A CO-SIMULATION APPROACH FOR ACSL-BASED MODELS

W. McKay, A. Monti, E. Santi and R. Dougal
Department of Electrical Engineering
University of South Carolina
Columbia, SC 29208 (USA)
e-mail: [mckay, monti, esanti, dougal]@engr.sc.edu

KEYWORDS
Simulators, Combined Simulation

ABSTRACT

This paper describes a co-simulation approach for complex system modeling. Different kinds of coupling are considered according to the VHDL-AMS standard. The problem is then analyzed in detail by means of examples for the case of a combined ACSL and VTB simulation (Dougal et al. 2001) (Gokdere et al 2000).

INTRODUCTION

This paper focuses on a method to interface a circuit-type simulator with a signal-flow-type simulator, both in the case of signal coupling and nature coupling (power conservation law).

It is assumed that the circuit simulator uses the resistive companion method (Chua and Lin 1975). Examples of simulators of this kind are Spice and the Virtual Test Bed (VTB), a simulation environment developed at the University of South Carolina. The signal-flow simulator uses the state-space approach to describe a system.

These two methods of describing dynamical systems are equivalent in the sense that any system can be described using both approaches. However the two descriptions are not compatible and some sort of interface has to be used if one wants to connect them together. Moreover, two different types of interface can be identified and will be referred to in the paper as signal coupling and nature coupling following the terminology of the VHDL-AMS standard.

The examples in the paper will focus on the interface of a multi-language environment called Virtual Test Bed (VTB) with the Advanced Continuous Simulation Language (ACSL). VTB is a circuit simulator and it uses the resistive companion approach to solve electrical networks. ACSL is a signal-flow simulator and it uses various user-selectable integration algorithms.

THE VTB PROJECT

The Virtual Test Bed project is dedicated to developing a new environment for simulation and virtual prototyping of power electronic systems. Within the context of "virtual prototyping" we include not only simulation of system dynamics, but also solid modeling of the system and visualization of the system dynamics. One of our challenges is to fully accommodate the breadth of disciplines that power electronics encompasses, including analog electronics, digital electronics, power systems, controls, electro-mechanics, and mechanical systems. We addressed these challenges by supporting:

Multiformalism: Different languages can be used to build models of the different components that make up a system. This allows an individual to build models using the language preferred within his or her discipline. In particular, in this paper we focus on the integration of dynamic models defined by ACSL, clarifying also how the same approach can be applied to other signal-flow-type simulators.

Highly interactive environment: Users can change the system topology or parameters while a simulation executes. This allows the user to rapidly investigate interactions between components or to explore the influence of design parameters on system performance.

High-level visualization: Visualization models can be easily created and linked to live simulation data. Visualization aids the user to rapidly comprehend the system performance. Visual outputs include data-driven animation of the motion of solid objects, imposition on top of the solid objects of novel representations of abstract simulation data, or simply oscilloscope-like graphs.

Distributed computation: Large or complicated systems require that the computational load be distributed across a network in order to achieve speeds suitable for interactive explorations. This naturally supports hardware-in-the-loop operations.
RESISTIVE COMPANION METHOD

Some remarks about resistive companion method (RCM) can help the reader appreciate the integration approach. The RCM is the basis of the VTB solver. Each device is represented by a set of discretized algebraic-differential equations. A major advantage of the resistive companion method over state-based solvers (ACSL, MATLAB etc) is that once the device models are constructed, any set of such interconnected devices can be easily handled, thus yielding a technique which is appropriate for large system studies using an object oriented implementation.

The RCM requires that device model equations be expressed in a specific form that allows for handling of device interconnections. Each device can be considered as a “box” with a number of terminals that are used to provide connections to other devices.

Each terminal is associated with one across and one through variable (voltage and current in the case of an electrical terminal). Each device is connected to neighboring devices via these terminals. A set of two or more terminals connected together form a node. Terminals of any physical type are supported, provided that they satisfy energy conservation equations in the form of across and through variables associated with each terminal. Of course, only terminals of the same physical type should be connected to the same node.

By applying discretization, the RCM transforms a dynamic network into a DC network. The solution of such a network represents the network variable at the next time step. The solution of the DC network is usually performed by means of Nodal Analysis. In order to assure better flexibility Modified Nodal Analysis (MNA) approach has been implemented (Chua and Lin 1975).

SIGNAL FLOW METHOD

A signal flow method (SFM) simulator allows integration in the time domain of systems of nonlinear differential equations of the form

$$
\dot{x}(t) = f[x(t), u(t), t]\quad (1)
$$

$$
y(t) = g[x(t), u(t), t],\quad (2)
$$

where $x(t)$, $y(t)$ and $u(t)$ are the state vector, the output vector and the input vector respectively. These systems can be visualized using block diagrams, as shown in Figure 1.

Figure 1: a circuit described using the block diagram approach.

Notice that in general an electrical network can be described and simulated using either the resistive companion method or the signal flow method. The choice of one method over the other is usually determined by convenience or by personal preferences of the person performing the simulation. The two representations are mathematically equivalent, but they are not directly compatible, as explained in the next section.

MODEL INTERFACES

According to the recent IEEE standard defining the VHDL-AMS language, three schemes for coupling between objects can be defined:

- Nature coupling: when interactions require enforcement of physical conservation principles.
- Signal coupling: when we have directed flow of information through objects.
- Data Coupling: when we pass data between objects.

In the rest we will focus on the first two types of coupling. They are represented in Figure 2. Notice that in the case of the resistive companion method, the interconnections between models are of the nature coupling type. In the case of the signal flow method, connections between blocks are of the signal coupling type. In the case of signal coupling the flow of information is directional and there are two types of ports, input ports and output ports.
In the case of nature coupling there is no preferential direction of flow of information and there is only one type of connection. At each nature coupling connection there are two terminal variables, voltage and current (in the case of electrical systems).

In the rest of the paper we will examine the case shown in Figure 3, where we have two systems, one simulated in an RCM solver and the other one simulated in a signal flow method (SFM) solver. The two systems are connected together, and we will consider both cases -- signal coupling and nature coupling.

![Signal Coupling and Nature Coupling](image1)

**Figure 2:** Signal coupling and nature coupling.

**Figure 3:** The connection of two solvers.

**SIGNAL COUPLING**

Let us now focus on the case of signal coupling. In this case, the system modeled in the SFM solver represents an external system interacting with the rest of the simulation model through a set of input and output ports. The signals exchanged with the RCM solver are represented as voltages (currents could be used as well). Outputs from the SFM solver are equivalent to ideal voltage sources and inputs are equivalent to infinite impedance ports.

For the sake of simplicity we will consider the case of a single-input single-output connection. The SFM system as it appears from the RCM viewpoint is shown in Figure 4. A typical example is an electrical system modeled in an RCM solver with a control modeled in an SFM solver. The input conveys a sensed voltage and the output conveys a control voltage.

![Signal Coupling: The SFM Model from the RCM Viewpoint Appears Like a Voltage-Controlled Voltage Source](image2)

**Figure 4:** Signal coupling: the SFM model from the RCM viewpoint appears like a voltage-controlled voltage source.

Let us suppose that we know the state of the RCM network at the instant k. If we suppose that the value Vin is kept constant between k and k+1 the SFM model can be solved inside the time step. As result the value of Vout at k+1 will be calculated. Substituting this value in the RCM network defined at the instant k+1, the new value for the input Vin can be calculated. For the solution of the resistive companion network Vout can be considered as an independent source and solved directly. This means that in case of linear network we do not need to calculate the conductance matrix at every step, and in case of non-linear network the conductance matrix is not sensitive to any non-linearity contained in the SFM model. This is an important concept with regard to speeding up the simulation. Any non-linearity related to the SFM model is directly solved within the SFM solver. This allows the user to select for the SFM model the most appropriate integration method and time-step without affecting the RCM solver. Any other kind of integration will require a compromise between the two solvers.

In terms of numerical approximation, the only limit is given by the constant input value. In terms of continuous-time reconstruction this means to apply a zero-order approximation instead of a first order as usually the RCM solver performs. Anyway, if we suppose to represent...
within the SFM a discrete-time controller the approximation is null. The approach can be generalized to any number of inputs and outputs.

**NATURE COUPLING**

In this case we have to enforce physical conservation laws. For the sake of simplicity we consider the case of a one-port connection between the two solvers. Looking at equations (1) and (2), there are two different cases: the case in which the input to the SFM is the voltage and the output is the current and the case in which the input is the current and the output is the voltage. In the first case the SFM model appears to the RCM solver as a voltage-controlled current source, in the second case the SFM model appears as a current-controlled voltage source. These two cases are shown in Figure 5 and Figure 6.

![Figure 5: Nature Coupling: ACSL model from the VTB viewpoint in the case of voltage-driven models.](image)

The algorithm used to interface the RCM solver with the SFM solver is analogous to the one used for signal coupling. Let us consider for example the case of the voltage-controlled current source shown in Figure 5. Let us assume the state of the RCM network at time $k$ is known. In particular voltage $V_{in}(k)$ is known. This value is sent to the SFM solver. Under the assumption that $V_{in}(k)$ remains constant between $k$ and $k+1$, the SFM solver can calculate current $I_{out}(k+1)$. The RCM solver uses this value to solve the network at time $k+1$. The same process is repeated at each step.

As in the case of signal coupling, the algorithm introduces a zero-order approximation to the output from the SFM solver which can introduce numerical instabilities.

**LATENCY INSERTION METHOD**

An extension of the previous methods is the so-called latency insertion method. It is beneficial in the case in which stability problems arise from introduction of the ideal voltage (or current) source to represent the SFM part of the system. In that case the latency method may alleviate the problem by introducing decoupling between the RCM and the SFM part of the system. The method consists of adding small reactive elements at the interface to introduce some latency between the two parts of the system.

The method is described for a generic network in (Shutt-Aine’ 2001). The main concept is that the introduction of the small delay makes the solution of each node equation independent. The same concept can be applied to co-simulation introducing a latency at the point where we want to separate the two solvers. The case of a voltage-driven model is shown in Figure 7, where capacitor $C_l$, and inductor $L_l$ are inserted at the interface.

![Figure 7: The latency insertion cell.](image)
The calculation of the output variable $y(k)$ is performed integrating the signal-flow oriented model. Using the approach described in the previous section (no latency), this value would be represented in the RCM system by an ideal source. Using the latency method, an L1-C1 cell is inserted at the interface between the RCM and the SFM models, introducing a delay (latency) that works in the direction of improving the stability for the interface. Guidelines for the selection of the capacitance and of the inductance values are also reported in [3].

**FUTURE EXTENSIONS AND ANALYSIS**

The authors are currently working on another extension of the co-simulation approach. One really promising idea is the possibility to use a linearised model of the signal-flow oriented system adapted at every step. An equivalent resistive companion circuit is built using the following information:

- value of the state variables at the step $k$
- linearized small signal model of the system using the Jacobian matrix.

This approach is possible with all the simulators able to extract the information listed above. Both ACSL and Simulink have these capabilities.

**IMPLEMENTATION**

The theory previously introduced has been applied for the interface of an ACSL model to the VTB environment. Here we introduce some details about the software structure.

ACSL compiles simulations into dynamic link libraries (DLL) when ACSL Sim or ACSL Graphic Modeller compiles simulations. These DLL files are named yoursimulation.prj. ACSL allows programmers to interface to these DLLs directly with the ACSL Open API (Application Programmable Interface). VTB interfaces to ACSL through the ACSL Open API’s C++ callback interface. The API allows the programmer to have almost total control of the compiled simulation. VTB wraps the ACSL API’s C++ callback interface in a C++ class. Pseudo code for the C++ wrapper is shown below.

```c++
class ACSLModel: public
IAcslInterfaceCallback
{
    LoadModel(const string &model_path);
    UnloadModel();
    StepModel(bool use_start = false);
    SetStopTime(double start_time);
    SetStopVariableName(const string &name);
    SetCommInterval(double interval);
    GetDouble(const string &sym);
    SetDouble(const string &sym, double val);
    SendRTCmd(const string &cmd);
    ...
}
```

A VTB model needs to provide two functions to the VTB environment: Initialize and Step. In the initialization function the DLL created by ACSL is loaded by the C++ wrapper. The step function is of much more interest. In the step function the C++ wrapper sets the communication interval and simulation stop time for the ACSL model to be the current time plus the step interval that is being used by the VTB environment. From the point of view of the ACSL model the model runs for a total of the simulation step size for all steps the model takes. At the first step the ACSL model is issued the ACSL START command; for all other steps the model is issued the CONTINUE command. The CONTINUE command is used so that the ACSL model uses the state that has been calculated up to that point in time and does not reset its state to the initial conditions. The sequence of operations for one time step is as follows. At the beginning of the stepping of the ACSL model the input values are set in the model according to values obtained from VTB, then the model is stepped as described above and finally at the end of the step the output values are sent to the VTB environment.

**EXPERIMENTAL RESULTS FOR SIGNAL COUPLING**

Let us now focus on a case where a signal coupling is performed using the interface described above. A classical situation is a feedback control system where the plant is an electrical system described using a circuit-oriented approach and the control is described using a signal-flow-oriented approach.

Figure 8 shows the VTB schematic representing a closed-loop buck-type switching
power converter. The power cell is described using VTB native models while the control system is described using ACSL format. The input to the ACSL signal flow model is the feedback voltage and the output is the converter duty-cycle calculated according to the feedback compensation law.

**Figure 8: A buck converter system.**

**Figure 9: Simulation results for the power converter example.**

To check the performance of the controller a disturbance is introduced into the source. In the simulation results are shown. The upper trace presents the input voltage while the lower trace presents the output voltage. The control behaves as expected, rejecting the input voltage variation after a short transient, confirming that the simulation link was effective.

**Experimental Results for Nature Coupling**

Regarding nature coupling both cases shown in Figure 5 and Figure 6 have been successfully tested. We will refer to very simple circuits comparing the results obtained in a simulation completely developed in VTB with a simulation performed using the co-simulation approach with ACSL. Finally a more complex example will be also introduced to demonstrate that the applicability of this simulation approach is not limited by the complexity of the model.

Let us start with nature coupling in the case of a voltage controlled current source. The system used for the test is shown in Figure 11. The system in the upper part of Figure 11 is based on co-simulation while the system in the lower part describes exactly the same system but completely in VTB. The interface point between the two simulators is between the resistor and the inductor. This component (the inductor) determines the characteristic of the nature coupling. Since the ACSL model appears inductive at its terminals, we can structure the ACSL model as a voltage controlled current source. This will preserve the stability of the external model by keeping the inductor current as a state variable. The main results are shown in Figure 12. In the upper part, the results of the co-simulation are shown while in the lower part the results of both the cosimulation and the homogeneous simulation are superimposed. The two traces are practically coincident having a percent error lower than 0.1% at any instant.

**Figure 11: Nature coupling example for voltage controlled current source.**
The dual case of an ACSL model having a capacitive nature is shown in Figure 13. In this situation the main state variable driving the connection is the voltage across the capacitor. Applying the same logic described in the previous example we now need to describe the external solver model as a current controlled voltage source. This is in effect the model adopted for the co-simulation system shown in the upper part of the figure. As for the previous case, the results of the co-simulation and the results of the same system realized entirely in VTB are virtually indistinguishable, as the plots in Figure 14 demonstrate. Also in this case the upper part reports the results of the combined simulation while the lower scope reports the results of the two simulations superimposed.

The previous cases can be considered as reference examples but still as trivial cases. Next the interface was tested for more complex systems. Figure 15 shows the connection between a ramp voltage source and a zonal DC power distribution system described by some 100 state variables. The ACSL model was imported from a previous project without introducing any changes.

The VTB ramp block operates as an external source to provide power to the DC zonal system. A comparison between a full ACSL simulation and a combined ACSL-VTB simulation is reported in Figure 16, where we plot the discrepancy in the input current scaled to the current steady-state value. It can be seen that this relative error is always less than 0.2%.
CONCLUSIONS

The paper describes in detail an interface logic to couple circuit-based and signal-flow-oriented simulators. By means of a set of examples the authors describe the performance of the approach for signal and nature coupling.

The authors are now applying this kind of interface to an advanced simulation project modeling the US-Navy Icebreaker Healy. Furthermore, a theoretical analysis is under development to formally check the stability and accuracy of the co-simulation approach here described.

ACKNOWLEDGMENT

This work was supported by the US Office of Naval Research (ONR) under grant N00014-00-1-0131.

BIBLIOGRAPHY


BIOGRAPHIES

**William R. McKay** is a graduate student in Electrical Engineering at the University of South Carolina. He received the B.S. in Electrical Engineering from the same institution in 1999.

**Antonello Monti** is an associate professor of Electrical Engineering at the University of South Carolina. He received his PhD from Politecnico di Milano in 1994. Dr. Monti research focuses on simulation environments, power electronics and digital control.

**Enrico Santi** is an Assistant Professor of Electrical Engineering at University of South Carolina. He received the Dr. Ing. degree in electrical engineering from the University of Padua, Italy in 1988 and the M.S. and Ph.D. degree from Caltech in 1989 and 1994, respectively. His research interests include switching-mode power converters, advanced simulation of power systems, and modeling of magnetics.

**Roger A. Dougal** is a professor of Electrical Engineering at the University of South Carolina and director of the Virtual Test Bed project. He received his PhD from Texas Tech University in 1983. Dr. Dougal’s research focuses on simulation environments for multitechnical systems, with emphasis on power electronics and electrochemical power sources.