High-Speed Real-Time Simulators for Engineering Design

R. E. Crosbie, N. G. Hingorani

The use of computer simulations is now an established technique in engineering design. Many of these simulations are used to predict the expected behavior of systems that are not yet built, or of existing systems in modes of operation, such as catastrophic failure, in which it is not feasible to test the real system. Another use of computer simulations is for training and testing purposes in which the simulation is interfaced to real hardware, software and/or a human operator and is required to operate in real-time. Examples are plant simulators for operator training or simulated environments for testing hardware or software components. The primary requirement of a real-time simulation is that it must complete all the calculations necessary to update the simulator outputs as well as all the necessary data I/O within the allotted frame time. Many real-time simulations use frame times in the range of a few milliseconds and greater.
There is an increasing number of applications, for example in power electronics and automotive systems, in which much shorter frame rates are required. This paper reviews some of these applications and the approaches to real-time simulation that can achieve frame times in the range 5 to 100 microseconds.

Keywords: computer simulation, real-time, engineering design, digital signal processors.

1 Introduction

The use of computer simulations is now an established technique in engineering design. Developed originally for military applications during and immediately after the Second World War, the use of computers (analog and digital) to simulate both existing and proposed systems spread to many fields of application, including engineering design, in the immediate post war years. In the first two or three decades after the war, analog computers were favored for simulations applied to engineering design. Their ability to solve differential equations more rapidly and cheaply ensured their dominance over the digital computers of the time. As the power and cost effectiveness of digital computers quickly increased they eventually replaced the analog computer which was handicapped by low accuracy, difficult programming and setup, and high maintenance demands. The development of the personal computer put the final nail in the coffin of analog computers as software was developed that allowed the personal computer to be used as a low-cost, flexible, accurate simulation tool. In recent years, all kinds of computers, from laptops to supercomputers have been used to support an ever increasing variety of computer simulations, including many applications to engineering design. Many software packages have been developed to support these simulations.

There are several reasons for using simulations to improve the design process affecting different phases of the engineering life cycle from conceptual design, through detailed design, testing, training, diagnosis of faulty operation, and decommissioning. In many cases, the simulation is required to produce time-domain or frequency-domain data representing the behavior of the system under different operating conditions. These can be used, for example, to confirm that a proposed design is likely to perform within specifications. This is particularly useful for evaluating systems operating under extreme fault conditions, such as emergency shut down of a plant after failure of a coolant pump, or the handling of an aircraft following an engine failure for example. Simulation can also be used to try to reproduce and hence diagnose faulty system operation.

None of these applications require the simulation to execute in exactly the same time as that in which the system operates. It is often convenient if they do not. A simulation of a slowly varying phenomenon that takes minutes or hours to complete, such as the slow build up of fission products in a nuclear reactor, benefits from simulations that execute much faster than real-time. Simulation of a very rapid phenomenon, such as the operation of electronic switches, will execute more slowly than real time to allow observation of the changes taking place. It is not even necessary for the simulation to execute in a strictly time-scaled fashion, i.e. N times faster or slower than real-time. Some sections of a simulation may run more slowly than others depending on the amount of computation required to complete a given time period. This is normally quite acceptable.

There are, however, applications of simulation in which a strict time synchronism with the operation of the simulated system is required. These are known as real-time simulations. They are used whenever the simulation is combined with real system components or human operators.

There are significant differences between non-real-time and real-time simulations that go beyond the need for real-time synchronization. A real-time simulation is always interfaced to external hardware, software, human operators or a combination of all three. The nature of the interface depends, of course on the nature of the external subsystem. It may be digital or analog and, for human operators in training simulators, it may involve the construction of a realistic system control center, such as an aircraft flight deck, or a plant control room. This means that in addition to performing the computations that solve the equations describing the behavior of the model, the simulation program must also handle the time-synchronized input and output of data via the real-time interfaces. It is very common for a real-time simulation

of this kind to be built upon a real-time operating system (RTOS) that is capable of timing the execution of the simulation program appropriately.

One of the key parameters of a real-time simulator is the frame rate at which its outputs are updated. Many real-time simulators, including those used for operator training, perform satisfactorily with frame times in the range 10 to 100ms. All the calculations needed to advance the simulation by one frame must be completed, along with all necessary data transfers within one frame. In some applications real-time simulation is used as a test environment for real hardware or embedded software, referred to as the system under test (SUT). In some cases much shorter frame times (<10μS) are required because of the high-frequency dynamics of both the simulated system and the SUT. Such applications are found, for example, in aerospace, automotive and power electronic systems. We consider approaches to producing real-time simulations in cases such as these. To distinguish these higher speed examples of real-time simulation we will apply the term high--speed real-time (HSRT) simulation to applications requiring frame times of less than 100μS.

2 The need for HSRT simulation

The maximum acceptable frame time for a real-time simulation is determined by the dynamics of the simulated system and the hardware to which it is interfaced. Training simulators that are interfaced to instruments and controls used by human operators would not normally benefit from HSRT operation because humans can not absorb information or react at the corresponding data rates. HSRT simulation is confined to situations in which a simulated system with a wide dynamic range is interfaced to equipment that is sensitive to the high-speed dynamics of the simulation. A common application occurs in the design of embedded controllers for high--speed systems. Two examples that have received a lot of attention are in automotive systems and power electronic systems.

Embedded computers are now widely used as automobile engine electronic control units (ECUs). Testing of these embedded controllers on real engines is expensive and time-consuming, and simulation is increasingly being used instead. These simulations often require frame times of 100 μS or less, especially for high-performance engines such as are found, for example, in Formula 1 racing cars.

Power electronic systems are used to convert electrical power from alternating to direct current and vice versa and for conditioning and stabilizing the resulting power outputs. They are widely used for producing power in the form required by an electrical load, ac or dc, with appropriate current and voltage ratings, and with the necessary stability and reliability. They range from encapsulated low-wattage power supplies for laptops and domestic electronic devices, through industrial electric drives rated at kilowatts to megawatts, to power distribution and transmission components and systems that convert and control hundreds of megawatts of electrical power. Converters, which convert ac to dc or dc to ac are key components of these systems. The converters consist of configurations of switches that can be turned on and off via a control signal. The timing of this switching determines the form of the converter output. The feedback controllers that control this switching are often based on pulse-width modulation (PWM) techniques with PWM frequencies of tens of kilohertz. The testing of controllers for this type of power electronic system using real-time simulations can require frame times of less than 10 μS.

Two trends are increasing the need for HSRT simulation. As the power and cost effectiveness of simulation technology is more widely recognized, it continues to penetrate new fields of application. Furthermore, within particular fields in which real-time simulation is already established, advances in technology that reduce response times and increase frequencies demand shorter frame times from the corresponding real--time simulations. These trends are likely to persist causing an increasing need for HSRT simulation.

We will discuss some of the commercially available solutions for implementing HSRT simulations, and will also present a new approach adopted by a team at California State University, Chico to a particularly demanding application requiring frame times beyond the reach of currently available commercial systems.

3 Available systems for HSRT simulation

Special techniques are required to achieve such short frame times. High-speed real-time (HSRT) simulations of this kind represent one kind of hard real-time embedded system, and can be based on the same real-time operating systems (RTOS) used for many high-performance embedded systems. A number of commercial systems that can support HSRT simulation are available from companies such as D-SPACE, Ref [1], Opal-RT, Ref [2], ADI, Ref [3], and RTDS, Ref [4].

The dSPACE DS1006 simulator (Ref [1]) is based on an AMD Opteron™ Processor 248, a high-performance server processor. Multiprocessor systems are available that use several processor boards connected via optical fiber. Opal-RT (Ref [2]) uses clusters of low-cost PCs – incorporating off-the--shelf technologies and FPGA-based reconfigurable I/O – that can simulate electric power converters and drives at time steps down to 10 μS. Applied Dynamics International (ADI) offers a HSRT simulator (Ref [3]), based on the Motorola MVME5500 PowerPC. One processor acts as a user interface processor, a second processor runs the models under RT-Exec, a real-time operating system that "*guarantees microsecond level determinism*". Multiple processors can be used

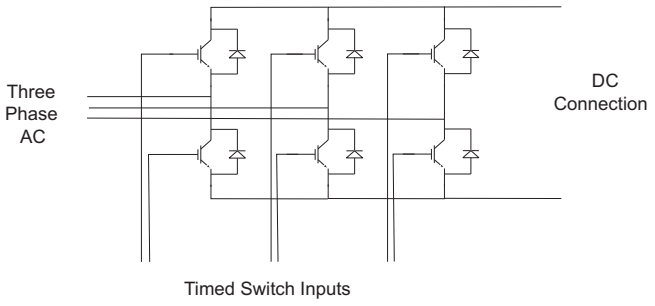

Fig. 1: Switch connections for a 3-phase voltage-sourced converter. Each switch consists of a turn-off device and a diode in parallel. For high-voltage/high-power converters each switch may contain many devices in series and/or parallel

for more demanding applications. RTDS Technologies (Ref [4]) specialize in providing simulation systems customized to specific, large-scale power system simulations. Their systems consist of racks of up to 20 rail-mounted cards. Two types of processor card are available, one based on dual IBM PPC750CXe RISC processors, the other containing three Analog Devices ADSP21062 SHARC DSPs. Other types of card are available for digital and analog I/O, user interface, inter-rack communication etc.

4 HSRT power electronic simulation

The Real-Time Power System (RTPS) simulations developed at the McLeod Institute of Simulation Sciences at CSU, Chico concentrate on simulations of systems based on three-phase bridge converters using electronic switches with turn-on and turn-off control. Typically, one side of the system is connected to three-phase a.c. and the other to a d.c. system. The direction of power flow, i.e. whether the converter is acting as a rectifier or as an inverter, is determined by the timing of the operation of the switches. A typical system (Fig. 1) consists of two such converters connected by a dc line with a three-phase transformer and three-phase ac system at the remote ends of both converters. The dc line is often represented by a large capacitance, but in some cases involving long lines a distributed line model of some kind is required. The converter switches are switched on and off by control pulses from a feedback controller that controls the timing of switch operation in order to maintain the required power specifications. Pulse-width modulation controllers are frequently used for this purpose. Switching frequencies can range from a few hundred Hz to tens of KHz. Frame times of 10 μS or less are necessary for accurate simulation in these cases. For real-time operation this implies that the simulation must re-compute the state of the system within a 10 μS frame time. Systems that use higher PWM frequencies require even shorter frame times. Simulation studies suggest that 20 KHz PWM controllers may require frame times as low as 2 μS.

5 Using DSP arrays for HSRT simulation

The MISS Center at Chico was asked to develop low-cost HSRT simulations of typical power electronic systems using off-the-shelf components that can perform with frame times shorter than those available from currently available commercial systems. A future need for frame times shorter than 5 μS was projected although the initial goal was to implement typical generic models at 10-μS frame times. A further requirement was that the technology should be scalable so that more complex models could be accommodated with minimal impact on achievable frame times.

The approach selected was to use PCI boards containing small arrays of digital signal processors (DSPs) inserted into a conventional desktop computer. The real-time simulation executes on the DSPs and the user interface runs under the Windows operating system on the host computer. Rather than develop a custom user interface, an existing simulation system was selected to provide this feature. The Virtual Test Bed (VTB), developed at the University of South Carolina (Ref [5]) was chosen for this purpose. The VTB supports user entry of model parameters, simulation control parameters, control of the simulation, and provides powerful graphical display capabilities through its VXE graphics system. Use of the VXE permits detailed examination of complex waveforms generated during the real-time simulation runs.

The DSP boards used are based on Analog Devices SHARC (Super Harvard ARChitecture) processors (Ref [6]). Boards containing 4 processors are available from several manufacturers; the boards used at Chico are from Bittware Inc. (Ref [7]). This board contains four Analog Devices TigerSHARC processors with a 250 MHz clock. Clock speeds for these processors appear to be quite slow compared to, say, Pentiums, but this can be misleading. The processors have pipelined floating point adders and multipliers that can, under suitable conditions, produce several floating-point results in each clock cycle.

6 Performance issues

Several simulations have been developed of varying complexity. The mathematical models on which they are based consist of up to 40 differential equations. The three most important factors affecting the execution speed of the array of SPs are model partitioning, data transfer and code efficiency.

The models that have been implemented so far lend themselves to a simple partitioning between processors. The systems split naturally into two similar halves, each of which is allocated to one of the four DSPs on a single PCI board. The controllers for both sides are assigned to a third processor and the remaining DSP is used to handle synchronization and communication with the host processor. Several considerations influence the way in which the model is partitioned. Ideally this should be done in a natural fashion in which each processor represents a natural subdivision of the complete system. This is particularly true of the controller model bearing in mind that one application is to combine the simulation of the actual power system with a real controller. Separation of the controller model into a separate processor dedicated to that purpose allows easier replacement with the real controller.

At the same time it is important to try to equalize the computing loads so that no single processor unduly delays the completion of each frame. Current models run with frame times between 4.5 μS, for a simple model such as is shown in Fig. 1 (with feedback control at each end), and 11.9 μS for a more complex model. The first of these models runs on 4 processors and the ratio of the longest to the shortest execution times for the individual processor tasks is less than 1.1:1. For the more complex model, almost all the extra load is taken by the two processors that simulate the power system, and the ratio in this case is almost 3:1. Clearly, distributing the load differently, or using more processors could improve performance significantly. Ultimately it is a matter of judgment how to make the trade off between convenience and natural partitions on the one hand and minimization of execution time by equalizing the frame times of the different processors.

The DSP board provides several ways of performing processor to processor and board to host data transfers. Initially common memory was used as a convenient method of inter-processor communication, but accessing common memory proved to be a much slower than expected. The use of link

ports proved more efficient. The board contains a number of ways in which DMA transfers can be used. Care has to be taken to use the DMA features of the board to maximum effect.

Much of the programming of the models is carried out using the C++ language, but the compiler does not make optimum use of the capabilities of the processor and time-critical code needs to be efficiently hand coded. This is not a trivial task, but fortunately libraries are available containing efficiently coded routines for common operations. The RTPS models use efficient matrix-multiply routines based on modifications of standard routines found in the libraries that are provided with the DSP software.

Work continues on optimizing code for the current models and to investigate scalability using larger numbers of processors in more complex simulations. Automated methods of model development that rely less on manual processes are under development. Further speed up should also be possible using the latest 500-MHz TigerSHARC processors.

7 Conclusions

Computer simulation is now used in almost all types of engineering design. In some cases simulations are required to work in synchronism with the execution of the system being simulated. This is required, for example, when the simulated system must be connected to real hardware, to embedded software or to human operators. A number of commercial systems are available that support the development of real-time simulations. Increasingly, applications are emerging that require frame times of 100μS or less and in some cases, such as power electronic systems with high-frequency PWM controllers, frame times of less than 10μS may be required. In these extreme cases special care is needed to achieve the required short frame times. One approach, used for high-speed power electronic simulations at California State University, Chico, is to use arrays of digital signal processors with custom designed software. These simulations have so far achieved execution times as low as 4.5μS for models consisting of approximately 20 differential equations.

8 Acknowledgments

The work described in this paper is the result of a team effort. The authors wish to thank Professors John Zenor, Richard Bednar, Dale Word and Ralph Hilzer and numerous graduate students for their contributions, and Dean Kenneth Derucher and Dr. Larry Wear for their support and encouragement. The research described in this paper is supported by the US Office of Naval Research (ONR) through Grants N00014-01-1-0394 and N00014-03-1-0950.

References

[1] dSPACE Inc., "dSPACE Sets the Pace in HIL Simulation Again", Feb. 2004, www.dspaceinc.com/ww/en/inc/company/press/pr_0402001.htm.

[2] Opal-RT Technologies Inc.: "RT-Lab Electric Drive Simulator". http://www.opal-rt.com/catalog/products/2_es/eds /description.html.

[3] Applied Dynamics International, "The RTS Ultra-High Performance Real-Time Simulator". http://www.adi.com/pdfs/product/datasheet_rts.pdf .

[4] RTDS Technologies Inc. Information available at http://www.rtds.com/hardware.htm.

[5] Dougal, R. A., Liu, S., Gao, L., Blackwelder, M.: "Virtual Test Bed for Advanced Power Sources." *J. of Power Sources*, Vol. **110** (09/02), No. 2, p. 285–294.

[6] Analog Devices Inc.: "SHARC Processors." http://www.analog.com/processors/processors/sharc/.

[7] Bittware Inc.: "Embedded SHARC DSP Solutions." http://www.bittware.com/.

Roy E. Crosbie
e-mail: rcrosbie@csuchico.edu

McLeod Institute of Simulation Science
California State University, Chico
Chico, CA 95929-0003, USA

Narain G. Hingorani
Consultant

Los Altos Hills, CA
USA

