Integrating Electromagnetic Transient Simulation with Other Design Tools

A.M. Gole, P. Demchenko,
University of Manitoba

D. Kell, G.D. Irwin
Manitoba HVDC Research Centre

Winnipeg, Canada

Abstract - **Recent work in the area of transients simulation has shown how a control analysis package such as MATLAB[1] can be interfaced to an emtp-type program. Here, this approach is extended one step further and allows for the integration of an arbitrary number of different programs. The paper describes how such interfaces can be made using the PSCAD/EMTDC[2] platform as a typical platform. The method of interfacing is demonstrated using the example of a Current- Reference Pulse Width Modulated Induction Motor Drive. The controls modelled in the GUI based control software SIMULINK[1] directly interface with the transient simulation program and the mathematical package MATHCAD[3] is used for further data processing.**

Keywords: Electromagnetic Transients Simulation, Open Systems, Hybrid Simulation

I. INTRODUCTION

There is a growing trend in software development towards an Open Systems approach. In this approach, each piece of software is designed to satisfy a certain protocol that makes it possible for these programs to be interfaced easily. The recent inclusion of the Dynamic Data Exchange (DDE) capability in Operating Systems such as Windows 98+ and interprocess communications in Solaris and other Unix implementations make such developments easily implementable. This makes it possible for programs developed by different vendors to be used together with full interaction. The emtp-type program PSCAD/EMTDC [1] has been designed to take advantage of such an approach.

Previous work in this area [2] includes writing of the emtp-type algorithm using the mathematics and control analysis package MATLAB [3] as well as the interfacing of MATLAB with other transient programs such as ATP and PSCAD/EMTDC [4,5]. This work is extended here to include other programs such as the graphical user interface (GUI) based control system simulation software SIMULINK[6] and the mathematical analysis scratchpad MATHCAD [7].

1. MATLAB and SIMULINK are registered to The MATHWORKS, Inc.
2. PSCAD/EMTDC is a registered trademark of the Manitoba HVDC Res. Centre
3. MATHCAD is a registered trademark of MATHSOFT Inc.

Consider the following scenario which is now possible with this approach: while viewing the graphical system diagram in the emtp-type program, the user can click on a component that opens up a design tool such as MATH-CAD. During the simulation, some of the components are modeled using various MATLAB and SIMULINK block-sets and others in the emtp-type environment. The end of run automatically spawns another process in which a brilliant 3-dimensional view of the results can be viewed using some other rendering software. Finally, clicking on another component opens up a word processor report, with the graphs and other information from the simulation directly incorporated into it.

This synergistic approach of combining an electromagnetic transients program with other analysis packages greatly enhances the power available to the simulation engineer. The emtp-type program is extremely fast and has well established power system models. The analysis program could have excellent capability for data analysis or control modelling but may not be as efficient for direct simulation. It is important however to make the interface as transparent as possible so that the user is not unnecessarily burdened with the details of the interfacing.

Two types of interfaces are possible. In the first type (referred to as an "On-line Interface"), the two programs exchange information continuously during the simulation run. This data is typically exchanged at the end of every timestep and is used by each program for computations during the next timestep. This approach is necessary for example when modelling the controls in MATLAB/SIMULINK and the circuit in PSCAD/EMTDC.

In the second type of interface (referred to as "Off-line Interface"), the analysis program is invoked from within the GUI of the transient simulation program, but is used for data pre-processing and post-processing. For example, a program for obtaining electrical motor parameters from laboratory data can be invoked, the data entered, and the computed parameters directly made available to the simulation program. Similarly the program can be invoked again at the termination of the run and be used for data post-processing, such as harmonic analysis, three-dimensional plotting and so on.

II. ON-LINE INTERFACE

As an example of an on-line interface, we describe an interface between PSCAD/EMTDC and SIMULINK. This interface is constructed using an interface to MATLAB described in an earlier publication[5]. This is because SIMULINK is started from within MATLAB. The technique for interfacing is as shown in Fig. 1. Data input to PSCAD/EMTDC is carried out using a GUI interface. After drawing the circuit diagram and entering values via pop-up menus that are selected by clicking on the component's icon, the resulting schematic diagram is translated into a data file which includes the fixed network data and a FORTRAN file called DSDYN that contains dynamic models such as control blocks. This interface is a powerful feature as it allows users to develop their own models in FORTRAN and add these to the PSCAD/EMTDC component repertoire. This feature has been used to establish the communication between the two programs.

MATLAB on the other hand is an interpreted language which contains a large number of mathematical functions and control system blocks. These are invoked using statements entered directly onto the MATLAB command line or directly called through a "m-file". One feature available in MATLAB is that the MATLAB engine can be started from a user written FORTRAN or C program. This feature allows the communication with PSCAD/EMTDC to be established. An m-file statement can be used to start the SIMULINK process.
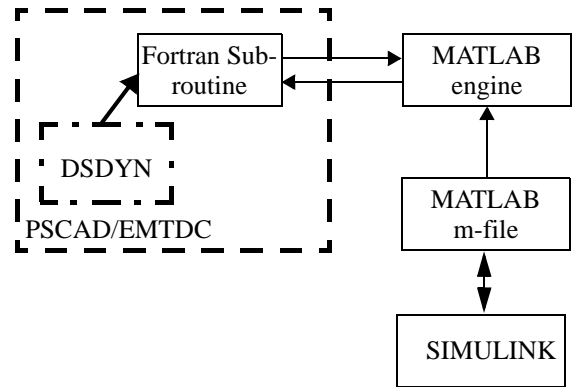

Fig. 1. Structure of PSCAD/SIMULINK INTERFACE

SIMULINK is a powerful general-purpose simulation tool with a graphical interface. A schematic of the control system to be simulated can directly be drawn using the GUI capabilities.

The process is automated so that very little user input is required to design the interface. The user supplies the name of the m-file through which SIMULINK is started, the name of the SIMULINK file and the number of input and outputs. Fig. 2. shows an example case of a Pulse-Width modulated Inverter Drive in which the controls are modelled in SIMULINK. The generic MATLAB/SIMULINK block in PSCAD has a single vector input and a single vector output. Components of this vector are the individual
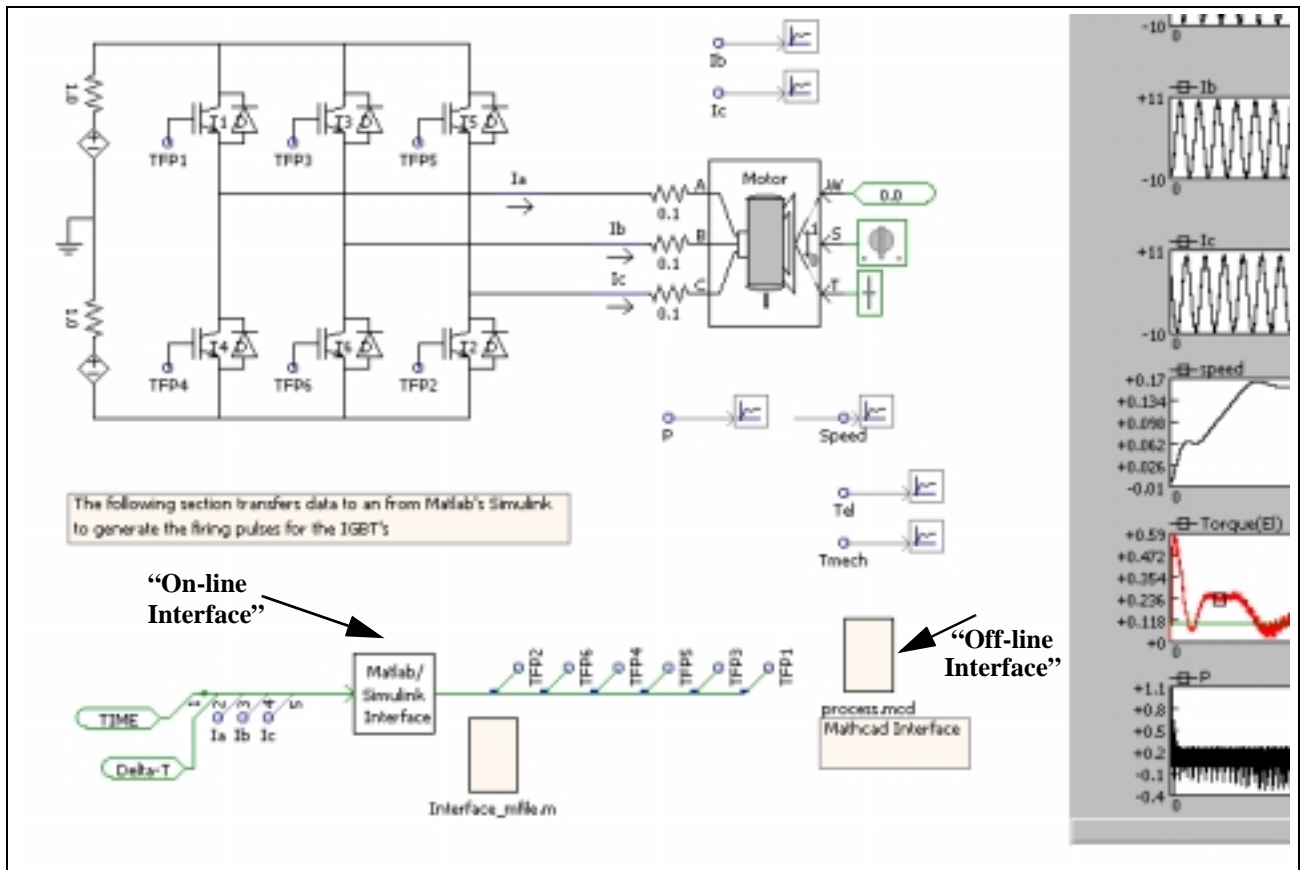

Fig. 2. PSCAD/EMTDC Palette Showing Interfaces

inputs or outputs. The block shown has five inputs: time, the timestep and the three ac currents; and six outputs, namely the six firing pulses for the IGBT (insulated gate bipolar transistor) switches. The SIMULINK model is shown in Fig. 3 which is a screen capture shot of the actual SIMULINK palette.

The interaction between the programs is two-way. Any change in the SIMULINK model immediately impacts the PSCAD/EMTDC model and vice versa.

Special attention has to be given to components in SIM-ULINK that require storage between timesteps. These so called "state variables" have to be passed to PSCAD for storage between timesteps and also in order to be able to execute a successful start from a "snapshot". PSCAD/EMTDC uses the snapshot feature to restart from an inter-mediate time instead of starting each run from t = 0.

Although this hybrid on-simulation gives the user a powerful repertoire of features available in SIMULINK, it should be noted that the simulation speed can be consider-ably less than that of the emtp-type algorithm running on its own. This aspect is further discussed in Section IV.

III. OFF-LINE INTERFACE

Whereas the on-line interface allows for data exchange to happen during the simulation run, the off-line interface allows for data exchange either before the run commences or after the run terminates. The pre-processing capability allows for parameter preparation before a run. For example, a program for obtaining electrical motor parameters from laboratory data can be invoked, the data entered, and the computed parameters directly made available to the simula-tion program.
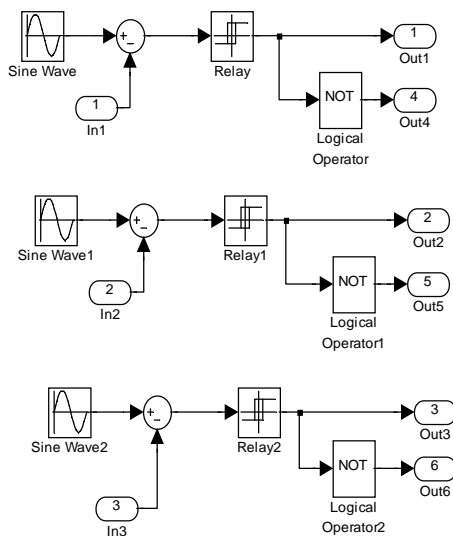

Fig. 3. Hysteresis PWM Control Model in SIMULINK

Similarly the program can be invoked again at the ter-mination of the run and be used for data post-processing, such as harmonic analysis, three-dimensional plotting and so on.

The off-line interface is easy to design. Reference to a file containing the application program is available on the PSCAD/EMTDC palette and merely clicking on this starts the program. The reference to such a file is seen in Fig. 2. Data exchange between the programs is carried out using intermediate storage files. The file produced by the applica-tion program is read by PSCAD/EMTDC and vice versa. As these files are accessible when the run is not active such data transfer is only permitted either before the run com-mences or after the run ends. Note that having all these files accesible from the PSCAD/EMTDC palette keeps all related programs together in the same environment. These programs could be started seperately, but then the user would have to enter filenames from which the data is to be picked up and so on, a process which could be fraught with confusion when there are a large number of different simu-lation cases under study.

IV. SIMULATION EXAMPLE

A. Illustrating the On-Line Interface

The example shown in Fig. 2 demonstrates the use of this approach. The example models a hysteresis Pulse-Width Modulated motor drive. In this method any desired current waveform can be applied to the load. If the actual load current in any phase is less than the applied reference current, the upper IGBT switch is turned on thereby con-necting the load to the positive dc source and hence increasing the current towards the reference. If the load current is larger than the reference, the negative switch is turned on. In order to avoid excessive switching, a toler-ance or "hysteresis" band is provided and no control action is taken if the current is within this band. The switches in the upper or lower arms respectively are turned on only when the current makes an excursion below or above this hysteresis band. In this manner, a load waveform which follows the reference with a maximum error equal to the hysteresis band is produced.

Typical simulation waveforms (phase current, electrical torque and speed) for starting the induction motor from zero speed with an applied 10 Hz current of 10 A peak are shown in Fig. 4. The motor eventually reaches a final speed of 0.16 pu (consistent with the applied 10 Hz current and the applied 0.1 pu load). As can be seen, the current in the machine is nearly sinusoidal showing that the hysteresis PWM control system modelled in SIMULINK is correctly interfaced with the transient simulation on PSCAD/EMTDC.
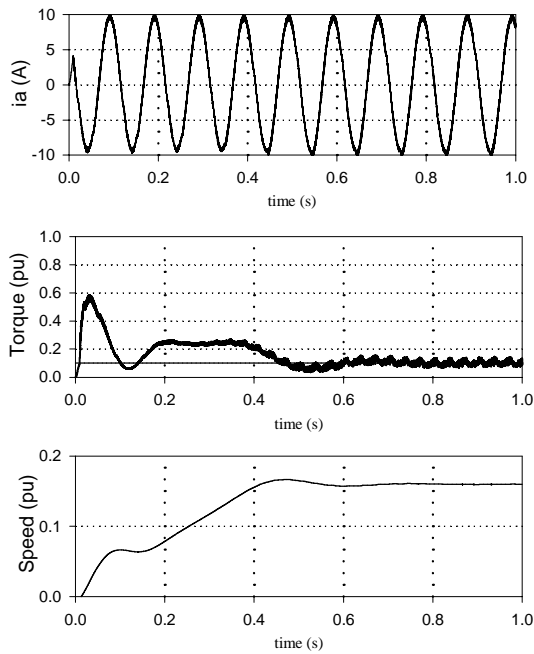
Fig. 4. Inverter and Motor Waveforms

The 1 second simulation required 4 minutes of CPU time on a 500 MHz Pentium III computer with 128 MB RAM running WIndows 98. The selected timestep was 50 μs. When the entire circuit is modelled completely in PSCAD/EMTDC, the same run takes about 15 s of CPU time - which is considerably faster. The advantage of using the interface is not the gain in computing speed, but rather the programming flexibility offered by the available packages.

B. Illustrating the Off-line Interface

As an example of the off-line interface, the PSCAD/EMTDC palette in Fig. 2 has a MATHCAD call that can be invoked by clicking on the appropriate icon (process.mcd). When opened, the MATHCAD program takes the computed results from the PSCAD/EMTDC output file and automatically calculates the rise time of the motor speed. Here the rise time is defined as the time at which the motor reaches within 90% of its final speed. A fragment of the resulting MATHCAD window is shown in Fig. 5. The settling time appears here as 0.367 s. If the run is repeated again with different data, clicking on the appropriate icon will automatically generate a new settling time without further user interaction.

Similarly other post-processing functions can be carried out as such as a calculation of rotor acceleration as seen in the portion of the MATHCAD Screen in Fig. 6. The rotor acceleration is calculated by differentiating the speed after suitable spline fitting. Again, a single click on the PSCAD palette produces this output at the end of the simulation run. The accelerating torque is equal to the electrical torque minus friction/windage losses and mechanical torque. The

mechanical torque is constant and the losses are relatively small, hence the rotor acceleration curve (Fig. 6), which is the accelerating torque divided by the moment of inertia, has nearly the same shape as the electrical torque curve of Fig. 4.
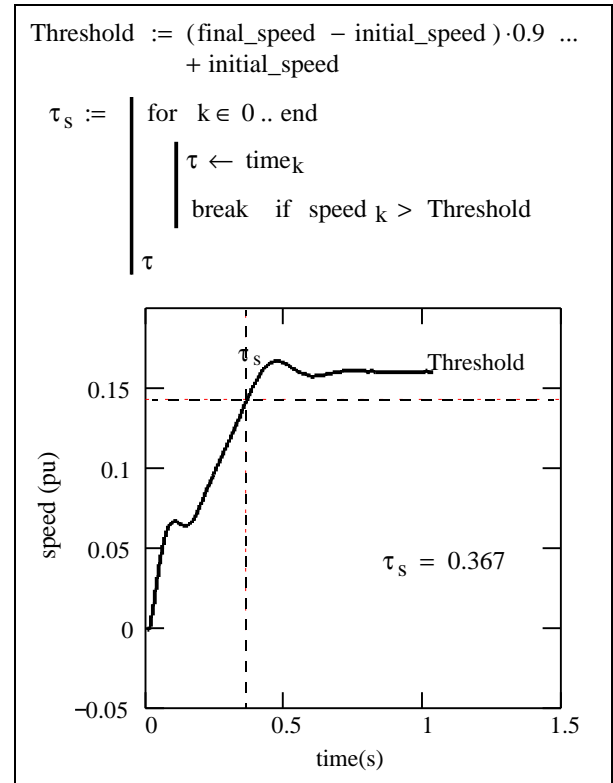

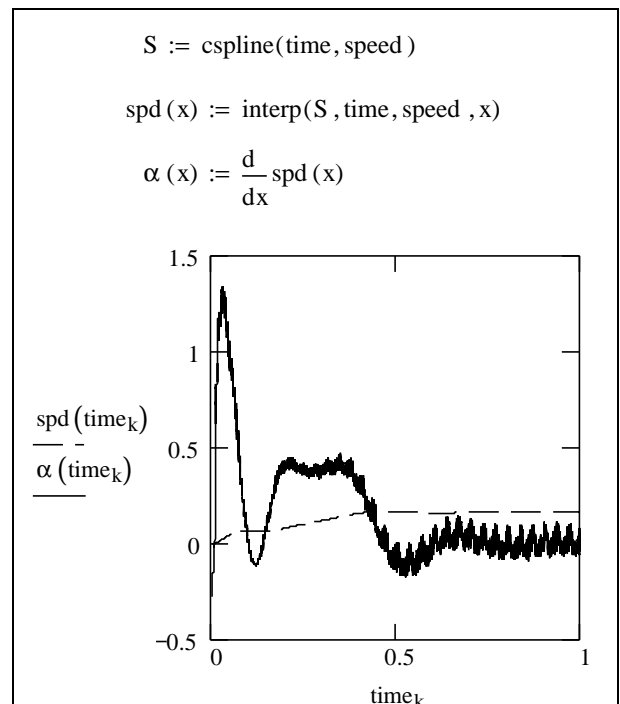Fig. 5. Fragment of MATHCAD Screen Showing Calculation of Settling Time

Fig. 6. Calculation of Rotor Acceleration

4

Although the above example illustrated the use of MATHCAD, the off-line interfacing technique can be applied to essentially any software that can write to a file. The authors have successfully made such links to spreadsheet programs, plotting packages and word processing packages.

V. CONCLUSIONS

An approach to interfacing other analysis packages with an electromagnetic transients simulation program was presented. Two types of interfaces are possible, the on-line interface which interacts with the simulation as it progresses and the off-line interface which can be invoked directly from the simulation package but only exchanges information at the beginning or end of a run. The on-line interface to SIMULINK described here makes the powerful modelling features of that program available within the electromagnetic transient simulation environment. However, there is a price to pay in terms of simulation speed. This is due to several reasons. Firstly both programs run simultaneously on the computer and thereby have to share the available memory and CPU resources. Secondly, the trapezoidal fixed-step algorithm of the emtp-type program[8] outperforms state variable based algorithms available in SIMULINK. Therefore, it is recommended that the hybrid approach be used for initial investigations, but that a model for the controller be developed within the emtp-type environment if lengthy studies are required.

Off-line interfaces provide a valuable resource for data pre and post processing and graphical presentation. By having them started from within the simulation environment, better control is maintained on the simulation project. This would not be the case if these programs were started and run independently. Thus the GUI based datafile of the simulation program maintains links to all necessary off-line programs.

VI. REFERENCES

[1] A.M. Gole, O.B. Nayak. T.S. Sidhu, M.S. Sachdev, "A Graphical Electromagnetic Simulation Laboratory for Power Systems Engineering Programs", IEEE PES Summer Meeting, Portland, OR, July 1995.

[2] Mahseredjian, J. and Alvarado, F.; "The Design of Time Domain Simulation Tools: the Computational Engine Approach" Proceedings of the International Conference on Power System Transients, Lisbon, Portugal, Sept. 3-7, 1995, pp 493-498

[3] The Mathworks Inc., "MATLAB, The Language for Technical Computing", The Mathworks Inc., September 1998.

[4] Kezunovic, M., Chen, Q.," A Novel Approach for Interactive Protection System Simulation", IEEE T&D Conference, Los Angeles, September 1996

[5] A.M. Gole, A. Daneshpooy, "Towards Open Systems: A PSCAD/EMTDC to MATLAB Interface", International Conference on Power System Transients, IPST'97, Seattle, June 22-26, 1997, Proceedings: pp 145-149.

[6] The Mathworks Inc., "SIMULINK Dynamic Simulation for MATLAB", The Mathworks Inc., December 1996

[7] MATHSOFT Inc., MATHCAD 8 Users Guide, August 1998

[8] 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