A MULTILANGUAGE ENVIRONMENT FOR INTERACTIVE SIMULATION AND DEVELOPMENT OF CONTROLS FOR POWER ELECTRONICS

ABSTRACT

Virtual prototyping of complex systems presents interesting challenges, especially with regard to systems in which different languages are used to represent different parts. We describe here one approach to solving such a problem. In particular, we describe how to integrate the Virtual Test Bed (VTB) solver engine with the Simulink solver. This produces a rich environment for virtual prototyping of power electronic applications due to the inherent mixture of circuital and control problems. The integration is conducted within the context of the resistive companion approach. We present here the theoretical foundation of the approach, and also suggest the generality and extensiblity to other solver engines such as SPICE. The theory is then enriched with some examples that illustrate the approach including the use of the VTB high level graphic user interface.

1. INTRODUCTION

Simulation of complex systems where many components interact presents peculiar challenges. Consider that a particular system might be analyzed differently by different users, each one of them focusing on a different aspect of the system performance and each one having a different metric for what is important. Consider that the complexity of the system may bridge several areas of technical expertise, and that users in each of those technical areas traditionally work with their own set of design and simulation tools. Consider also that some parts of the system may be already available while other parts are still being designed. To the extent possible, one may wish to substitute real components for their models every time that a new component becomes available. Such an approach keeps the design/simulation "alive", promotes iteration, allows an opportunity to validate models, and introduces at the earliest possible time potentially complicating nuances that were not originally accounted for in the component models. The cost of this approach is that it requires a sophisticated capability for working with Hardware In the Loop (HIL). All of these considerations suggest the desirability of a new high-level interface that allows many types of users to be comfortable with the virtual prototyping tool. An attempt to develop such a tool has been underway at the University of South Carolina for several years now under the program name Virtual Test Bed.

The work described here applies specifically to the creation of an interactive link between VTB and Matlab/Simulink that is especially valuable during the process of designing control algorithms for power electronic systems where usually a power cell interacts with a control structure. While the native VTB represents a circuit based approach that is useful for the power cell modeling, Simulink is better suited for design of the controls. The approach described here solves, in one way, the traditional dichotomy in modeling that universally plagues power electronic designers, allowing them now to use a proper instrument for each part of the system design problem.

2. 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 have addressed these challenges by choosing to support:

Multiformalism: Different languages can be use 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 Matlab/Simulink, clarifying also how the same approach can be applied for SPICE models.

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 of the system 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

3. RESISTIVE COMPANION METHOD

Some remarks about resistive companion method (RCM) can help the reader to 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 which allows for handling of device interconnections. Each device can be considered as a “box” with a number of terminals which are used to provide connections to other devices.

Each terminal is associated with one across and one through variable (current and voltage 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.

Applying the discretized approach, 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 performance a Modified Nodal Analysis (MNA) approach has been considered.

4. INTEGRATING DIFFERENT SOLVER ENGINES

Let us now focus on the specific problem addressed in this paper. Consider a specific network problem where two different solvers are applied. For the sake of simplicity we can suppose that the link between the two is well described by a two port connection.

The situation is well represented in Figure 1.

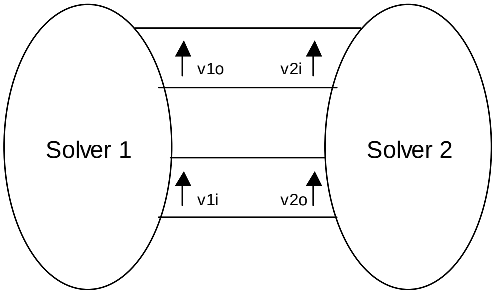

Figure 1: The solver connection

Now suppose that the input ports are infinite impedance ports and the output ports are zero impedance ports. This assumption is reasonable when the interconnection is between a circuit and a controller: an input port for the Simulink model will generally convey a measured voltage while the output will usually be to a controller. In this case, from the point of view of the VTB solver, a new two-port element can be defined:

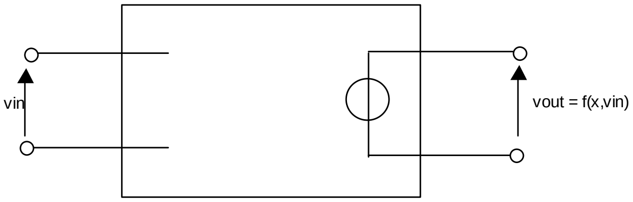

Figure 2: The Simulink solver as Voltage-controlled Voltage-Source

Let us suppose that we know the state of the network at the instant k. If we suppose that the value Vin is kept constant between k and k+1 the Simulink model can be solved inside the time step. As result the Vout at k+1 will be calculated.

Substituting this value in the resistive companion network defined at the instant k, 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 Simulink model. This is an important aspect to speed up the simulation. Any non-linearity related to the Simulink model is directly solved within the Simulink solver. This allows the user to select for the Simulink model the most appropriate integration method and time-step without affecting VTB. 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 VTB performs. Anyway, if we suppose to represent within Simulink a discrete-time controller the approximation is null. The approach can be generalized to any number of inputs and outputs.

5. INTEGRATING SIMULINK SOLVER

The heart of the communication protocol with the Simulink solver is the Matlab engine. The Matlab engine allows data exchange between Matlab and C/C++ custom software. The interface is realized by a dedicated class inside the VTB architecture. This class manages the following procedure:

- At simulation start the VTB-class calls the Matlab engine
- The VTB calls a null-time execution of the target Simulink model. The result of this step is the calculation of the Simulink model size in terms of state variables, input and output number.
- At every step the VTB calls a step execution of the Simulink model giving as input the required data and receiving back the data for the next step
- The output is inserted in the input vector of the RCM

This data exchange is implemented by the authors via a dynamically linked library object and new Matlab commands that run, through the Matlab engine, a single simulation step of the target Simulink model. A great advantage of this solution from the user viewpoint is the possibility to interact directly during the simulation both with the VTB Schematic Editor and the Simulink user interface. Parameters can be changed on the fly in both the environments thereby speeding up any testing activity.

6. A FIRST EXAMPLE: A VOLT/HERTZ CONTROL OF AN ASYNCHRONOUS MOTOR

The first example is a simple V/Hz control for an induction motor. As per the theory described previously, the power plant is described using VTB native models while the control is described by means of a Simulink model. Figure 3 shows the VTB model directly from the schematic editor screen. (Note that the drag and inertia of the mechanical load are represented here by their electrical analogs. The resistor and capacitor connected to the motor shaft are not actually parts of the power electronics nor of the controls.)

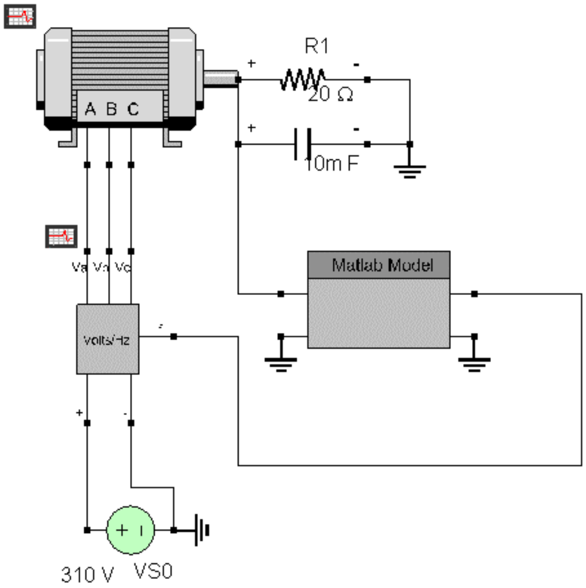

Figure 3: The VTB model used as example

The “Matlab Model” block represents the two-port interconnection previously introduced. The user is provided with the following options:

- Simulink file name
- Number of inputs
- Number of outputs
- Differential or absolute input
- Differential or absolute output

The Simulink model of the control algorithm is shown in Figure 4.

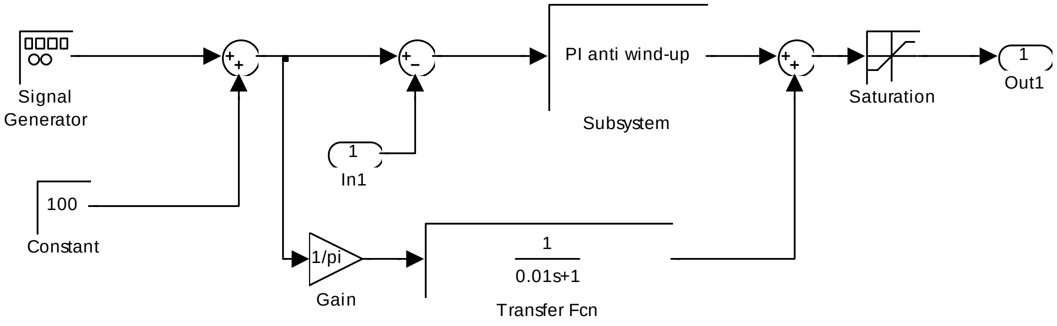

Figure 4: The Simulink model for the V/Hz control

While running the simulation, the VTB provides a interactive visualizations via the Visualization eXtension Engine (VXE). Figure 5 shows a running graph of the motor speed. In this case the speed is controlled to follow a reference signal in the form of a square-wave with constant offset. The transient dynamics are apparent.

Picked: none Independent: time Triggering: disabled Frame rate: 0.1 Hz

Figure 5: The motor speed during the simulation

Figure 6, shows an example of the corresponding high level visualization: the 3D animation show both motion of the rotor and the temporal-spatial distribution of the airgap magnetic flux

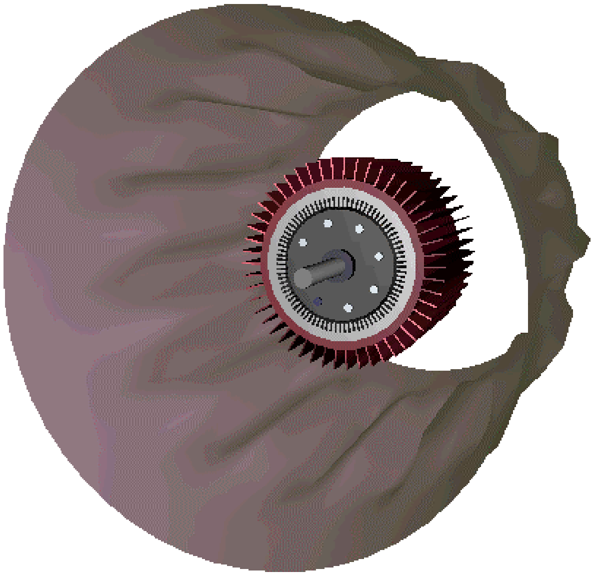

Figure 6: Dynamic visualization showing rotor motion and magnetic flux distribution in the air gap

7. ADDING INTERACTIVITY: THE STEERING CONTROL

The previous example is a simple introduction to what can be performed thanks to the interface described here.

In the final paper more detail will be added about a more interactive example under development: steering control of an electric mining vehicle. Figure 7 shows a simplified schematic of the electric power system as shown in the Schematic Editor.

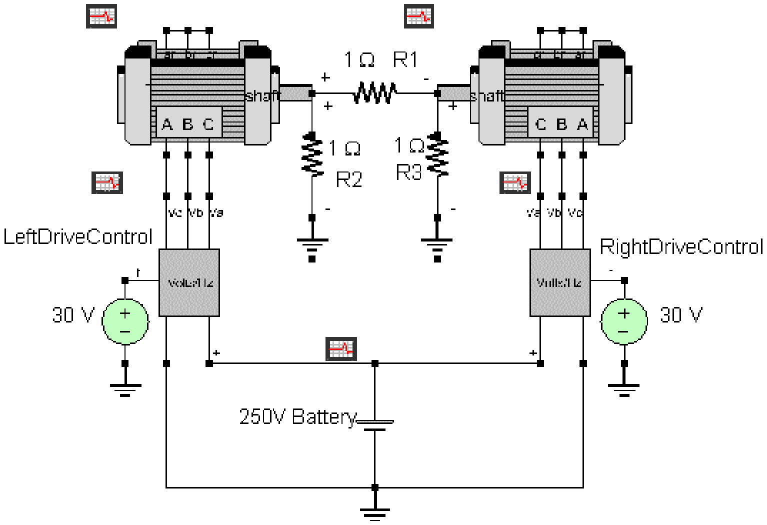

Figure 7: The motor system

The main advantage of the model under construction, from the user viewpoint, is the possibility to interact directly with 3D objects instead of fixing on a typical simulation interface: in a few words, the user will be able to literally drive the simulation model around. This should allow to test the control system with various levels of sensitivity, taking into account the perspective of the final user.

8. CONCLUSIONS

A new integrated platform for simulation has been described. It is an interesting evolution of the VTB platform. The new approach for simultaneous simulation allows the user to work within the proper environment to build the proper model. A high level of user interactivity is guaranteed both by the VTB solver engine and the Simulink interface. Currently, the authors are working on an extension of this approach to incorporate SPICE models.

We emphasize that this solution guarantees the following advantages:

- The user does not need any compiler to run the simulation. Either part of the system model -- electrical circuit or control system -- can be changed on the fly
- Each model is solved within its native engine, obtaining the maximum performance from each part
- The integration of environments is performed without any model translation or interpretation
- The Matlab link is performed by means of a compiled DLL, limiting the impact on the simulation speed
- No restrictions are applied to the Simulink model, so that any special Matlab toolbox can interact with the model.