A Fully Interpolated Controls Library for Electromagnetic Transients Simulation of Power Electronic Systems

A.M. Gole, S.A. Woodford,
University of Manitoba

J.E. Nordstrom, G.D. Irwin
Manitoba HVDC Research Centre

Winnipeg, Canada

Abstract - **Interpolation is a means for accurately modelling the exact switching instant in transient simulation programs without using a small timestep. The paper describes the construction of a number of control components which perform such interpolation using PSCAD/EMTDC as the simulation platform. A user can connect these together to design an arbitrarily complex controller of his or her choice. The developed library includes comparators, logic gates, flip-flops and other typical control blocks necessary for power electronics systems. A pulse width modulated (PWM) inverter system is used to demonstrate the approach.**

Keywords: Transient Simulation, Interpolation, Power Electronics, Pulse Width Modulation. Mixed Mode Simulation

I. INTRODUCTION

Power Electronic Circuits present a special problem for electromagnetic transients simulation. A typical power electronic switch such as a thyristor or GTO may open and close several times in a cycle. Finite time-step emtp-type programs usually allow such switchings to occur on integral multiples of the time-step. However, this can cause various problems such as spurious inductive current chopping and additional jitter in the values of various measured quantities. On the other hand, the finite time-step approach usually results in faster simulation compared to programs such as SPICE which can use a variable timestep. The interpolation approach allows for an exact representation of the switching event, without having to use a smaller timestep. This is achieved by linearly interpolating all system quantities to the time of switching from the calculated values at the timesteps bracketing the switching instant. This approach was originally introduced in the Netomac Program [1], and has been shown by several authors to produce highly accurate simulations [1,2].

The incorporation of interpolation into the solution of an electromagnetic transients simulation program is usually carried out internally, i.e., access to this feature is not readily available to arbitrary user developed control blocks. The proliferation of power-electronic subsystems in today's electric industry has meant that vendors of emtp-type software are not able to provide typical inner-level control loops for every conceivable circuit topology. In order to allow full flexibility to the user, a set of primitive control building blocks is often present in several emtp-type programs. However, these blocks have generally not allowed the use of interpolation at every level, i.e., in elements such as logic gates and memory registers.

The fully-interpolated controls library developed here introduces the interpolation feature in all blocks- be they analog or logic devices. The user can connect these together to design a controller of his or her choice and be assured of accuracy at every point in the path of the control signal. The paper describes the implementation of such a library for the PSCAD/EMTDC simulation environment which allows for digital as well as analog (mixed-mode) control blocks. The method of introducing interpolation in comparators, logic gates, flip-flops and other typical control blocks is described.

The efficacy of the approach is evaluated with simple example cases such as a Pulse-width-modulated (PWM) induction motor drive.

II. FIXED TIMESTEPS AND INTERPOLATION

A. Problems with Fixed Timestep Simulations

Emtp-type programs typically use a fixed time-step for the numerical solution of the power system's differential equations. The trapezoidal rule is used to convert the differential equations into algebraic equations that allow for the calculation of the network voltages at the end of a time-step from a knowledge of the sources and the initial values at the beginning of the time-step[3]. For most HVDC/FACTS type studies a value of Δt in the range 25 μs-50 μs is considered acceptable.

With semiconductor switches such as thyristors, GTO devices and so on, the switch can only be turned on or off at an integral multiple of this basic time-step. A 50 μs time-step corresponds to typically 1^o of a fundamental frequency waveform. This results in a number of drawbacks:

1. In the case of a thyristor or diode turning off, the current zero may occur in between two points of the discrete time grid. If the series element is inductive, spurious voltage spikes may arise. This is one reason why artificially large snubber circuits across the device may have to be modelled [4].
2. The 1^o (approximately) jitter in firing angle leads to the generation of spurious non-characteristic harmonics. This is particularly serious if there is a network

resonance in the neighborhood of these harmonic frequencies [2].

3. The error may be compounded if several cascaded control or measurement blocks (i.e., extinction angle measurement) each have the 1^o or so of uncertainty. Kuffel et. al.[5] report a total jitter of up to 5^o while using a 50 μs time-step.
4. Certain simulation studies, such as modulation of dc for Sub-synchronous Resonance (SSR) damping require a modulation signal to the firing angle with an amplitude of fractions of a degree. The 50 μs time-step is too coarse [2].

B. Methods for Precise Modelling of Switching Instants

HVDC and Static Var Compensation schemes have a firing angle jitter specified to be about 0.1^o. It is desirable that the simulation program at least reproduce this accuracy. The 0.1^o requirement translates into a time-step in the order of 5μs or smaller. However, a time-step this small could lead to a prohibitively large computation time.

One way around this is to use a variable time-step [6] so that if a switching instant is detected, the program changes to a small time-step and reverts to the original time-step. However this requires a re-triangularization of the matrix and in general results in longer CPU times as compared to the interpolation method used in the current program.

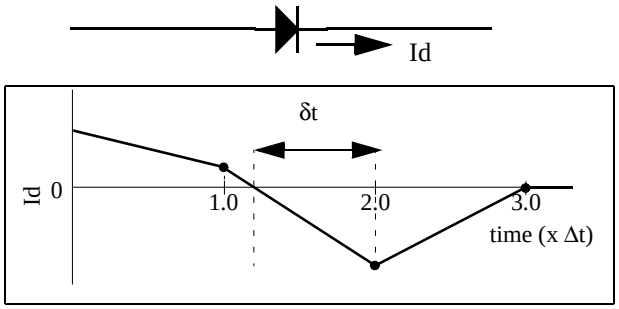

Fig. 1. Simulated diode switching off at a current zero - fixed time-step

An alternative is to use linear interpolation [1,2,5]. This method can best be described with the example of a simple diode. The waveform in Fig. 1 shows the current through a diode with a standard fixed time-step switching algorithm. The current reverses at some time in between Δt and 2Δt, but because of the discrete nature of the time-step, the impedance of the device can only be made infinite (i.e., diode turns off) at integrals of Δt. Here the simulated diode actually turns of at time 2Δt, whereas in reality the process should have occurred dt time units earlier. The first recorded instant of zero current is thus at 3Δt.

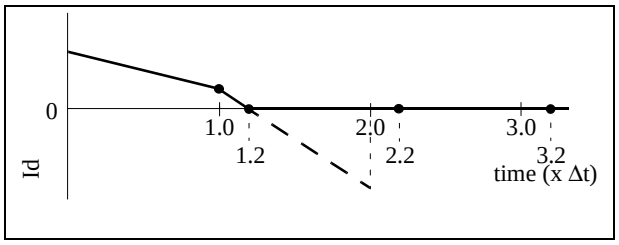

Fig. 2. Simulated diode switching off at a current zero - with interpolation

Fig. 2 shows the same device with the switching interpolated to the correct instant. As before, the program calculates the solution at t = Δt and t = 2Δt. However, on noticing that at the latter time, the current has already crossed zero, it estimates the turn-off time to be t =1.2 Δt based on a linear interpolation of the current within the switching interval. All the voltages and currents in the trapezoidal solution method are then also interpolated to this intermediate time in a linear fashion. The admittance matrix is then re-formulated and the solution continues with the original time-step, yielding the new solution one time-step later at t = 2.2 Δt. One additional interpolation step between t = 1.2 Δt and 2.2 Δt yields the solution at t = 2 Δt. The latter apparently cosmetic step is taken to put the solution back on the original time grid.

Similarly, interpolation can be introduced in the generation of the firing pulses for power electronic circuits as shown in Fig. 3. The firing angle (α) is defined as the time (converted to units of angle by multiplying with the system's angular frequency) from the zero crossing of the forward biasing voltage to the time at which the firing pulse is issued. The firing pulse is generated for example, by the intersection of a timing ramp with the firing angle order signal. The firing pulse forwarded to the emtp-type algorithm now consists of two parts - the pulse itself which can only be synchronous with a timestep and the interpolation correction δt_2 which we refer to as the "timestamp".

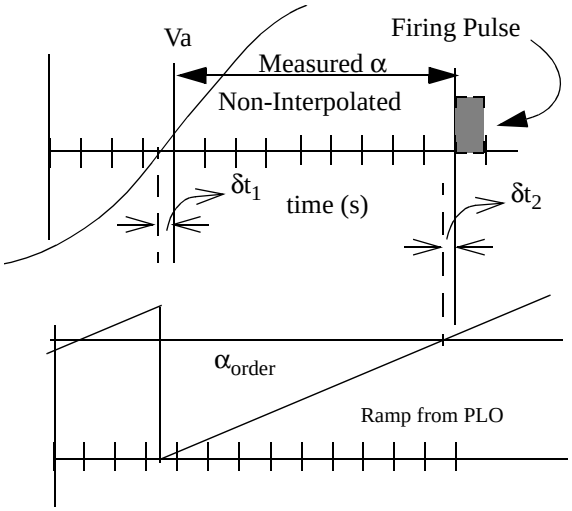

Fig. 3. Firing angle measurement with and without interpolation

The additional correction δt_2 is then passed to the emtp-type algorithm so that the above described changes to the solution process can be made.

Similarly, linear interpolation allows for a much more accurate measurement of α by using the additional correction timestamp δt_1 in Fig. 3.

Although interpolation requires a few additional computations in the algorithm, it has been shown that this only introduces a small computation time overhead[2,5].

The results obtained from interpolation using a large time-step match those obtained with using a very small time-step [2]. Thus using the above approach it is possible to get very precise switching times without recourse to a small time-step and the associated large CPU time.

III. INCORPORATING INTERPOLATION IN LOGIC AND CONTROL BLOCKS

Most programs supporing the interpolation feature use interpolation internal to the algorithm. For example, a firing angle order in degrees or radians could be supplied by the user and implemented internally in the program using the procedure described in the previous section. However, it is a more complicated to include interpolation in an arbitrary user-developed control system, particularly one using logical components. This is important when the user is attempting to model a firing control system for an HVDC or FACTS converter as it would be implemented in the real world where such components are used.

A. Incorporating Interpolation into Logic Gates

The process for incorporating interpolation into logic components is described by considering the following examples of modelling AND and OR gates. It should be noted from the discussion in section IIB that for accurate firing of the valve, two signals are necessary - firstly the pulse itself which by necessity must be synchronous with a timestep and secondly the "timestamp" δt. Any implementation of a logic gate should propagate both these signals properly.

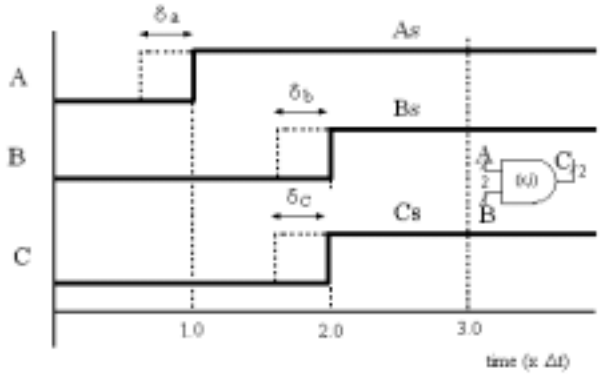

Fig. 4. Implementation of the AND Gate

Fig. 3 illustrates an example situation, in which a 2-input, interpolated AND gate logic function encounters an output transition during a simulation run. As shown,

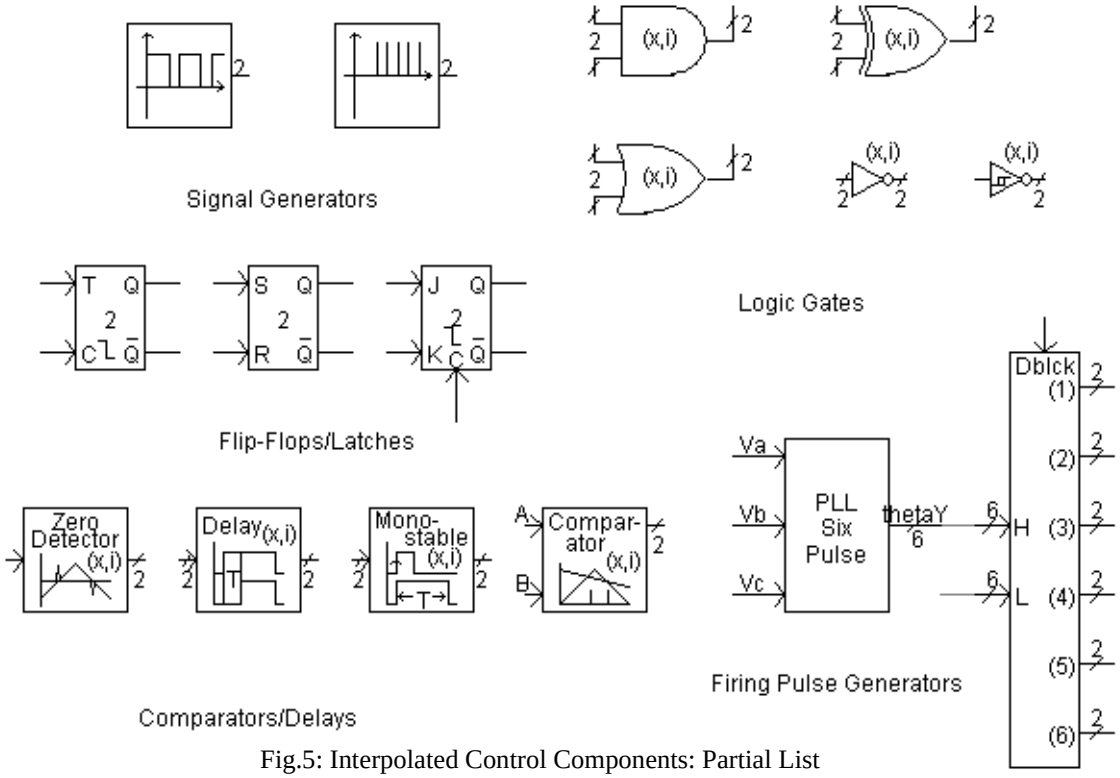

Fig.5: Interpolated Control Components: Partial List

logic input A transitions from 0 to 1 at time $\Delta t - \delta_a$, whereas logic input B transitions from 0 to 1 at time $2\Delta t - \delta_b$. If A and B were input to an interpolated AND gate, the output C would transition from logic 0 to logic 1 at $2\Delta t - \delta_b$. Therefore, the interpolation information passed to the output is $\delta_c = \delta_b$. If both inputs A and B were to transition during the same timestep, then the input with the smallest interpolated value would be passed to the output C.

In an interpolated OR gate, the output C would transition from logic 0 to logic 1 at $\Delta t - \delta_a$, so then the interpolation information passed to the output is $\delta_c = \delta_a$. Bold lines As and Bs represent the output values (the first part of the signal) as would be seen on a plot which would of course be synchronous with the timestep. Similar considerations apply to the timestamp for the 1 to 0 transition.

Thus the interpolated gate models include the above rules for propogating the timestamp in addition to generating the output pulse.

B. Incorporating Interpolation into other Control Blocks

Interpolation can be introduced into other control blocks in a similar manner. For instance, a comparator component will detect when two signals cross one another. Incorporating an interpolated time here is a simple matter of generating an interpolated value as described in the previous section with reference to the firing pulse generation (Fig. 3).

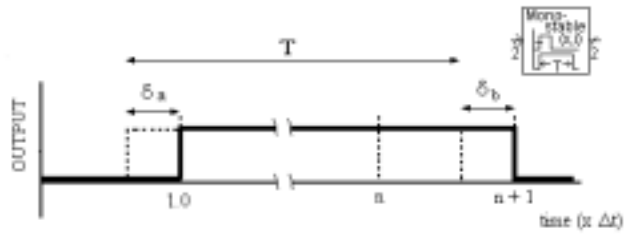

Fig. 6. Implementing the Monostable

An interpolated monostable multi-vibrator block must consider an interpolated input, along with a user defined time period. The block must now provide two interpolation timestamps - δ_a associated with the trigger pulse and another correction δ_b,at the end of the monostable pulse as seen from Fig. 6.

Using this approach several components have been designed within the PSCAD/EMTDC interface. These include logic gates, firing pulse generators, comparators, timing circuits, signal generators and resetable integrators. A partial list of these components is shown in Fig. 5. The numeral two at the inputs or outputs is to indicate that two pieces of information are being propagated - the signal itself as in the conventional simulation and the interpolation timestamp.

IV. SIMULATION EXAMPLE

The efficacy of the interpolation technique is demonstrated using the simple Pulse Width Modulated Voltage Sourced Inverter shown in Fig. 7 (taken from a screen capture of the PSCAD/EMTDC simulation palette). The converter consists of six switching modules, each consisting of a back to back IGBT (insulated gate bipolar transistor) switch and diode. An input - side LC filter is provided to smooth the current going to the dc supply. The load is a delta connected RLC load. The firing pulses are generated by comparing a high-frequency triangle 'carrier' wave with the desired sinusoidal 'signal' waveform. The switch in the upper bridge arm is turned on when the triangle wave is below the signal and a lower arm switch is turned on when the triangle wave is larger. This ensures a reduced harmonic content in the load current. The control circuit includes interpolated components, such as the waveform generator and comparators as well as logic gates.

Typical load voltage and current waveforms are shown in Fig. 8 where a modulation index (ratio of carrier to signal frequencies) of 15 is used. A closer look at the waveform of phase current is available in Fig. 9 which shows the impact of various timestep and interpolation options.

The results can be timestep dependent if interpolation is not used as seen from Fig. 9a obtained with a 50 μs timestep with and without interpolation. As can be seen, there is a difference in the two waveforms. When a much smaller timestep of 5μs is used, the results are virtually identical to those obtained with the much larger timestep of 50 μs using interpolation.

A similar result is obtained from a plot of the spectrum of the load current. From Fig. 10a, without interpolation, using a 50 μs timestep results in a fair amount of low order non-characteristic harmonics, in addition to the expected harmonics of order 5, 7, 11, 13 and so on. When interpolation is used, these non - characteristic harmonics are no longer present and the effect is that of using a much smaller timestep of 5 μs as seen in Fig. 10b. These spurious non-characteristic harmonics can cause even further inaccuracies if there are resonances in the network that can be excited at these frequencies.

Thus it is evident that the accuracy of the simulation can be significantly improved when interpolation is used.

V. CONCLUSIONS

The method of interpolation is extended to include arbitrary logic and control components. Selecting these new control elements from the graphical simulation palette allows the user to model complex firing circuits for power

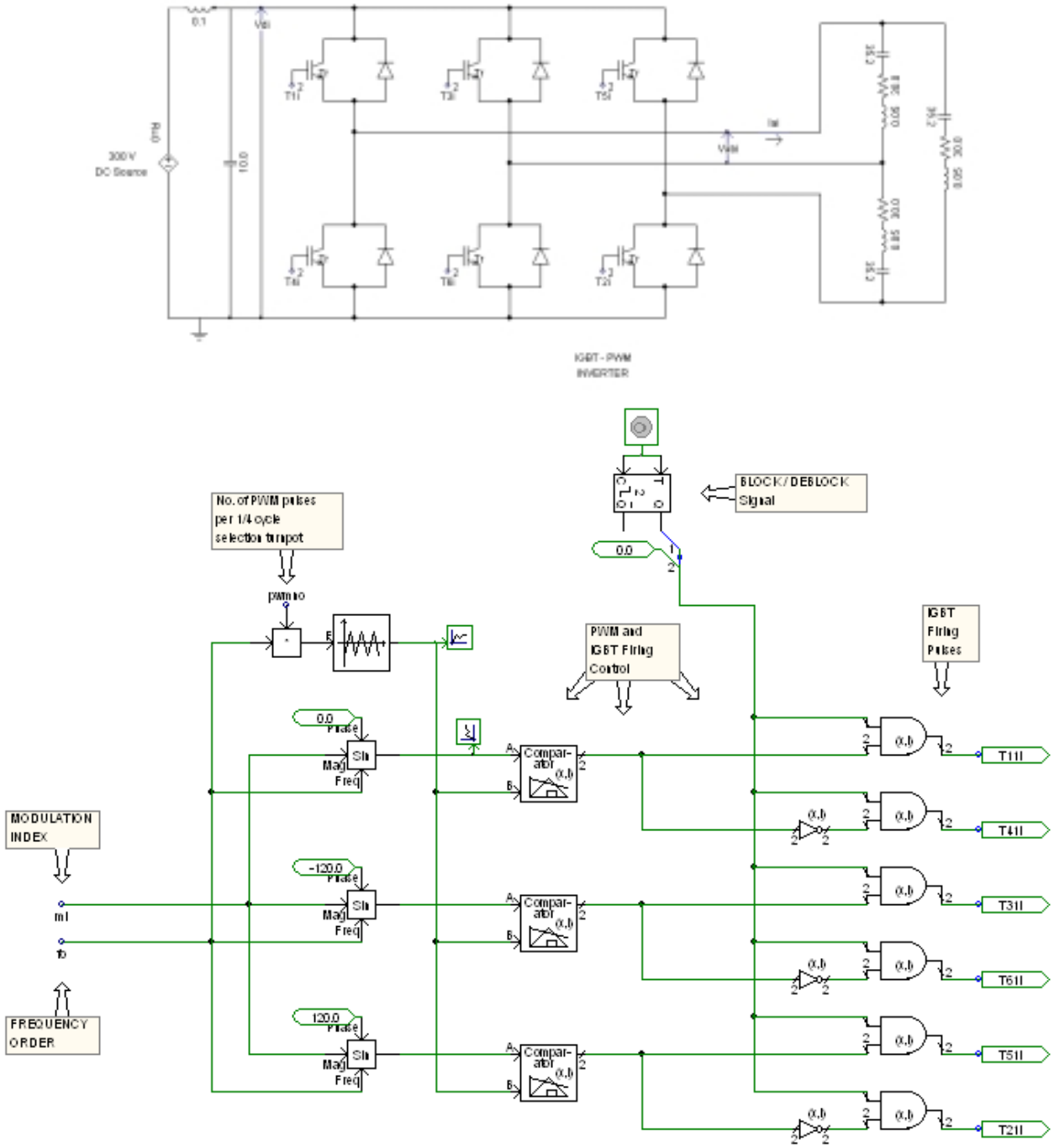

Fig.7: Pulse Width Modulated Inverter and Control Circuit

electronic equipment. The solution accuracy is maintained without recourse to a much smaller timestep. The additional penalty from using the interpolation method is minimal. The simulated example of a PWM inverter clearly shows this improved accuracy, both in the actual waveform as well as in its harmonic spectrum.

VI. REFERENCES

[1] K.H. Krueger, R.H. Lasseter, “HVDC Simulation Using NETOMAC”, IEEE Montech ‘86, Conference on HVDC Power Transmission, Montreal, Canada 1986.

[2] A.M. Gole ,I.T. Fernando, G.D. Irwin , O.B. Nayak, “Modeling of Power Electronic Apparatus: Additional Interpolation Issues”, International Conference on

Power System Transients, IPST'97, Seattle, June 22-26, 1997, Proceedings: pp 23-28

[3] Dommel, H.W., "Digital Computer Solution of Electromagnetic Transients in Single- and Multi-Phase Networks", IEEE Trans. PAS, Vol. PAS-88, No. 4, April 1969, pp 388-399

[4] Lefebvre S., and Gerin-Lajoie L. , "Static Compensator Model for the EMTP", IEEE Transactions on Power Systems, Vol. 7, No. 2, May 1992, pp 477-486.

[5] P. Kuffel, K. Kent and G.D. Irwin, "The Implementation and Effectiveness of Linear Interpolation Within Digital Simulation", International Conference on Power System Transients, Lisbon, Portugal, September 3-5, 1995, Proceedings, pp 499-504.

[6] Gole A.M. and Sood V.K., "A Static Compensator Model for Use with Electromagnetic Transients Simulation Programs", IEEE Trans. on Power Delivery, Vol. 5, No.3, July 1990, pp 1398-1407

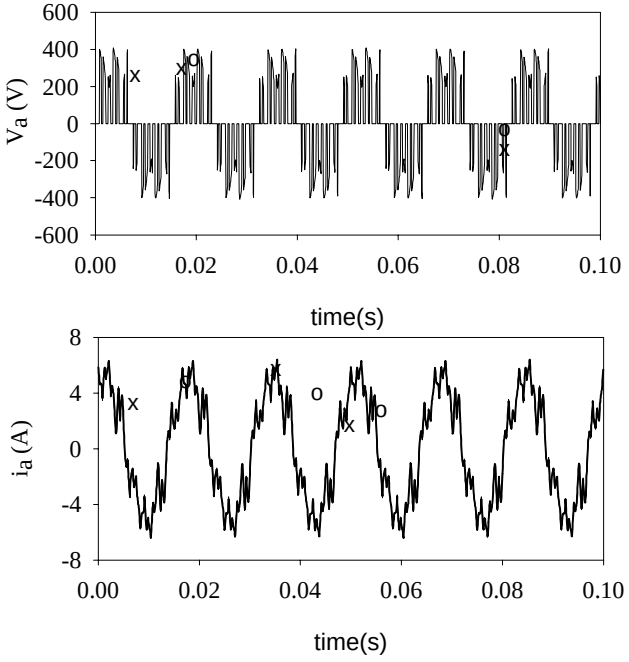

Fig. 8. Steady State Waveforms for PWM Inverter

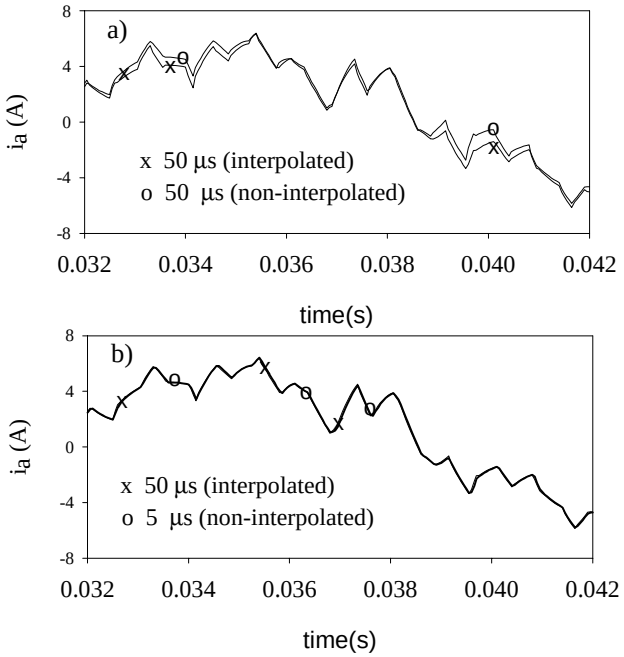

Fig. 9. Interpolated / Non-interpolated Current Waveform

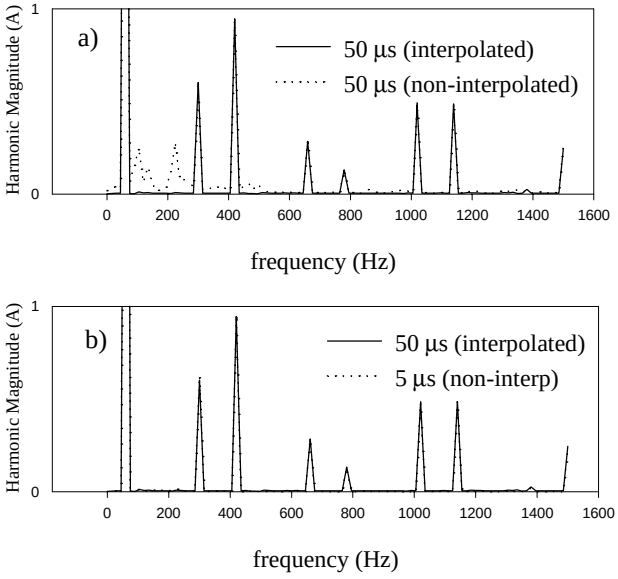

Fig. 10. Comparison of Harmonics