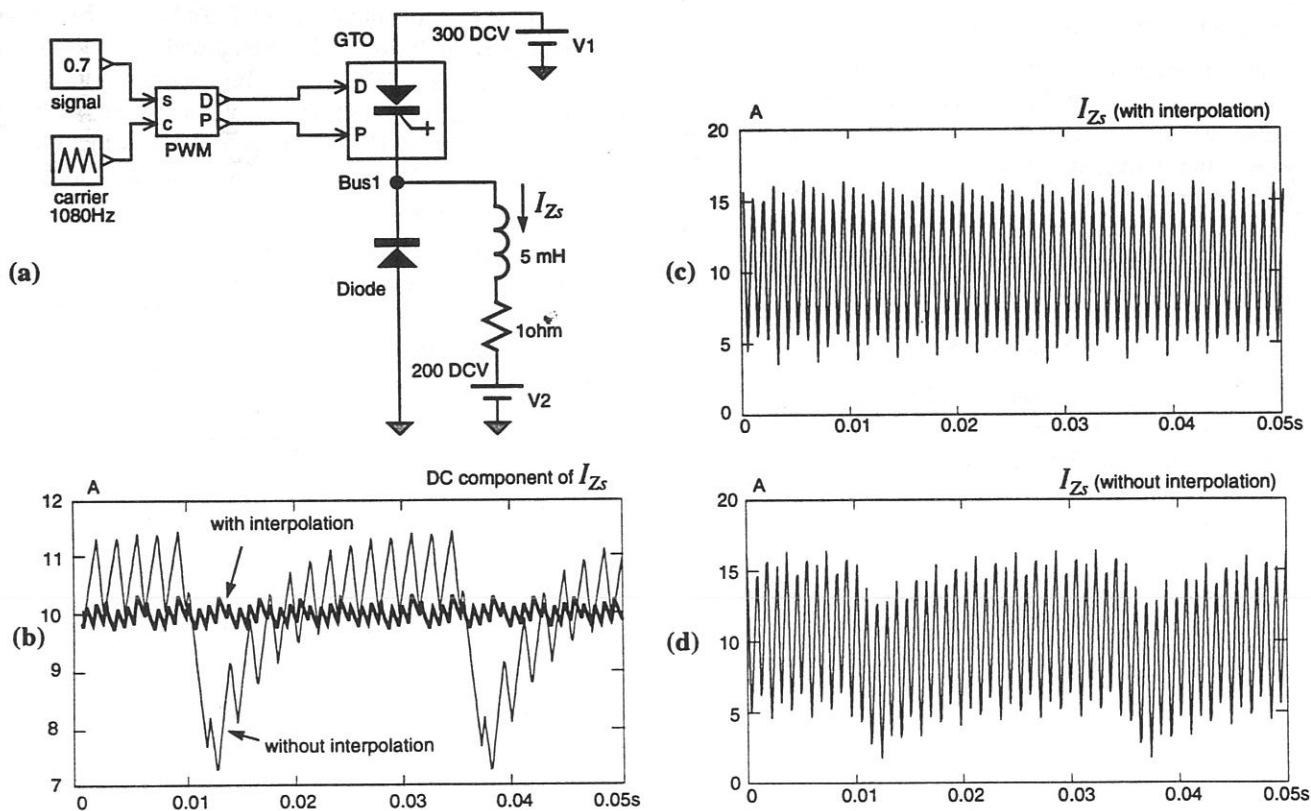


Fig. 5 Application of the interpolation technique to a PWM circuit used as a DC-DC converter

due to the jittering between the switching of GTO and the simulation steps. This oscillation is considerably reduced when the interpolated valve model is used.

6. CONCLUSIONS

This paper has presented an algorithm implemented in Hypersim to efficiently solve node equations involving non-linear and switching elements. Reordering the \mathbf{Y} matrix to put rows corresponding to these elements at the bottom reduces the number of operations performed at each admittance change. Further reordering of \mathbf{Y} has also reduced the number of fill-in elements and consequently reduced the time required for refactorization.

Switching elements are normally turned on and off asynchronously with respect to the calculation steps. Without any corrective method, this causes an annoying jitter effect which translates into oscillations and can interact with control systems. The backward-forward interpolation technique described in this paper helps to considerably reduce this erroneous effect and allows a much better representation of power electronic devices.

Compared to the conventional solution, the backward-forward interpolation technique adds only two interpolations but no extra refactorization is required. Computation time increases by around 25% and depends on the number of switches and non-linear elements involved. Furthermore, with a 50 μ s of time-step and with the interpolation technique, oscillation caused by jitter is less than with a 5 μ s step. The reasonably small increase in computation time is therefore largely compensated by the improvement in the simulation accuracy.

The interpolation technique has however some drawbacks: It supposes that waveforms are almost linear between two points. But if switchings have occurred in between, the interpolation is not good anymore. This happens particularly in cases involving three-phase PWM circuits and with a high modulating frequency. More works need to be done to generalize this technique.

REFERENCES

- [1] V.Q. Do, J.C. Soumagne, G. Sybille, G. Turmel, P. Giroux, G. Cloutier, S. Poulin, "Hypersim, An Integrated Real-Time Simulator for Power Network And Control Systems," *Third International Conference On Digital Power System Simulators (ICDS's99)*, Vasteras, Sweden, May 25-28, 1999
- [2] C. Larose, V.Q. Do, G. Sybille, F. Guay, J.C. Soumagne, "A PWM GTO-Inverter Model for the Hypersim Digital Power System Simulator," *Third International Conference On Digital Power System Simulators (ICDS's99)*, Vasteras, Sweden, May 25-28, 1999
- [3] Y. Maharsi, V.Q. Do, V.K. Sood, S. Casoria, J. Bélanger, "HVDC Control System Based on Parallel Digital Signal Processors," *IEEE Transaction on*

Power Systems, vol. 10, no. 2, May 1995, pp. 995-1002.

- [4] A. Gomez, L.G. Franquelo, "Node Ordering for Sparse Vector Method Improvement," *IEEE Transactions on Power System*, vol. 3, no. 1, February 1988, pp. 73-79.
- [5] A. Gomez, L.G. Franquelo, "An Efficient Ordering Algorithm to Improve Sparse Vector Methods," *IEEE Transactions on Power System*, vol. 3, no. 4, November 1988, pp. 1538-1544.
- [6] A.M. Gole, L.T. Fernando, G.D. Irwin, O.B. Nayak, "Modeling of Power Electronic Apparatus: Additional Interpolation Issues," *IPST'97 - International Conference on Power System Transients*, Seattle, June 22-26, 1997
- [7] B. De Kelper, L.A. Dessaint, V.Q. Do, J.C. Soumagne, "Algorithm for Accurate Switching Representation in Fixed Step Simulation of Power Electronics," *2000 IEEE PES Winter Meetings*, Singapore, January 23-27, 2000

BIOGRAPHIES

Van-Qué Do obtained his B.Sc degree in Electrical Engineering from École Polytechnique of Montréal, Canada in 1970, an M.Sc. degree in Electrical Engineering from Université Laval, Québec, Canada in 1972, and a Ph.D. from Université du Québec, Canada in 1986. He joined Hydro-Québec in 1972. His present interest is in the application of parallel computing to power systems simulation. He is a senior member of the IEEE and a member of the Ordre des Ingénieurs du Québec.

David McCallum obtained his B.Sc.Eng in Electrical Engineering from Imperial College, London in 1966. He worked initially for GEC Power Engineering and joined Hydro-Québec in 1980. He has been responsible for many simulation studies on HVDC projects. He is a member of IEE, a member of IEEE and a member of the Ordre des Ingénieurs du Québec.

Pierre Giroux obtained his B.Sc. degree in Electrical Engineering from École Polytechnique of Montréal in 1976. He joined the Institute de recherche d'Hydro-Québec (IREQ) in 1988. He has coordinated and realized several simulator studies related to SVS, HVDC and FACTS controls. His current interest is focused on the design and evaluation of real-time control systems used in the power quality domaine. He is a member of Ordre des ingénieurs du Québec.

Bruno De Kelper received his B.Eng in 1993 from École de Technologie supérieure of Montréal and his M.Sc.A in 1998 from École de Technologie supérieure. He is currently preparing his Ph.D. in electrical engineering at École de Technologie supérieure. He has been working with GREPCI, a research group in power electronic and digital control, for the past 5 years. His research interests are real-time simulation of large power electronic system and parallel computing.