A New Testing Tool for Power Electronic Digital Control
S. Lentijo, A. Monti, E. Santi, C. Welch, and R. Dougal
Department of Electrical Engineering
University of South Carolina
Columbia, SC 29208, USA

Abstract—Digital control systems require sophisticated testing procedures before final implementation. Recently, various procedures for rapid prototyping have been proposed. However, in many cases they focus on implementation into platforms that are different from the final target. In this paper, we focus on a testing procedure for micro-controller-based control where the real final target is adopted for software testing. The test is performed without real-time constraint so that it can be performed with conventional hardware available in any office desktop. At the same time, information about real-time execution is extracted, which is a significant advantage over simulation environments.

I. INTRODUCTION

Digital technology has had a great impact on 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 boosted the reliability of designs.

Electrical drives and power electronics in general demand an intimate link between the discrete controller (control software) and the analog controlled sub-system (the plant). Discrete and analog parts of the system interact intimately, yet these two parts are normally modeled separately and the whole system is not tested in a global sense.

In this paper, we focus on testing the correctness of the software code after the implementation in the final target processor. In the following, we will call this procedure Processor in the Loop (PIL). This capability is one of the new features of the new version of the Virtual Test Bed Platform (VTB) [1] [2] [3].

In the rest of the paper, the PIL procedure is first introduced in a very simple example and then described in detail in the context of a complete experiment where a state space feedback controller is designed for a boost converter. The results obtained with this procedure are also compared with a standard control simulation and with experimental results.

II. PROCESSOR IN THE LOOP

The Processor in the Loop (PIL) capability allows one to test the actual control software running in a dedicated processor that controls a virtual prototype of the plant. PIL simulation provides an intermediate stage between simulation and deployment, where VTB models the plant, while code generated for the controller subsystem runs on the actual target hardware. The PIL simulation does not run in real-time, because the plant model runs in the host PC (Windows OS) and only the controller runs in the target hardware. The communications between them are via a serial link.

During PIL simulation, VTB simulates the plant model for one sample interval and exports the output data (output of the plant) to the control system under test via a serial communications link. When the target processor receives signals from the plant model, it executes the controller code for one sample step. The controller returns its output signals (output of the controller) computed during this step to VTB, via the same communications link. At this point one sample cycle of the simulation is complete and the plant model proceeds to the next sample interval. The process repeats and the simulation progresses.

The simulation system is based on the logic described in Fig. 1. VTB starts the simulation by running the communication process, enabling the serial port, then it sends the synchronization bytes followed by the data bytes, and wait for the microcontroller’s response. Meanwhile the microcontroller initializes the serial port and waits for the synchronization and data bytes. Once those bytes are received, a digital output is set in order to measure the control routine time length. After this, the state space control algorithm is called, and the control action is ready to be sent to VTB as well as the synchronization bytes. The following features are worth mentioning:

- The VTB blocks describing the plant do not depend on the processor used for the control. As long as we have a serial link and the interface protocol here described is implemented, the system can work. Extensions to other communication interfaces such as Universal Serial Bus (USB) are also under consideration.
- The Software is modular and it is easy to customize the communication to any number of inputs and outputs.
Thanks to the management of a digital output pin, a precise measurement of the computational time is also possible. This allows the user to check whether there is any condition, under which time limits are reached.

---

III. APPLICATION EXAMPLE

A. First Example: Second Order System

Let us focus on a simple example where a second order plant is controlled by using a simple PI Controller. The second order plant can easily be represented in VTB by means of a transfer function or by means of an equivalent electrical network such as in [1], [4]. The output voltage of this electrical network is controlled by means of a PI controller that will be implemented using a low-cost microcontroller board (Infineon C167CR-LM). The PIL block in VTB embeds the communication between the PC and the external target system.

The results for this PIL simulation are shown in Fig. 2, where a square wave reference with amplitude equal to 10V is applied to the system. The output of the system presents a small overshoot but never reaches the desired steady state; the signal rings indefinitely. This problem is caused by the use of fixed-point math in the 16-bit micro controller (the problem was intentionally introduced by performing a bad rescaling on the control variables to illustrate that the proposed procedure is capable of capturing these implementation-related interactions between plant and controller).

---

B. Second Example: Boost Converter

Now let us focus on a more interesting example: the control of a boost converter using a state feedback digital controller (see Fig. 3). We use this example to illustrate the approach through a full design process. The design steps described in detail below are:

1) Controller Design
2) Simulation Results in Simulink and VTB
3) PIL Simulation in VTB and Infineon Microcontroller
4) Hardware Verification

---

1) Controller Design: Let us start by considering the classic time-averaged model of the boost converter:

\[
\dot{X}_1 = -\frac{X_2}{L} (1 - u) + \frac{V_g}{L}
\]  
\[
\dot{X}_2 = \frac{1}{C} (1 - u) - \frac{X_2}{RC}
\]  
\[0 \leq u \leq 1\]
Where $X_1$ is the inductor current, $X_2$ is the capacitor voltage and $u$ is the switch duty cycle.

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

$$
\begin{bmatrix}
\Delta X_1 \\
\Delta X_2
\end{bmatrix} = 
\begin{bmatrix}
0 & -\frac{(1-\bar{u})}{L} \\
\frac{(1-\bar{u})}{C} & \frac{1}{RC}
\end{bmatrix} 
\begin{bmatrix}
\Delta X_1 \\
\Delta X_2
\end{bmatrix} + 
\begin{bmatrix}
\bar{X}_2 \\
\frac{1}{RC} \bar{X}_1
\end{bmatrix} k_1 + k_2 \begin{bmatrix}
\Delta X_1 \\
\Delta X_2
\end{bmatrix} \tag{4}
$$

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

$$
\Delta u = k_1 \Delta X_1 + k_2 \Delta X_2 \tag{5}
$$

The new state matrix including the feedback will be:

$$
\begin{bmatrix}
\bar{X}_1 k_1 \\
\frac{L}{C} \bar{X}_1
\end{bmatrix} + 
\begin{bmatrix}
\bar{X}_2 \left( \frac{1}{RC} \bar{X}_1 (1-\bar{u}) \right) \\
\left( \frac{1}{LC} \bar{X}_1 (1-\bar{u}) \right)
\end{bmatrix} \tag{6}
$$

The characteristic equation of the $\Phi_{cl}$ is given by:

$$
\lambda^2 + \left( \frac{1}{RC} \bar{X}_1 + \frac{L}{C} \bar{X}_1 \right) \lambda + \frac{1}{RC} \bar{X}_1 = 0 \tag{7}
$$

Using the properties of the second order equations and the eigenvalues of $\Phi_{cl}$ we obtain the following equations in matrix form:

$$
\begin{bmatrix}
-\bar{X}_1 \\
\bar{X}_2
\end{bmatrix} + 
\begin{bmatrix}
\bar{X}_1 \\
\bar{X}_2
\end{bmatrix} \frac{1}{RC} \bar{X}_1 \left( \frac{1}{LC} \bar{X}_1 \right) \tag{8}
$$

Given:

$$
\begin{bmatrix}
q_{11} & q_{12} \\
q_{21} & q_{22}
\end{bmatrix} \begin{bmatrix}
k_1 \\
k_2
\end{bmatrix} = \begin{bmatrix}
m_1 \\
m_2
\end{bmatrix} \tag{9}
$$

and solving for $K1$ and $K2$:

$$
\begin{bmatrix}
k_1 \\
k_2
\end{bmatrix} = \frac{1}{q_{11} q_{22} - q_{12} q_{21}} \begin{bmatrix}
q_{11} & q_{12} \\
q_{21} & q_{22}
\end{bmatrix} \begin{bmatrix}
m_1 \\
m_2
\end{bmatrix} \tag{10}
$$

2) Simulation Results in Simulink and VTB: This controller is first implemented in a Simulink block (see Fig. 4). The S-Function takes four inputs: the inductor current, input and output voltages and a reference voltage (internal parameter), and calculates the real duty cycle.

The controller is first implemented in Simulink for a preliminary test.

The Simulink schematic is imported into VTB through a compilation process to check the control performance together with a plant model. The schematic and the results are summarized in Fig. 5.

![State feedback controller block in Simulink](image1)

![VTB boost converter schematic including the simulink control block and the simulation results](image2)

3) PIL Simulation in VTB and Infineon Microcontroller: Once the state feedback controller has been successfully tested under the simulation environment, the next step is the implementation of the state feedback controller in the target hardware, which in this case is the Infineon C167CR-LM. This step could be performed automatically (some commercial tools support this process) or it could be performed by hand.

Notice that in the previous step the controller calculations were performed by VTB using floating-point, whereas in this step the operations are performed by the Infineon microcontroller using fixed-point.

Once we have the new control software we can embed the algorithm in the PIL logic and perform the PIL simulation as
mentioned in section II. VTB simulates the boost converter for one sample interval and exports the output signals (reference voltage, output voltage, inductor current, and input voltage) to the Infineon C167CR-LM board via a serial communications link. When the Infineon C167CR-LM processor receives signals from the plant model, it executes the controller code for one sample step. The state space controller returns its output signals computed during this step to VTB. At this point one step of the simulation is complete.

The simulation results for this PIL simulation, where VTB represents the plant model (Boost Converter) and the state space controller is running in the Infineon microcontroller, are presented in Fig. 6.

The simulation starts with a manual duty cycle equal to 0.5 and an input voltage equal to 12 V; in this case the output voltage stabilizes at 24 V after a initial transient, as shown in Fig. 6. Then the state space controller is enabled. Initially the reference output voltage is set to 25V, and as shown in fig.6 the actual output voltage is also 25V. When this reference voltage is step changed to 28V, 32V and finally 40V, the output voltage continues to correspond.

![Fig. 6. PIL Simulation Results. Output Voltage.](image)

By connecting the scope to the microcontroller and monitoring the digital output described in the flow chart of Fig. 1, we also obtain the information about processing time. In this case, for example, we measured the calculation time length needed by the microcontroller to perform one cycle of the control algorithm, using different approaches. First, the state space control algorithm is implemented using floating numbers and the time length was equal to 132 µs, then the same control algorithm was implemented using just integers numbers the time length was 42.5 µs, both results are shown in Fig. 7. Thanks to this information, the designer will decide whether this control algorithm as well as the target system meets the time specifications needed to control the plant, before implementing the whole experiment in hardware.

![Fig. 7. Control algorithm computation time in the microcontroller during PIL.](image)

IV. COMPARISON BETWEEN SIMULATION AND EXPERIMENTAL RESULTS

After testing the state feedback controller in the actual target, it was safe to move on and do the final test, which consisted of controlling the real boost converter with the state feedback controller implemented in the Infineon C167CR-LM.

A laboratory prototype was designed and built. The small-scale power converter system has the following nominal characteristics:

- Rated Input Voltage: 12 V
- Rated Output Voltage: 40 V
- Maximum Load: 100 W
- Input Inductance: 46 mH
- Output Filter Capacitance: 1.360 mF
- Main Switch: IRF540N

We want to show a comparison between the simulated and experimental results. These results show the transient that follows step change of the reference voltage from 13 to 16 V. (shown in Fig. 8 and Fig. 9)

![Fig. 8. Output Voltage (simulation).](image)
During the state space controller design, the two poles chosen were -3000 and -3500; the dominant pole gives a time constant equal to 333 $\mu$s. In the experimental results shown in Fig. 9 we can notice that the time constant is approximately 400 $\mu$s. Another difference between the simulated and experimental result is that the simulated output voltage is larger than the experimental one. This difference is due to the fact that the model of the boost converter used in the simulation does not account for any losses. For this particular example, the output voltage in the simulation is equal to 16.1 V and the experimental result is 15.6 V, if we consider that the main losses are in the diode in the boost converter, the average voltage drop can be approximated using (11)

$$V_D = 0.7 \times (1 - u)$$

the duty cycle ($u$) is equal to 0.25 and the average voltage drop in the diode is approximately 0.525 V, which is the difference between the simulated and experimental results.

**V. CONCLUSIONS**

This paper introduced a new procedure for control testing that can be applied to digital control systems. This procedure, called Processor in the Loop, allows the user to check the control software running together with a simulated model of the plant. Although the procedure is not performed in real-time, it allows the measurement of important quantities such as execution time that can increase confidence in the control software before the final testing.

**ACKNOWLEDGMENT**

This work was supported by the US Office of Naval Research (ONR) under grant N00014-02-1-0623

**REFERENCES**


