Coupling Digital Controllers with Real-Time Digital Simulators for Switched Power Systems

V. R. Dinavahi[†] M. R. Iravani[‡]

Center for Applied Power Electronics (CAPE)
Department of Electrical and Computer Engineering
University of Toronto
Toronto, Ontario M5S 3G4, Canada

***Abstract*— This paper addresses the issue of interfacing a digital controller with a real-time digital simulator modeling power electronic apparatus. One of the main challenges during such interfacing is the synchronization between the simulator operation and the outputs of the digital controller which are discrete switching pulses. A lack of synchronization between the fixed step-size simulator operation and the timing of the incoming switching event can adversely affect the accuracy of the simulation. Various off-line simulation algorithms are examined with an objective of assessing their applicability under real-time conditions. A new computationally efficient real-time simulation is presented. To demonstrate practical feasibility of the algorithm, a modular design of a real-time digital simulator and a digital controller for a Pulse Width Modulated (PMW) Voltage Source Converter (VSC) system is described. The design is based on a DSP-FPGA digital processing platform. Real-time simulation and experimental results of a 5kVA VSC system are presented.**

***Keywords*— Real-Time Simulation, Power Electronics, Digital Control, Digital Signal Processing.**

I. Introduction

Sophisticated power electronic apparatus are finding wide ranging applications at all levels of utility systems for enhancing system performance, stability and reliability. Controllers for such power electronic systems are increasingly being digital in nature, as opposed to analog, mainly due to their flexibility, programmability and superior computing and communication capabilities among a host of other advantages. Before commissioning a controller on the actual power system, however, testing is undertaken in order to achieve three main objectives:

- Evaluate controller performance under open-loop and closed-loop control conditions.
- Evaluate controller performance under steady-state and dynamic system conditions.
- Verify built-in diagnostic and protection strategies.

With the availability of powerful and affordable digital computers and DSPs the current trend to test a digital controller is to employ a real-time digital simulator that models the power electronic apparatus and the host power system. This approach, however, has brought new challenges with it. A real-time digital simulator simulating power electronic systems takes discrete switching signals as external inputs from the digital controller. Digital simulation being itself discrete in nature is unable to cope effectively with switching signals that arrive between two calcula:ion steps of the simulator. The conventional off-line approach of using small step-sizes for simulation to overcome the problem is not a favourable option under real-time conditions for reasons of practicality and economy. State-of-the-art simulators such as RTDS^{TM} employing fast DSPs are capable of simulating realistic size power systems with a step-size that is typically between 50 to 75 μs. Inclusion of switching elements in the simulation, however, necessitates the use of a much smaller step-size (of the order of 10 μs or lower) which creates a bottleneck for a simulator using the fixed step-size approach.

The paper starts with an overview of existing tools for testing digital controllers for power systems. The interaction of a real-time digital simulator with a digital controller is examined to give an insight into the problem. Thereon, a survey of currently available algorithms for accounting ***inter-step*** switching events in digital simulation is presented. The survey is carried out with an objective of assessing the suitability of the various algorithms for real-time simulation. A new algorithm is presented to synchronize the operation of a real-time digital simulator with incoming external switching events. The algorithm provides several fold improvement in accuracy over the fixed time step algorithm using the same step-size without sacrificing computational efficiency and without violating any real-time constraints. The viability of the algorithm is tested by using it to design a real-time digital simulator for a PWM VSC system. Real-time simulation and experimental results for the test sytem are presented.

II. Tools for Testing Digital Controllers

A. Transient Network Analyzers (TNAs)

A TNA is an assemblage of scaled down models of physical components. The topological layout of a TNA is similar to a physical power system. Its main strength, as with any analog set-up, lies in its real-time capability thus allowing a comprehensive testing of control hardware. However, there are several drawbacks with TNAs; they require significant resources to build and maintain, excessive time to prepare test set-ups, they are inflexible to rapidly readjust to different test conditions and they lack scalability for an accurate system representation. Some of these drawbacks are overcome by using a *hybrid* simulator which is a combination of analog components and digital computer models.

[†] Email : venkata@ele.utoronto.ca
[‡] Email : iravani@ecf.utoronto.ca

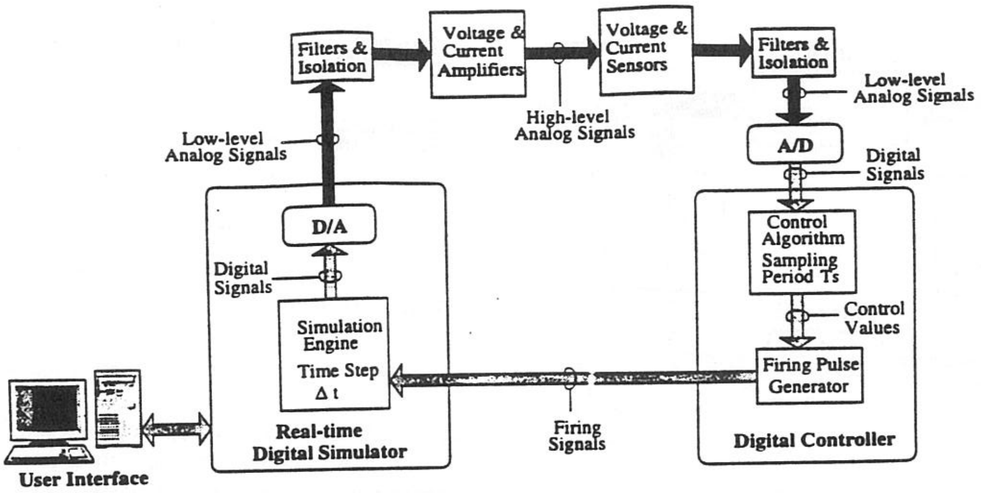

Fig. 1. A generic setup of a digital controller interfaced with a real-time digital simulator

B. Off-line Digital Simulation Programs

A digital simulation program emulates a physical system by solving the mathematical equations which describe the system behavior. A numerical integration rule is often employed to discretize the continuous time system differential equations and the simulation proceeds with a time step which may be fixed or variable. A digital controller that needs testing is represented, in the simulation program, in its functional form by logic statements and transfer functions that describe the behavioral properties between its input and output. Unlike a TNA there is no physical interface between the system simulator and the controller, thus allowing only a theoretical verification of the control design concept and precluding any real-time operation. Furthermore, it is a daunting task to accurately represent, in a simulation program, all the functional details of a practical digital controller. The advantages of a software simulation program, however, lie in its low-cost, low initial set-up time and the ease with which different test conditions can be accomodated. Examples of software tools for the simulation of power system and power electronics apparatus include EMTP, PSCAD/EMTDC, NETOMAC, and MATLAB/SIMULINK.

C. Real-time Digital Simulators

The best alternative to an analog TNA is a digital simulator that can solve the system equations in real-time. Rapid advancement in DSP technology coupled with the application of parallel processing techniques is giving rise to fully digital real-time simulators such as the RTDS^{TM}. Such simulators can easily accomodate tests, for example system faults, that are normally not performed on an actual power system during commissioning due to the engineering and economic risks involved. Despite their prevalence, their interaction with digital controllers has not been studied in sufficient detail to address the problems therein. The following section examines how a digital controller is interfaced with a real-time digital simulator simulating power systems containing power electronic apparatus.

III. A Digital Controller Interfaced with a Real-time Digital Simulator

Fig.1 illustrates a generalized setup of a real-time digital simulator interacting with a digital controller. The signals internal to the simulator are digital in nature due to the discrete numerical integration (with time step Δt) process undertaken in the simulator. The signals coming out of the simulator are analog in nature due to the D/A conversion. The low-level analog signals after adequate isolation and filtering are amplified before being applied to a test equipment or a controller. The digital controller setup usually consists of sensoring equipment and A/D converters as external peripherals to convert the high-level analog signals to low-level digital signals which are used as inputs to the control algorithm. Inside the controller the control algorithm may be executed on one or more digital processors. A firing pulse generator then translates the control values into discrete gating signals which are fed back to the simulator to control the power electronic switching devices modeled in the simulation. A user interface comprising the functions of communication, display and storage may be connected either to the simulator or the controller or both depending on the system/control variables that need continuous monitoring.

Fig. 2 illustrates a model which includes the essential elements taken from the set-up of Fig.1. In deriving this model it has been assumed that all amplifiers, sensors and signal converters are ideal without any restrictions on their precision or bandwidth. Two interfaces I and II are shown between the real-time digital simulator and the digital controller. Interface I represents the sampling action performed by the A/D converter on the analog signals v, i and is modeled by a sampler which operates at a specific rate T_s (called the controller *Sampling Period*) and a zero-order hold. The simulator can acquire the gating signal $g(t)$ from the digital controller only at finite intervals Δt due to the discrete nature of the numerical integration process it is carrying out. Interface II represents this Δt latency on the gating signal and is modeled by another sampler with sampling rate Δt. Given a certain T_s, based on the control design re-

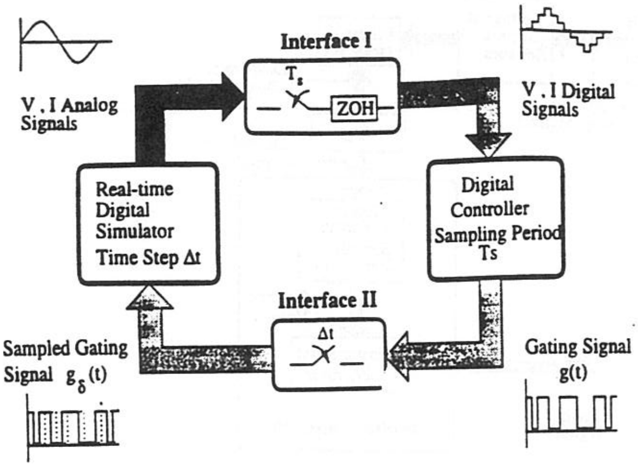
V, I Analog Signals
Interface I
T_s
ZOH
V, I Digital Signals
Real-time Digital Simulator Time Step Δt
Digital Controller Sampling Period Ts
Interface II
Δt
Sampled Gating Signal $g_\delta(t)$
Gating Signal g(t)

Fig. 2. Interfaces between a digital controller and a real-time digital simulator

quirements, Interface I does not present any problem if the signal converters are sufficiently accurate and fast. However, Interface II does present a significant obstacle since the timing of the incoming discrete firing signal may not necessarily be in synchronism with the time step chosen for the real-time digital simulator.

Fig. 3 illustrates the problem on a time grid progressing from left to right $t_0 < t_1 < t_2 \ldots$ with a time step of $\Delta t = t_{i+1} - t_i$, $(i = 0, 1, 2, \ldots)$. A fixed step-size numerical integration program operation is depicted by the curved arrows. Let x_i, $(i = 0, 1, 2, \ldots)$ be the states of the system computed by the simulator at every time step Δt and y_i, $(i = 0, 1, 2, \ldots)$ be the true states of the system obtained by taking the discrete events into account at their exact locations. The firing signal comes in at time t_e but is accounted for at time t_2 when the real-time simulator has already calculated the incorrect state x_2. The change in state requested by the switching event is acknowledged at time t_2 and is used to calculate state x_3 at time t_3.

The error in the solution of the system state stems from the *delay* introduced in the switching due to improper synchronization of the two discrete processes. Unlike the actual physical system the digital simulator is unable to respond instantaneously to a firing signal that comes in between two calculation steps. The physical system would respond to the firing signal at time t_e with the state y_e and y_2 would be the true state of the system at time t_2. Since the simulator can only respond at the end of a calculation step the actual switching may be delayed upto one time step in the worst case. Furthermore, since the incoming firing signal does not always occur at the same instant on the time grid of the simulator, the switching is not always realized at the same moment and therefore the delay introduced is not constant. The larger the time-step of the simulator the larger is the delay in switching.

Typically, the firing signal $g(t)$ for a power electronic switch is comprised of a series of rectangular pulses with varying pulse widths. Theoretically, its frequency spectrum $g(\omega)$ is not bandlimited. Sampling such a signal at a fixed rate Δt will result in an *aliased* signal according to Shannon's Sampling Theorem [6]. The spectrum of $g_\delta(\omega)$ would consist of overlapping cycles of $g(\omega)$ repeating every $\omega_{\Delta t} = 2\pi/\Delta t$ rad/sec. Aliasing arises due to the effect of spectral folding where a certain band of high frequency components get reflected in the low frequency region due to the limited sampling rate Δt. The resulting gating signal $g_\delta(t)$ thus is always an undersampled version of $g(t)$ and contains pulses having different positions and widths than that of $g(t)$. At the outset, the differences may seem rather insignificant, however, when an altered gating signal such as $g_\delta(t)$ is used to control a power converter, the effects become apparent in the harmonics of the output voltages and currents [4].

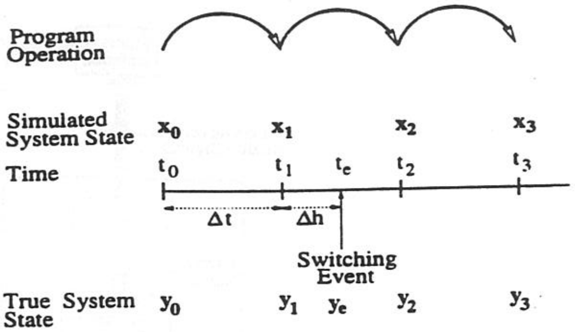

Fig. 3. Inter-step switching event in digital simulation

The following section examines some of the techniques used in off-line digital simulation to handle inter-step switching events and discusses their applicability for real-time simulation.

IV. Survey of Algorithms for Accounting Switching Events in Digital Simulation

Switching delay was first encountered in off-line simulation while simulating thyristor based power electronic circuits using line commutation. Since a thyristor turns off when the current through it goes to zero, switching is delayed by a fraction of one time step if the current zero occurs between two calculation steps. The techniques developed to solve this problem were later used for circuits using forced commutation and for circuits employing switches with gate turn-off capability. Although the current in a circuit equipped with switches such as GTOs or IGBTs can be interrupted at any time, switching delay in the simulation arises if the firing signal for the switch comes in between two time steps of the simulator. Fig.4 illustrates the algorithms that have been used to account for discrete switching events in off-line digital simulation.

The intuitive approach (Fig. 4(A)) to alleviate the errors due to delay in switching is to carry out the entire simulation with a small step-size $\delta t = \Delta t/n$, (n integer), so as to reduce the delay, but at the cost of a larger total simulation time.

In a variable time-step program (Fig.4(B)), as the name suggests, the time-step of the simulation is changed whenever a switching event comes in between two calculation steps. When the switching event is detected at time t_2 the algorithm backtracks to the state x_1

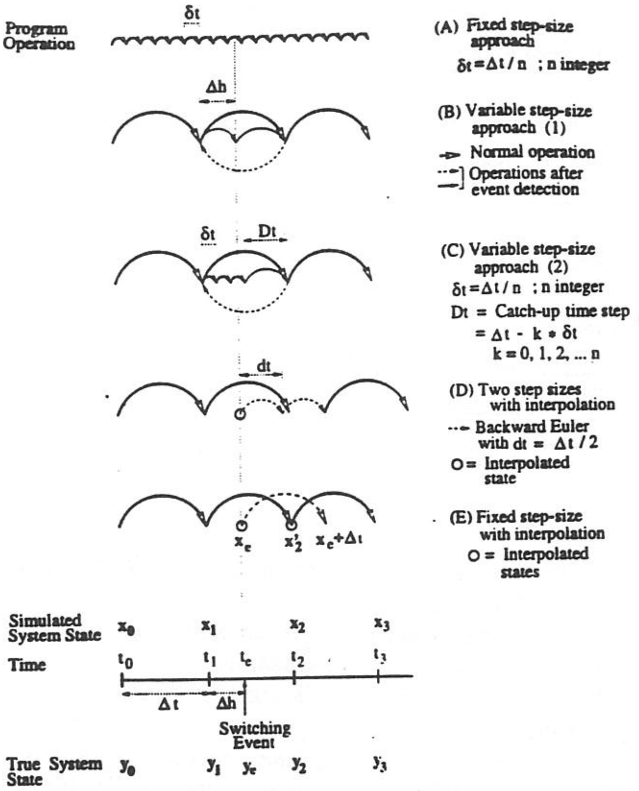

Fig. 4. Techniques for handling inter-step switching events in off-line digital simulation

at the previous time step, takes a smaller time step Δh and calculates state y_e, takes another time step $(\Delta t - \Delta h)$ and calculates state y_2, and then proceeds with a fixed time step Δt till the next switching event is detected. The variable time-step process would involve a re-formulating of the admittance matrix during the course of program execution everytime the time-step changes. If the study system is large, this would mean a significant computational burden.

Another variation (Fig. 4(C)) of the variable time-step method is a method where two time steps- one Δt and another $\delta t\,(= \Delta t/n)$ an integer submultiple of Δt are maintained. When the discrete event is detected at time t_2 the algorithm backtracks to time t_1 and starts calculating states every δt so that instead of accurately pinpointing the instant t_e it is finely straddled between two calculation steps δt. Once the event has been accounted for a catch-up time step $Dt\,(= \Delta t - k * \delta t,\ k = 0, 1, 2, \ldots n)$ is taken to time t_2. The advantage of this method is that the system admittance matrix can be pre-calculated and stored for the three time steps Δt, δt and Dt. This method was used for simulating an SVC (Static Var Compensator) circuit [3] using state variable formulation. Adam's second order integration rule was used to iterate around the event until adequate accuracy was achieved.

In the method of Fig. 4(D) trapezoidal integration is used for normal operation with time step Δt. When a switching event is detected, the method interpolates to obtain circuit variables at time t_e. Then, Backward Euler (BE) integration is used for the subsequent two half time $(\Delta t/2)$ steps after which normal operation resumes. Notice that the original time grid has now been altered. The main motivation behind this approach is that using half time step BE rule the resulting system matrix is the same as that needed for a full time step trapezoidal rule. Any other step-size would entail a matrix reformulation. This method has been implemented in the Microtran version of EMTP.

Fig. 4(E) illustrates another method which uses linear interpolation but the step-size is now fixed. Once the switching event is detected at time t_2 states x_1 and x'_2 are linearly interpolated to obtain state x_e at time t_e and the solution continues with the original time step Δt yielding a new solution $x_{t_e+\Delta t}$ one time-step later. An additional interpolation step between t_e and $(t_e + \Delta t)$ is taken to find the correct state at t_2 i.e., x_e and $x_{t_e+\Delta t}$ are now linearly interpolated to get x_2. This second interpolation step is taken to put the solution back on the original time grid. So, there are two interpolation steps and two regular solution steps from the time the switching event is detected at t_2 till the time the state x_3 at t_3 is calculated. This approach again has the advantage that the admittance matrix need not be re-formulated since the time step is fixed. However, this advantage comes at a significantly higher computational cost.

With regard to real-time computation the fixed step-size approach (Fig. 4(A)) with a small Δt is not feasible in the case of realistic size systems due to excessive computational speed requirement. The other algorithms (Fig. 4(B)-(E)) adopt one or more of the following practices that contradict real-time operation:

- Stepping back in time and changing states that have already been calculated.
- Altering the original time grid of the simulation.
- Iterations around the switching event.
- Varying the step size.

V. Fixed Step-Size with Interpolation and Clock Synchronization (FICS) Approach

Based on the discussion in Section IV, it was concluded that a new real-time simulation algorithm was needed which conformed to the following requirements:

- A fixed simulation time step.
- No alteration of the original time grid during simulation.
- Registration of the timing of the external switching events independent of the simulation process.
- A computationally efficient way to utilize the regis-

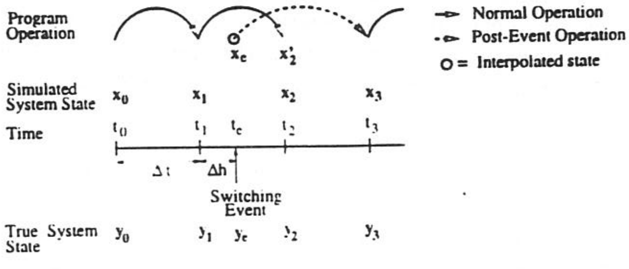

Fig. 5. FICS real-time digital simulation algorithm

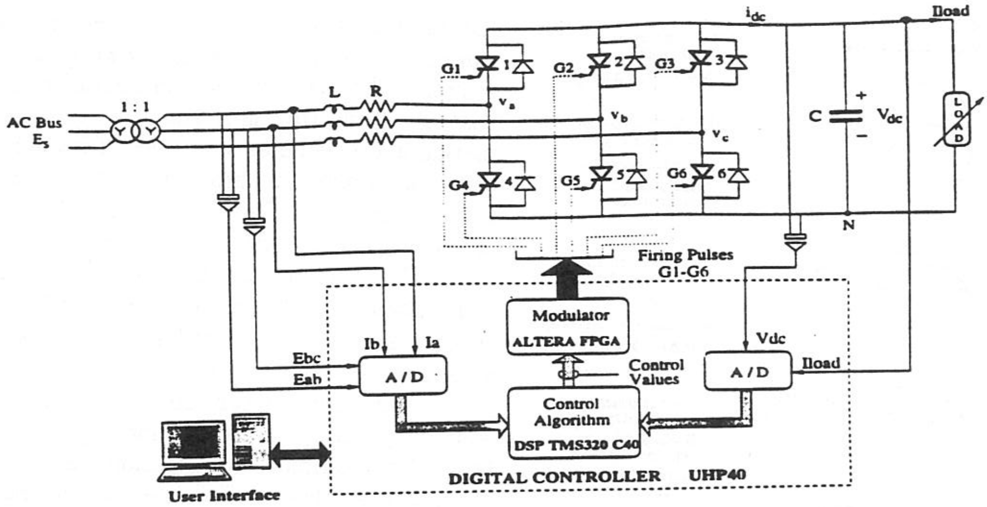

Fig. 6. Experimental set-up of the VSC system

tered timing to correct the simulated state.

The proposed algorithm (Fig. 5) relies on registration of the timing of the discrete switching event and a subsequent *correction* before the next state calculation.

At time t_2 the simulator has computed and sent out the state x_2 which is incorrect because a switching event arrived at time t_e while the simulator was still engaged in the calculation of x_2. Real-time operation does not allow recalling and changing state x_2. However, if the simulator were provided with the timing of the event relative to the simulation time grid, it can take certain corrective action before proceeding to calculate state x_3 at time t_3. The simulator cannot take preemptive action before calculating x_2 since it does not know *a priori* when the switching event will arrive, the event being controlled by processes external to the simulator.

At the beginning of every time step the simulator polls for the switching event and its timing information. Based on that knowledge the following operational paths are undertaken:

A. Normal Mode

This mode of operation is carried out when no switching event has been detected in the previous calculation cycle and it involves the following steps :

[A] The power electronic switch status is updated based on the available gating information.

[B] For network solution, trapezoidal numerical integration method is employed allowing recursive solution of the system state equations. All electrical elements are represented as equivalent admittances and current injections. Nodal analysis is then used to formulate a set of linear algebraic equations. The transient solution then proceeds by solving the linear equations for the unknown node voltages at each time step.

B. Post-event Mode

This mode is executed when a switching event has been detected in the previous calculation cycle :

[A'] The network variable (all states and inputs) are linearly interpolated at time t_e based on the states x_1. x_2 and Δh.

[B'] The network admittance matrix is reformulated based on $(2\Delta t - \Delta h)$.

[C'] The power electronic switch status is updated based on the new gating information.

[D'] A regular solution step is taken to calculate state x_3 at time t_3.

Notice that during the *correction* procedure (Steps A' & B') the original time grid is not altered. From the perspective of an external observer the apparent time step of the simulator is still fixed at Δt, there is only an internal adjustment of the time step to re-synchronize with the real-time clock whenever a switching event is detected. Linear interpolation provides a computationally efficient way to estimate the circuit state without going through the full calculation procedure. For the practical implementation of the FICS algorithm. however, there should be a provision to presicely capture the inter-step switching event and relay its timing information to the simulator at the beginning of the next calculation cycle.

VI. Study System

A PWM VSC system (Fig. 6) was selected as the study system for building the real-time digital simulator and the experimental set-up. The system parameters are as follows : $E_s = 110V_{ll}$, $60Hz$. $R = 0.5\Omega$, $L = 3.0mH$, $C = 4900\mu F$, $f_s = 1kHz$. The VSC is composed of six IGBT switches with anti-parallel diodes. The IGBTs are rated for a constant current of 25A and a maximum voltage of 300V. The primary function of the digital controller is to regulate the reactive power injected by the VSC into the AC bus while maintaining the DC link voltage constant.

VII. Design of the Real-time Digital Simulator and Controller

The real-time digital simulation and the experimental control was carried out with the aid of the Universal High Performance (UHP40)[5] digital processing platform. Its architecture is based on a 32-bit floating point DSP TMS320C40 and a FLEX 8000 FPGA. The FPGA shares the global data and address space of the DSP. A

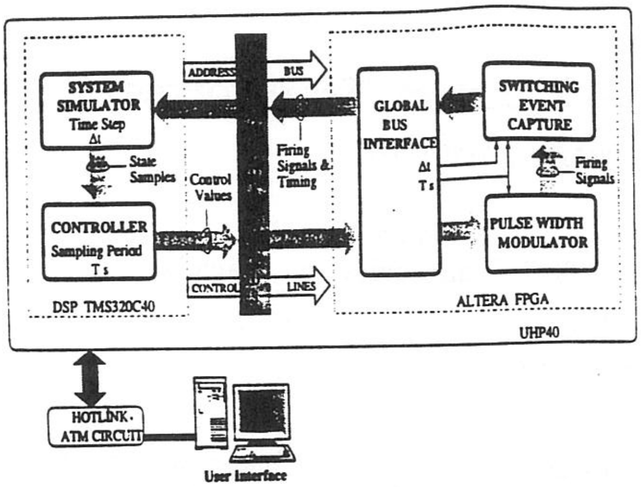

Fig. 7. Real-time digital simulator hardware

Monitor program[5] running on the PC provided two-way interaction between the user and the UHP40 for signal monitoring and for varying simulation parameters.

The real-time simulation test-bed (Fig. 7) consists of five main modules: System Simulator, Controller, Pulse Width Modulator, Switching Event Capture unit and the Global Bus Interface. The DSP performed the functions of the simulator and the controller, while the FPGA does the functions of modulator, switching event capture and the bus interface. The DSP source code is written in C and the FPGA is programmed using digital hardware design techniques. The scenario of a real-time digital simulator interacting with a digital controller is made possible by a synchronized bidirectional data transfer between the DSP and the FPGA managed by the Global Bus Interface. Once the real-time simulation in the DSP has been initialized the user can alter system parameters and view system signals on-line using the Monitor program. In the experimental set-up (Fig. 6) the DSP was used for implementing the control algorithm and other high-level functions while the FPGA was used for modulation purposes. Dedicated A/D converters mounted on the UHP40 platform were used for the acquisition of system voltages and currents.

VIII. Experimental Results

Several tests were conducted under similar operating conditions on the real-time simulator and on the experimental set-up in order to verify the effectivenss of the FICS algorithm. The real-time simulation was carried out with a time step of $100\mu s$ selected based on the performance of the C40 DSP. Fig. 8 through Fig. 11 show steady state currents and voltages of the VSC. These results also agree well with those obtained from off-line simulation using a time step of $10\mu s$.

IX. Conclusions

A fully digital real-time simulator is a powerful and economical means to verify a power system controller. This paper focussed on interfacing a digital controller with a real-time digital simulator modeling power electronic apparatus. Existing off-line simulation algorithms were examined from the point of view of their application to real-time simulation. Most of these algorithms, although very effective under off-line conditions, were found to violate real-time constraints. A novel simulation approach was presented to handle inter-step switching events in real-time digital simulation. Real-time simulation and experimental results for a PWM VSC system prove the feasibility and validity of the proposed approach.

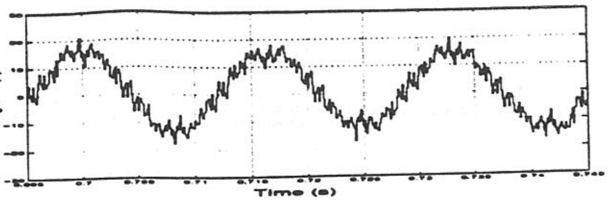

Fig. 8. Steady state current i_a (Real-time simulation)

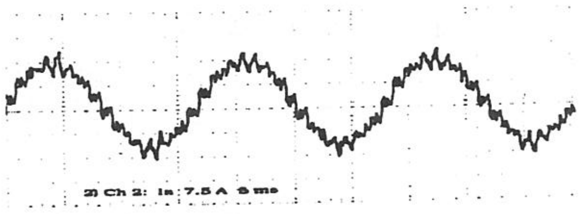

Fig. 9. Steady state current i_a (Experiment)

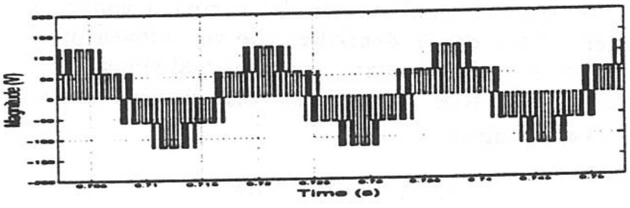

Fig. 10. Steady state voltage v_{an} (Real-time simulation)

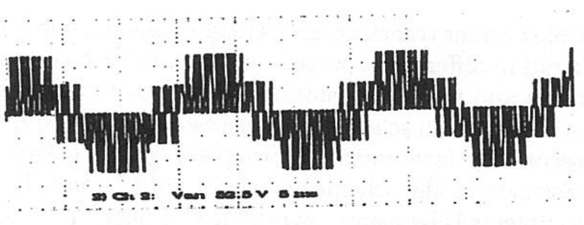

Fig. 11. Steady state voltage v_{an} (Experiment)

References

[1] S. Acevedo, L. R. Linares, J. R. Marti, Y. Fujimoto, "Efficient HVDC converter model for real-time transients simulation", *IEEE Transactions on Power Systems,* Vol. 14, No. 1, pp. 166-171, February 1999.

[2] R. C. Durie, C. Pottle, "An extensible real-time digital transient network analyzer", *IEEE Transactions on Power Systems,* Vol. 8, No. 1, pp. 84-89, February 1993.

[3] A. M. Gole, V. K. Sood, "A static compensator model for use with electromagnetic transient simulation programs". *IEEE Transactions on Power Delivery,* Vol. 5, No. 3, pp. 1398-1405, July 1990.

[4] V. R. Dinavahi, "Real-time Digital Simulation of Switching Power Circuits", PhD Dissertation, University of Toronto, 2000.

[5] S Krebs, "*UHP40 User's Manual*", Technical Report, University of Toronto, December 1995.

[6] A. V. Oppenheim, A. S. Willsky, *Signals and Systems*. Prentice Hall 1997.