

5-1-2003

## Rapid Prototyping of Digital Controls for Power Electronics

Antonello Monti

*University of South Carolina - Columbia*, [monti@enr.sc.edu](mailto:monti@enr.sc.edu)

Enrico Santi

*University of South Carolina - Columbia*, [esanti00@enr.sc.edu](mailto:esanti00@enr.sc.edu)

Roger A. Dougal

*University of South Carolina - Columbia*, [dougal@enr.sc.edu](mailto:dougal@enr.sc.edu)

Marco Riva

*University of Milan*, [marco.riva@mi.infn.it](mailto:marco.riva@mi.infn.it)

Follow this and additional works at: [https://scholarcommons.sc.edu/elct\\_facpub](https://scholarcommons.sc.edu/elct_facpub)



Part of the [Electrical and Computer Engineering Commons](#)

---

### Publication Info

Published in *IEEE Transactions on Power Electronics*, Volume 18, 2003, pages 915-923.

<http://ieeexplore.ieee.org/xpl/RecentIssue.jsp?punumber=63>

© 2003 by IEEE

This Article is brought to you by the Electrical Engineering, Department of at Scholar Commons. It has been accepted for inclusion in Faculty Publications by an authorized administrator of Scholar Commons. For more information, please contact [digres@mailbox.sc.edu](mailto:digres@mailbox.sc.edu).

# Rapid Prototyping of Digital Controls for Power Electronics

Antonello Monti, *Senior Member, IEEE*, Enrico Santi, *Senior Member, IEEE*, Roger A. Dougal, *Senior Member, IEEE*, and Marco Riva, *Member, IEEE*

**Abstract**—The process for designing digital controls for power electronics is typically quite convoluted and affords many opportunities for errors to occur. We present here a new and complete method for rapid prototyping of digital controls that allows rapid realization of new designs. The approach uses a collection of tools that include both software (the virtual test bed (VTB) and Matlab/Simulink) and hardware (dSpace DSP). An example application of the methodology completes the discussion.

**Index Terms**—Digital controls, dSpace DSP, power electronics, rapid prototyping, virtual test bed.

## I. INTRODUCTION

DIGITAL technology has greatly impacted the design of controls for power electronics. In the beginning, when only low-performance processors were available, writing code for controllers was largely an art form, without a well-defined and standardized design procedure. With the advent of digital signal processors (DSP) and high-performance processors in general, the design approach has significantly changed. The availability of high-level languages has allowed the introduction of software design methodologies and testing procedures that have both boosted the reliability of designs and shortened the design cycle.

Electrical drives, and power electronics in general, demand an intimate link between the controller (control software) which operates in the discrete domain, and the controlled sub-system (the plant) which operates in the analog domain. These systems, described through a control loop such as that presented in Fig. 1, are called *hybrid* systems [1] to emphasize the co-operation between the analog systems (the plant) and the discrete systems (the control software).

Discrete and analog parts of the system interact intimately, yet these two parts are normally modeled separately and the whole system is not generally tested in a global sense before actual construction. Aggravating this situation is the fact that the real-time control software faces critical time constraints that are difficult to test before the final implementation. These circumstances beg for definition of a formal procedure for system specification, which we intend to introduce here.

Manuscript received February 25, 2002; revised November 1, 2002. This work was supported by the U.S. Office of Naval Research (ONR) under Grant N00014-00-1-0131. Recommended by Associate Editor S. B. Leeb.

A. Monti, E. Santi, and R. Dougal are with the Department of Electrical Engineering, University of South Carolina, Columbia, SC 29208 USA (e-mail: monti@enr.sc.edu; esanti@enr.sc.edu; dougal@enr.sc.edu).

M. Riva is with the Department of Physics, University of Milan, Milan 20133, Italy (e-mail: marco.riva@mi.infn.it).

Digital Object Identifier 10.1109/TPEL.2003.810864

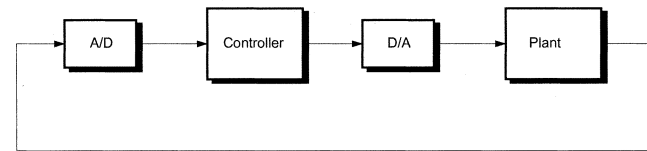


Fig. 1. Control system structure.

In industrial practice, informal methods that are well-known and simple are often preferred to formal methods. However, informal methods do not support the integrated design of the analog and discrete parts and so provide no means for ensuring that time constraints are honored.

While formal methods are clearly better, they are often not used because of the difficulties of modeling the various components (both analog and discrete). Two key points can be identified and should be solved:

- 1) it is desirable to increase the use of formal methodologies for software specification;
- 2) it is desirable to increase the use of software for simulation and testing.

A complete and integrated environment is required to support a designer throughout the development of a drive system, from the initial design phase until the final steps of code generation. Major requirements of this environment are: a complete set of standard functions, ease of operation especially by means of a simple graphical interface, tools to assist in production of documentation, and a short training period for operators.

While software tools that support various parts of the process are commercially available, no tool provides a comprehensive environment. Thus, it is necessary to use several different tools that are each dedicated to different phases of the drive design. Unfortunately, then, the designer has to cope with a lack of compliance (or interoperability) between the various environments that he has to use.

While this is a critical issue in power electronics, very few papers address the problem. Reference [2] does describe the problem in detail and suggests a complete formal design procedure that allows generation of code and simulation of the plant together with the control software. But the emphasis is on developing the controller software; the approach for plant modeling does not provide an easy route for description of circuit-oriented problems.

A method for developing software for drives applications is proposed in [3]. The proposed approach supports automatic

code generation but the problem of testing before final implementation is not discussed in detail. Another interesting discussion of the problem is also reported in [4].

On the other hand, many large companies have invested heavily to develop custom solutions for the problem. They usually prefer to develop an internal standard for software development that is applied to development of their own products. This approach often hinders the development of complex systems, though, because more than one company is often involved. In this case it is desirable to be able to export the description of the control in detail while maintaining the required level of security for proprietary information.

In this paper, we present an integrated approach that solves most of these problems thus giving the designer a fully-integrated environment.

The main features of this environment are:

- 1) different languages can be used to describe different parts of the system. In particular, Simulink block diagrams can be used to define the control structures;
- 2) if necessary, proprietary information can be secured;
- 3) the system simulation can be executed even if not all of the models are available;
- 4) controller code can be generated automatically.

The kernel of this approach is the virtual test bed (VTB) environment, which is described in the following section.

## II. VTB ENVIRONMENT

The purpose of the VTB project is to develop 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 the challenges is to fully accommodate the breadth of disciplines that power electronics encompasses, including analog and digital electronics, power systems, controls, electro-mechanical, and thermal systems. Simulation of such complex systems where many components interact presents peculiar challenges. Consider that a particular system might be analyzed differently by different persons, 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 engineers in each of those technical areas traditionally work with their own set of design and simulation tools.

Consider that suppliers of a particular subsystem may have already invested significant effort in creating a simulation model for that subsystem, which encapsulates their in-depth knowledge of the system. Porting the model to a different simulator that may be chosen for system integration is difficult, time-consuming and represents a duplication of effort.

All of these considerations suggest the desirability of a 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 VTB [5], [6].

The VTB approach described in this paper, solves the traditional dichotomy in modeling that universally plagues designers, allowing them to use a proper instrument for each part of the system design problem. In contrast, classical simulators, where a single language can be used to specify the system, really limit the analysis of such systems.

The VTB environment, on the other hand, addresses these challenges by choosing to support:

— **Multiformalism:** different languages can be used to build models of the different components that compose a system. This allows an individual to build models using the preferred language within his or her discipline (in this case mechanical, electrical, chemical).

— **Co-Simulation:** users can use other solvers together with the main VTB solver. This means that any part can be solved with the more appropriate integration step and method without affecting the solution of the rest of the system.

— **High-level visualization:** visualization models of the system can be easily created and linked to live simulation data. Visualization helps 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. This feature seems to be particularly interesting for the case analyzed here (an electric drive), where management of the electrical variables produces a concrete and visible mechanical effect. Furthermore, a high level visualization better supports the exchange of information among the designers cooperating on the project.

— **Hardware in the simulation loop:** A real-time extension to VTB allows performing HIL simulations.

## III. PROPOSED DESIGN APPROACH

Control designers often use simple models of the plant for two reasons. First, it allows them to focus on the relevant control problems, and second, while they are experts in control, they are not necessarily experts with respect to the particular system that is being controlled. But when working with highly nonlinear time-variant systems, such as power converters, use of simplified models can only be considered during the earliest stages of system design.

To be confident of the design, the control designer must test his controller with more refined models. An example of the various model levels that might be considered in the case of switching power converters could be:

- 1) averaged model;
- 2) switching-level model;
- 3) Hardware in the loop test (if available).

In order to perform all these tests we need a single simulation and testing environment where we can:

- 1) easily describe the digital control algorithms;
- 2) easily describe the power components and the system topology;
- 3) perform hardware in the loop testing.

The following design path can be adopted:

- 1) Define the (full fidelity) system model in VTB

- 2) Design the control using Simulink (and using, if necessary, a very rough model of the plant)
- 3) Link the control model to the system model and test the controller with the full-fidelity system model, interactively tuning the controller if necessary
- 4) Use the Matlab real-time workshop to compile the Simulink block diagram and use VTB tools to create a compiled model of the controller that can be used in VTB.
- 5) Refine the VTB model of the power plant and confirm that the controller functions as intended
- 6) Export the Simulink definition of the control-subsystem to the dSpace environment
- 7) Perform hardware in the loop simulation between VTB (plant model) and dSpace (control System)
- 8) When satisfied that the control is correct, insert it into the hardware of the real plant
- 9) Optimize the control for a low-cost platform (if the C-language can be adopted the porting is quite trivial).

The design approach is summarized in Fig. 2. In the following we will detail.

- 1) The concept of co-simulation between VTB and Simulink to clarify the interaction from the circuit theory viewpoint.
- 2) The procedure for automatically generating VTB models from Simulink block diagrams, which speeds-up creation of a complete system model in VTB.
- 3) Finally, we will describe an example application, including experimental results.

#### IV. INTERFACING VTB AND MATLAB/SIMULINK

Two different options to interface VTB and Matlab/Simulink are made available in the VTB environment. The first option is an interactive co-simulation mode, in which VTB and Matlab/Simulink run concurrently and exchange data at every simulation step. This option facilitates rapid iteration of both the discrete controller design (Simulink) and the plant design (VTB) and is appropriate in the first steps of the control design process. The designer typically performs a few design iterations, makes a first-cut design of the controller and verifies that it performs reasonably well. The second step is to compile the controller model and insert the compiled version of the controller into the VTB simulation. This yields three benefits.

- 1) As compared to the interactive co-simulation model (which runs in interpretive mode in Simulink) the compiled model dramatically increases the simulation speed.
- 2) The use of compiled models preserves the secrecy of any proprietary information that may be contained in the control design, and so allows the controller to be distributed to other project participants when more than one company is involved in the system design.
- 3) The compiled model can be freely executed in the VTB environment. It does not require a Matlab/Simulink license at run-time, which again simplifies use in a team-oriented design environment.

These two options allow the user to design the control according to the following procedure:

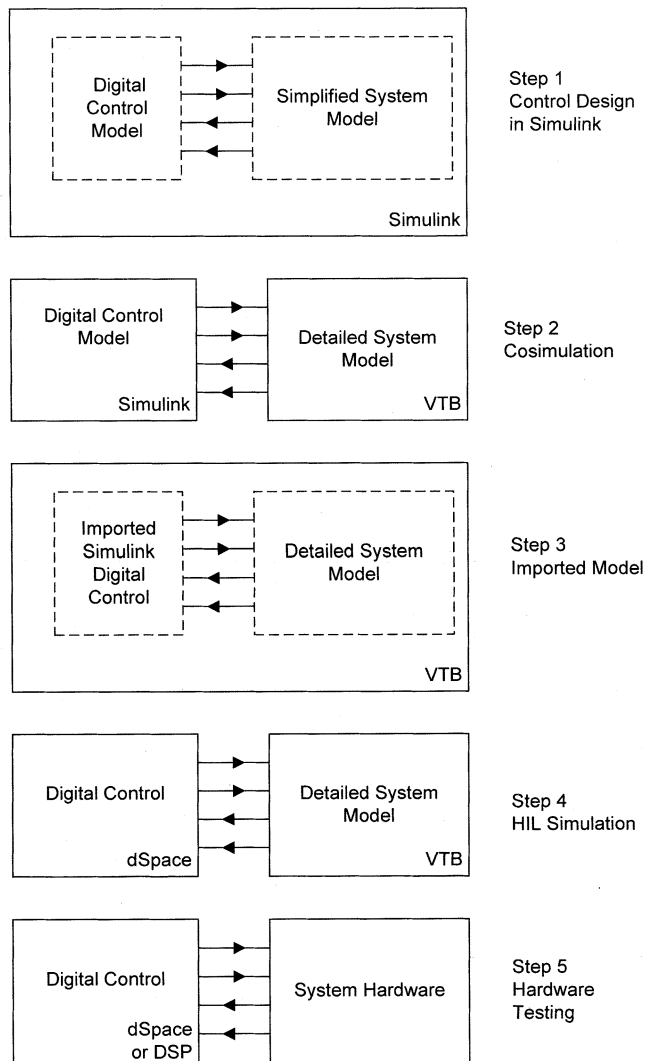


Fig. 2. Proposed design approach.

- 1) start the control design within the Simulink environment;
- 2) complete a test phase using the interactive interface with VTB;
- 3) compile the model for further analysis and testing.

#### V. CO-SIMULATION BETWEEN VTB AND MATLAB/SIMULINK

The first option to interface VTB and Simulink is the interactive co-simulation option. Co-simulation is an important element of the design process, as described in [7]. 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 solvers is well described by a two-port connection as shown in Fig. 3. In this case the solver 1 subsystem appears to solver 2 as a block with one input and one output. From the solver 2 viewpoint, the definition of input and output is clearly the opposite.

Now suppose that the input ports have infinite impedance and the output ports have zero impedance. This assumption is reasonable when the interconnection is between a circuit and a controller: an input port for the Simulink model will generally

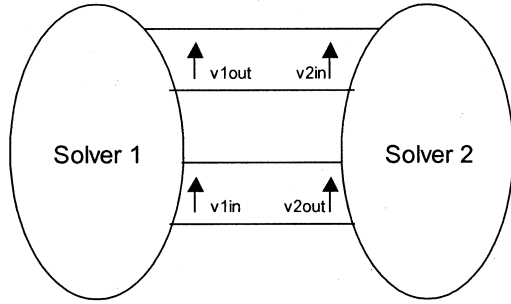


Fig. 3. Solver connection.

convey a measured voltage while the output will usually be a control signal represented by a voltage source. In this case, from the point of view of the VTB solver, a new two-port element can be defined as in Fig. 4.

In the general case a controller with  $n$  inputs and  $m$  outputs can be described from the circuit-oriented point of view of VTB in the following way.

- 1) The  $n$ -inputs represent an  $n$ -node network with zero conductance with respect to ground.
- 2) The  $m$ -outputs represent  $m$  ideal voltage sources with respect to ground.

Using modified nodal analysis and the resistive companion approach [8], the generic controller is described by an  $n+2m$  conductance matrix having

$$G_{n+m+k,k} = G_{k,n+m+k} = 1 \quad \text{for } 0 < k < m \quad (1)$$

$$G_{ij} = 0 \quad \text{elsewhere} \quad (2)$$

while for the history vector we have

$$A_i = 0 \quad 0 < i < n + m \quad (3)$$

$$A_{n+m+j} = y_j \quad 0 < j < m \quad (4)$$

where  $y_j$  represents the generic control output.

The co-simulation proceeds as follows. Let us suppose that we know the state of the electrical network at the instant  $k$ . If we suppose that the value of  $V_{in}$  is kept constant between  $k$  and  $k+1$  the Simulink model can be solved inside the time step. As a result the value of  $V_{out}$  at  $k+1$  will be calculated.

Substituting this value in the resistive companion network defined at the instant  $k+1$ , the new value for the input  $V_{in}$  can be calculated. For the solution of the resistive companion network,  $V_{out}$  can be considered as an independent source and solved directly. This means that, in case of a linear network, we do not need to calculate the conductance matrix at every time step, and in case of nonlinear network the conductance matrix is not sensitive to any nonlinearity contained in the Simulink model. This has important consequences for simulation speed. Any nonlinearity in the Simulink model is solved within the Simulink solver. This allows the user to select for the Simulink model the most appropriate numerical integration method and time-step without affecting the VTB simulation time step or algorithm. Other approaches to integration of the two-solver engines may require an iterative solution, which would affect simulation performance.

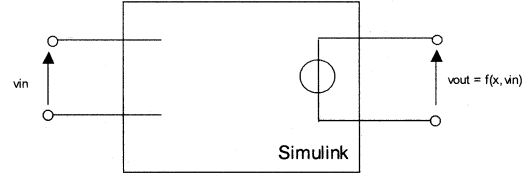


Fig. 4. Simulink model as voltage-controlled voltage-source.

The only numerical approximation made here is that related to assumption of a constant value for  $V_{in}$  during each time step—equivalent to applying a zero-order approximation instead of first order as VTB would usually do. But if the controller described by the Simulink diagram is a digital controller, then no approximation is made—the simulation model corresponds exactly to the real system. The approach can be generalized to any number of inputs and outputs.

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

- 1) At simulation start the VTB class calls the Matlab engine
- 2) The VTB calls a null-time execution of the target Simulink model. The result of this step is a calculation of the Simulink model size in terms of the numbers of state variables and input and output variables.
- 3) 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.
- 4) The output from Simulink is inserted into the input vector of the resistive companion model.

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

Furthermore, thanks to the new features of the VTB-2002 version, it is now possible to link the Simulink schematic to the Signal backplane [6]. In this case, according to the VHDL-AMS standard [9], the connection between the two solvers follows the signal coupling logic instead of the one of nature coupling. This simplifies the communication by removing the conservation-of-energy constraint (typical of nature coupling), thus requiring the calculation of the terminal voltage only in a signal-flow-like approach [10].

Besides the co-simulation approach, Simulink controller can be embedded in a VTB model in compiled form. VTB tools have been developed to automatically link the code generated by the Simulink Real-Time Workshop to a VTB model class, thus creating an autonomous dll for the VTB library.

This link relies on:

- 1) customized kernel of the Simulink engine;

- 2) automatic code generation procedure to embed the Simulink solver in the VTB logic.

The code generation procedure automates the creation of the VTB model, creating the conductance matrix and history vector according to (1)–(4).

## VI. APPLICATION EXAMPLE: STATE SPACE CONTROL OF A BOOST CONVERTER

Now let us focus on an interesting example: the control of a boost converter using a state feedback digital controller.

We use this example to illustrate the approach through a full design process. The process clearly shows the trade-off between model complexity and accuracy, and shows how, at different points during design and implementation, it is desirable to transition from simpler models to more complex and accurate models, which correctly predict observed second-order effects.

The first step is an analytical control design using pole placement on the linearized system. This gives an analytical closed-loop model, which represents the ideal closed loop model of the system. The second step is VTB cosimulation using an averaged converter model to test the design and fine-tune the controller, if needed. After this, the Simulink model is compiled and imported into VTB. The simulation results are then compared with the analytical closed-loop model. The next step is to use a switching model for the converter, which allows to capture various second-order effects. According to the progression depicted in Fig. 2, the next step is HIL simulation. For simplicity, this step is skipped in this example, but it could be easily realized. The final step is the actual hardware implementation using the dSpace board. Experimental results are compared with simulation results obtained from the various simulations.

### A. Analytical Control Design

Let us start considering the classical time-averaged model of the boost converter

$$\begin{aligned}\dot{X}_1 &= -\frac{X_2}{L}(1-u) + \frac{V_g}{L} \\ \dot{X}_2 &= \frac{X_1}{C}(1-u) - \frac{X_2}{RC},\end{aligned}$$

Where  $X_1$  is the inductor current,  $X_2$  is the capacitor voltage and  $u$  the switch duty cycle.

Let us suppose we want to adopt a state linear feedback: a local linearization procedure is applied to the model [5]

$$\begin{bmatrix} \Delta \dot{X}_1 \\ \Delta \dot{X}_2 \end{bmatrix} = \begin{bmatrix} 0 & -\frac{(1-\bar{u})}{L} \\ \frac{(1-\bar{u})}{C} & -\frac{1}{RC} \end{bmatrix} \begin{bmatrix} \Delta X_1 \\ \Delta X_2 \end{bmatrix} + \begin{bmatrix} \frac{\bar{X}_2}{L} \\ -\frac{\bar{X}_1}{C} \end{bmatrix} \cdot [k_1 \quad k_2] \begin{bmatrix} \Delta X_1 \\ \Delta X_2 \end{bmatrix}.$$

Our objective is to obtain the control law as a function of the variation of state variables  $X_1$  and  $X_2$

$$\Delta u = k_1 \Delta X_1 + k_2 \Delta X_2.$$

The new state matrix including the feedback will be

$$\Phi_{CL} = \begin{bmatrix} \left( \frac{\bar{X}_2 k_1}{L} \right) & \left( \frac{\bar{X}_2 k_2 - (1-\bar{u})}{L} \right) \\ \left( \frac{(1-\bar{u}) - \bar{X}_1 k_1}{C} \right) & \left( -\frac{1}{RC} - \frac{\bar{X}_1 k_2}{C} \right) \end{bmatrix}.$$

The characteristic equation of  $\Phi_{CL}$  is given by

$$\lambda^2 + \left( \frac{1}{RC} + \frac{\bar{X}_1 k_2}{C} - \frac{\bar{X}_2 k_1}{L} \right) \lambda - \frac{\bar{X}_2 k_1}{RCL} + \frac{(1-u)^2}{LC} - \frac{\bar{X}_1 k_1 (1-\bar{u})}{LC} - \frac{\bar{X}_2 k_2 (1-\bar{u})}{LC}.$$

Let the desired closed-loop eigenvalues be  $P_1, P_2$ . Using the properties of second order equations, we obtain the following equations in matrix form giving feedback coefficients  $k_1, k_2$  as a function of desired eigenvalues  $P_1, P_2$

$$\begin{bmatrix} -\frac{\bar{X}_2}{L} & \frac{\bar{X}_1}{C} \\ \left( \frac{\bar{X}_2}{RCL} + \frac{(1-\bar{u})\bar{X}_1}{LC} \right) & \frac{(1-\bar{u})\bar{X}_2}{LC} \end{bmatrix} \begin{bmatrix} k_1 \\ k_2 \end{bmatrix} = \begin{bmatrix} -\left( P_1 + P_2 + \frac{1}{RC} \right) \\ P_1 P_2 - \frac{(1-\bar{u})^2}{LC} \end{bmatrix}.$$

This equation has the form

$$\begin{bmatrix} q_{11} & q_{12} \\ q_{21} & q_{22} \end{bmatrix} \begin{bmatrix} k_1 \\ k_2 \end{bmatrix} = \begin{bmatrix} m_1 \\ m_2 \end{bmatrix}$$

and solving for  $k_1, k_2$  gives

$$\begin{bmatrix} k_1 \\ k_2 \end{bmatrix} = \frac{1}{q_{11}q_{22} - q_{12}q_{21}} \begin{bmatrix} q_{11} & q_{12} \\ q_{21} & q_{22} \end{bmatrix} \begin{bmatrix} m_1 \\ m_2 \end{bmatrix}.$$

### B. VTB Cosimulation Using Averaged Converter Model

In the rest of the paper we refer to a small prototype converter with the following parameters.

- Rated Input Voltage: 12 V.
- Rated Output Voltage: 40 V.
- Maximum Load: 100 W.
- Input Inductance:  $L = 46$  mH.
- Output Filter Capacitance: 1.360 mF.
- Main Switch: IRF540N.
- Switching Frequency: 60 kHz.

The controller, shown in Fig. 5, is first implemented in Simulink for a preliminary test based on a very simplified averaged model of the system.

The controller defined in Simulink was then simulated in VTB by using first the interactive interface and then the compiled interface. The two cases are shown in Figs. 6 and 7. The first case is used to test the controller performance while the second one represents a prototype of the actual code implementation.

As shown in Fig. 5 the system starts working open loop thanks to the flag coming from input 4 applying as duty cycle the value

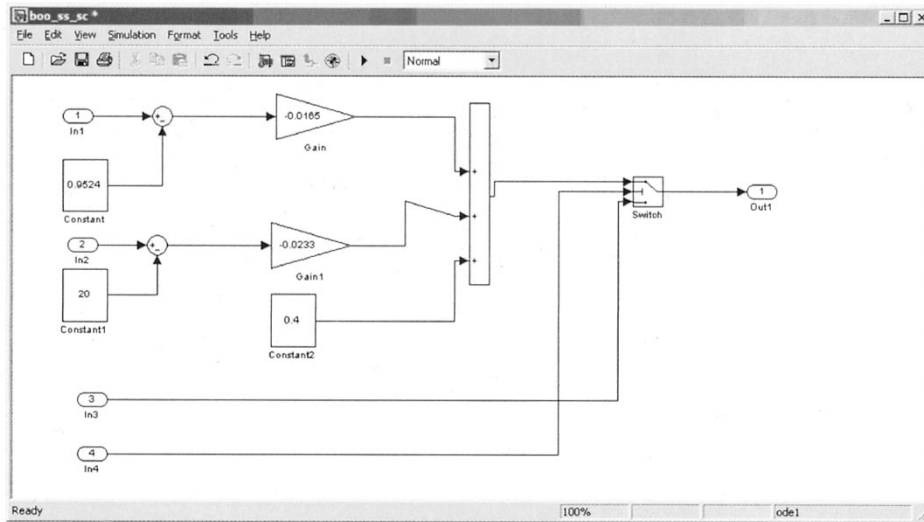


Fig. 5. State feedback controller block in Simulink.

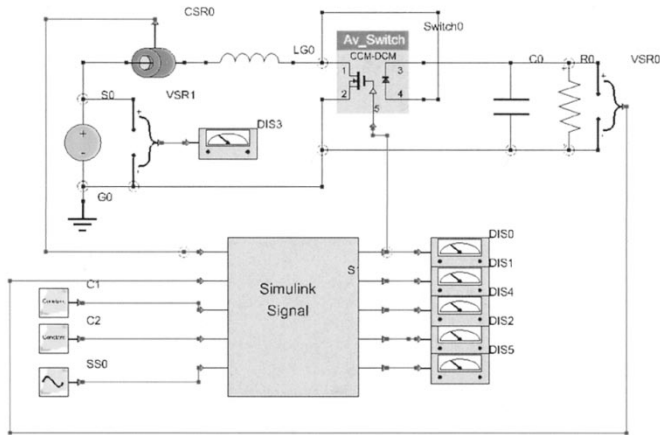


Fig. 6. VTB schematic for interactive co-simulation (with Simulink) of the plant and the controller.

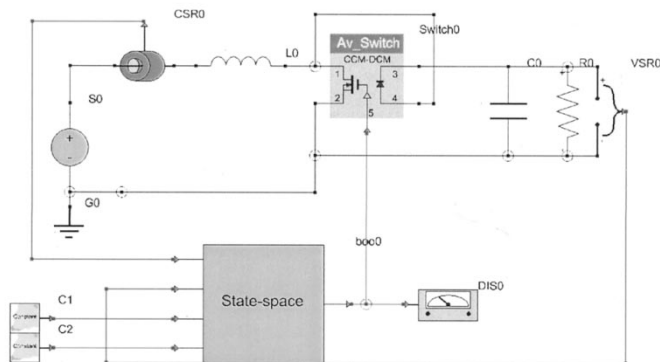


Fig. 7. VTB schematic for testing compiled version of the controller as automatically generated from the Simulink model.

on input 3. Then when the flag is toggled the closed loop control is activated. The designed steady state output voltage is 20 V.

The simulation based on the average model gives results exactly coincident with the theory so that the steady state output voltage is exactly equal to the reference. This result can be veri-

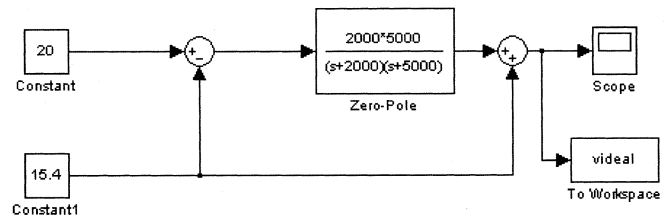


Fig. 8. Schema based on the analytical closed-loop transfer function of the system (ideal case).

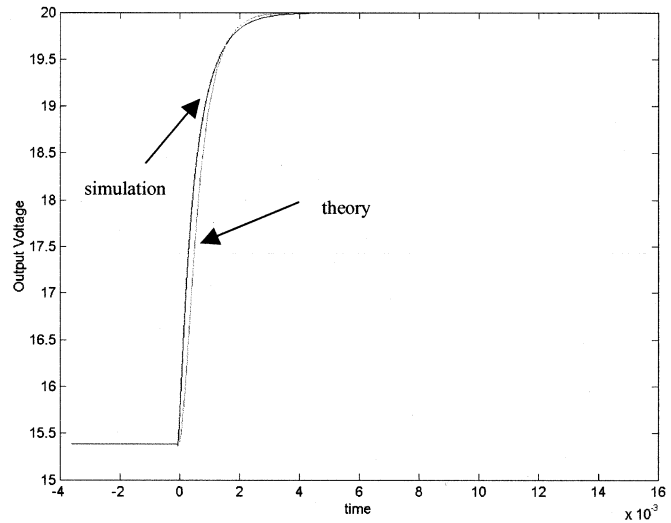


Fig. 9. Comparison between the ideal case and the averaged simulation.

fied by means of the Simulink schema in Fig. 8. This represents the ideal closed loop behavior of the system.

The closed loop system reacts to the system error, with the dynamics of the two closed loop poles defined in the design. In this case, we fixed the two poles at 2000 and 5000 rad/s.

Fig. 9 shows a comparison between the output voltage obtained by using the averaged model and the output voltage pre-

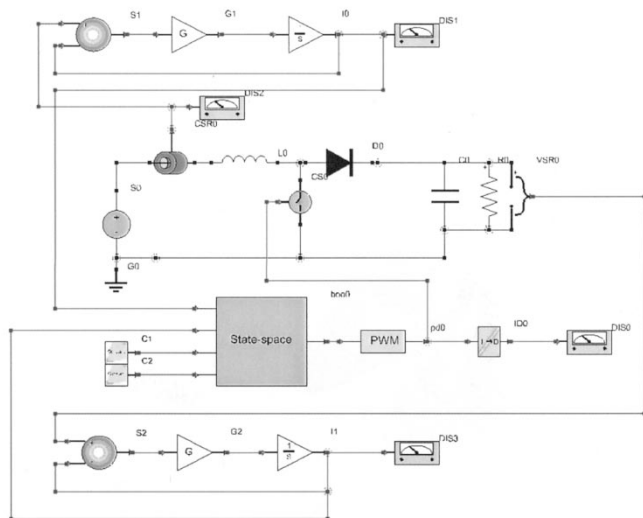


Fig. 10. VTB schema for switching-level simulation. Notice the anti-aliasing filters at the top and bottom of the figure.

dicted by the ideal simulation according to the control design specification.

The steady-state output voltage perfectly matches the design value, and the transient response is also reasonably much in agreement. The small difference can be explained by the fact that the VTB/Simulink interface introduces a zero-order-hold delay action on the step calculation. Considering the relation between the time-step ( $1 \mu s$ ) and the system bandwidth the effect is quite limited, but still observable.

*C. VTB Simulation Using Converter Switching Model and Imported Controller Model*

At this stage the design appears to be satisfactory but we can test to a greater level of accuracy by using a more sophisticated model as shown in Fig. 10.

For this more refined test some new elements are included:

- 1) switching element;
- 2) analog PWM unit (this is described by a subcircuit masked by the PWM icon in the figure);
- 3) analog filters on both the current and voltage measurements to avoid aliasing effects on the discrete-time control.

This stage constitutes the last experiment before implementation of the control algorithm using the dSpace hardware. For this reason it is significant to compare the results of the averaged simulation and of the switching model simulation with the experimental results. These comparison will increase our understanding of how the more refined model replicates effects that we will face in the real converter. In the case of the switching model we always use for comparison purposes the filtered quantities; that is also the case for comparisons with the experimental setup. For this reason no switching frequency ripple is present in the waveforms.

Fig. 11 compares the measured output voltage from the experiment with the output voltage predicted by the averaged simulation model.

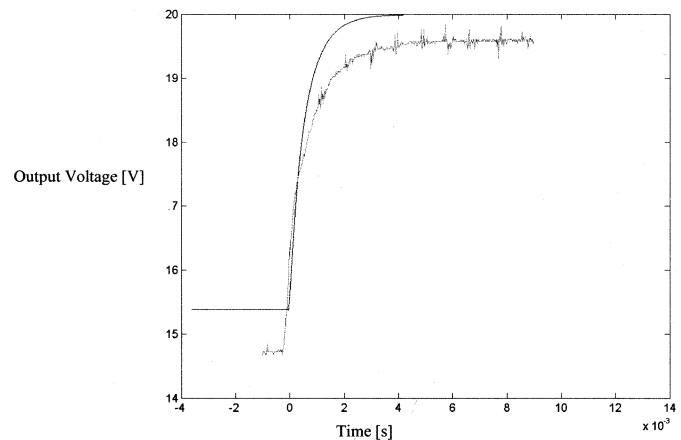


Fig. 11. Comparison between averaged simulation and experimental results (output voltage).

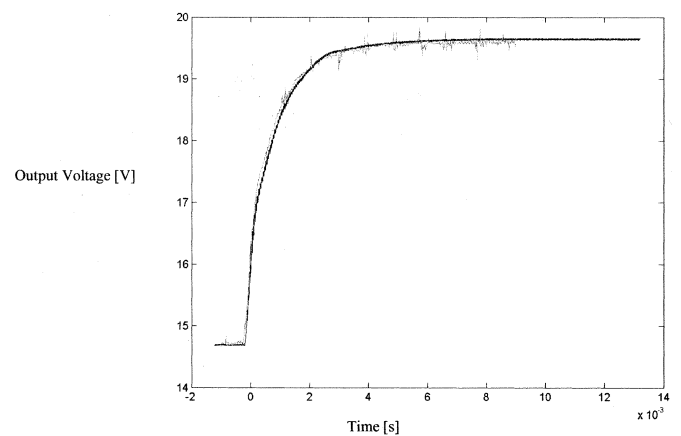


Fig. 12. Comparison between switching simulation and experimental results (output voltage).

While the dynamics of the two transients are quite similar, the two steady state conditions are quite different. This result occurred because the output diode was not accounted for in the average value model of the power converter, yet that diode produces a significant voltage drop.

This result makes sense and confirms that for low-voltage applications such as the one analyzed here, averaged models that do not include voltage drop on silicon components are too rough. In the case of control development, this kind of problem seems mostly to affect the steady-state point, while the dynamic is still reasonably correct.

In Fig. 12, the output voltage from the switching-model is compared with the experimental results. In this case the two waveforms perfectly match showing that this level of modeling is accurate for any kind of control design.

Furthermore, thanks to the approach that is the main focus of this paper, the control software that was tested in the simulation did not require any changes before being downloaded to the dSpace platform.

An analysis of the input current waveform gives other interesting insight into the quality of the simulation models at the various levels of detail. In the case of the averaged model, the



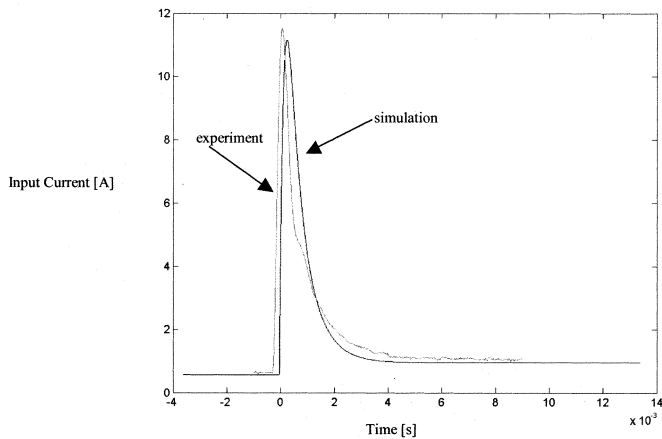


Fig. 13. Comparison between averaged simulation and experiment (input current).

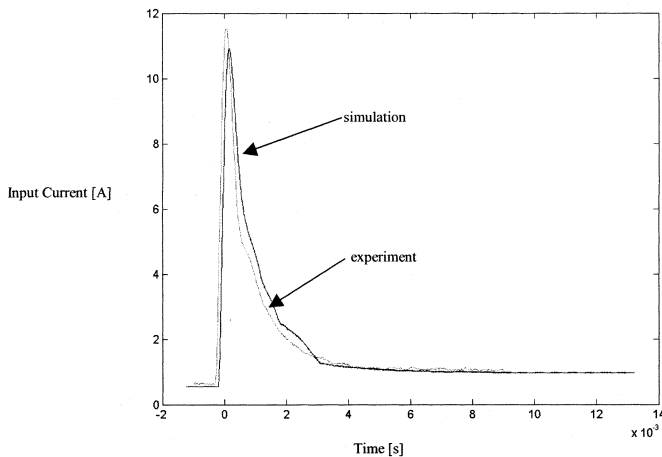


Fig. 14. Comparison between switching simulation and experiment (input current).

current waveform perfectly replicates the theoretical behavior defined by the linear combination of two exponential signals.

This seems to be perfectly consistent with the control design but it is quite different from the experimental results.

The measured input current showed some nonlinear effects that were not considered in the averaged model (see Fig. 13).

Vice-versa the same effect is also present in the switching model as shown in Fig. 14 after the first part of transient, the current shows a step variation of the derivative.

This nonlinear effect could be connected with the PWM structure adopted both in the experimental set-up and in the VTB model.

The PWM unit is an independent analog circuit that is not synchronized with the digital control (we made this choice because it was not possible to create the desired 60 kHz switching frequency directly from the digital outputs of the dSpace platform): this lack of synchronization can create a nonlinear effect related to the delay action that is the source of the nonlinearity described above.

Looking at the steady state value of the current we do not notice any significant differences between the two simulation test cases. The small difference in the peak value could be easily

compensated by introducing a more accurate value for the on resistance of the MOSFET component.

## VII. CONCLUSION

We described a complete methodology for development of digital controls for power electronics. The approach allows a detailed validation of the control system in the VTB environment. Thanks to the Matlab real-time workshop it is possible to automatically generate code for the hardware implementation. Thanks to the VTB-Simulink interface these steps are simple and quite automatic, guaranteeing a completely unified procedure from design to implementation.

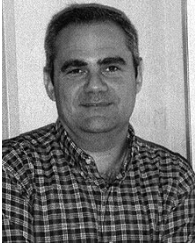
Through the analysis of a real application example we have described the level of accuracy that it is possible to reach with the different levels of model detail. While an averaged model allows a good evaluation of the dynamic performance, the switching model yielded more precise evaluations of all numerical aspects of the simulation. As far as evaluating the effectiveness of the control design, it seems to be unnecessary to use models any more sophisticated than the switching models.

We were able to show that control designs can be rigorously tested under extremely safe conditions because of the possibility to embed Simulink models into the analog simulation model. Automation of most of the validation steps, from insertion of the Simulink model of the controller into the VTB simulation to installation of the Simulink-generated controller code on the micro-controller removes most of the possibilities for error and gives the designer the possibility to rapidly and deeply test the control before hardware implementation.

Such an approach to the digital control design removes the vision of the microprocessor-based system as a sort of black-box, giving the designer the option to thoroughly test the interaction between plant and control.

## REFERENCES

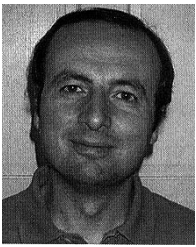
- [1] R. Alur, T. A. Henzinger, and E. D. Sontag, *Hybrid Systems III*. New York: Springer-Verlag, 1996, vol. 1066.
- [2] S. Carmeli, E. Cosatto, A. Monti, and C. Penno, "Software design methodology for power electronics applications," in *Proc. IEEE COMPEL'98*, Como, Italy, July 1998.
- [3] J. Guo, I. Celanovic, and D. Borojcic, "Distributed software architecture of PEBB-based plug and play power electronics systems," in *Proc. 16th Annu. IEEE Appl. Power Electron. Conf. Expo. (APEC'01)*, vol. 2, 2001, pp. 772-777.
- [4] S. Lloyd and Y. Zhang, "A software design tool for the regulator-controlled induction motor," in *Proc. IEEE Workshop Comput. Power Electron.*, 1992, pp. 251-257.
- [5] A. Kolesnikov, A. Monti, F. Ponci, and E. Santi, "Synergetic synthesis of dc-dc boost converter controllers: Theory and experimental analysis," in *Proc. IEEE APEC'02*, Dallas, TX, Mar. 2002.
- [6] T. Lovett, A. Monti, and R. A. Dougal, "The new architecture of the virtual test bed," in *Proc. IEEE-COMPEL'02*, Mayaguez, Puerto Rico, June 2002.
- [7] R. Dougal, T. Lovett, A. Monti, and E. Santi, "A multilanguage environment for interactive simulation and development of controls for power electronics," in *Proc. IEEE PESC'01*, Vancouver, BC, Canada, 2001.
- [8] L. O. Chua and P. Lin, *Computer-Aided Analysis of Electronic Circuits*. Englewood Cliffs, NJ: Prentice-Hall, 1975.
- [9] S. Cooper, *The Designer's Guide to Analog and Mixed-Signal Modeling Illustrated With VHDL-AMS and MAST*. New York: Avant!, 2002.
- [10] W. McKay, A. Monti, E. Santi, and R. Dougal, "A co-simulation approach for ACSL-based models," in *Proc. Huntsville Simulation Conf.*, Huntsville, AL, Oct. 3-4, 2001.



**Antonello Monti** (M'94–SM'02) received the M.S. and Ph.D. degrees in electrical engineering from the Politecnico di Milano, Milan, Italy, in 1989 and 1994, respectively.

From 1990 to 1994, he worked in the Research Laboratory, Ansaldo Industria, Milan, where he was responsible of the design of the digital control of a large power cycloconverter drive. From 1995 to 2000, he was Assistant Professor at the Department of Electrical Engineering, Politecnico di Milano. In August 2000, he became Associate Professor at the Department of Electrical Engineering, University of South Carolina, Columbia. He is author or co-author of about 100 papers in the field of power electronics and electrical drives.

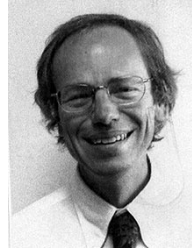
Dr. Monti became a member of the Computer in Power Electronics Committee of PELS, in 1996; in 1998 he served as Chairman of the IEEE Workshop on Computer in Power Electronics, Como, Italy, and from 2001 to 2002 he served as Chair of the same Committee.



**Enrico Santi** (SM'00) received the Dr.Ing. degree in electrical engineering from the University of Padua, Padua, Italy, in 1988, and the M.S. and Ph.D. degrees from the California Institute of Technology, Pasadena, in 1989 and 1994, respectively.

He was a Senior Design Engineer at TESLAco from 1993 to 1998, where he was responsible for the development of various switching power supplies for commercial applications. Since 1998, he has been an Assistant Professor in the Electrical Engineering Department, University of South

Carolina, Columbia. He has published several papers in power electronics and modeling and simulation and holds two patents. His research interests include switched-mode power converters, advanced modeling and simulation of power systems, modeling and simulation of semiconductor power devices, and control of power electronics systems.



**Roger A. Dougal** (SM'95) received the Ph.D. degree in electrical engineering from Texas Tech University, Lubbock, in 1983.

He then joined the faculty at the University of South Carolina, Columbia. He is the Director of the Virtual Test Bed Project, which is developing an advanced simulation and virtual prototyping environment for multidisciplinary dynamic systems.

Dr. Dougal received the Samuel Litman Distinguished Professor of Engineering award.



**Marco Riva** (M'99) received the M.S. degree in electrical engineering and the Ph.D. degree from the Politecnico di Milano, Milan, Italy, in 1994 and 1997, respectively.

From 1994 to 1997, he joined the Power Electronics and Electrical Drives Laboratory, Dipartimento di Elettrotecnica, Politecnico di Milano. His research activity was mainly addressed in the field of electronic converters analysis and control system optimization. In 1998, he joined the Electronic Laboratory, Physics Department, Univesità degli

Studi di Milano as Assistant Professor. He is author or co-author of about 40 scientific papers. His research topics include stability analysis, design organization and modeling of high efficiency, and low weight power converters for space application.

Dr. Riva is a member of AEI (Administration Committee), ANIPLA, and the IEEE Power Electronics, Industrial Electronics, and Circuit and System societies.