Processor-in-the-Loop Simulation, Real-Time Hardware-in-the-Loop Testing, and Hardware Validation of a Digitally-Controlled, Fuel-Cell Powered Battery-Charging Station

Zhenhua Jiang, Rodrigo Leonard, Roger Dougal, Hernan Figueroa, Antonello Monti
Department of Electrical Engineering
University of South Carolina
Columbia, SC 29208, USA
{jiang, leonard, dougal, figueroh, monti}@engr.sc.edu

Abstract—The fuel-cell powered battery-charging station provides an appealing solution to charging many batteries in remote missions. Final construction of such a complex system may be costly while traditional simulation is often insufficient to exactly capture the control dynamics. One way to bridge the gap between simulation and final system construction are processor-in-the-loop (PIL) and real-time hardware-in-the-loop (RT-HIL) simulations. This solution increases the realism of the simulation and provides access to hardware features that are currently not available in the software-only simulation. This paper presents approaches and results of PIL and real-time HIL simulations as well as hardware validation of embedded control implementation of the fuel-cell powered battery-charging station. The control algorithm is described and the embedded control implementation is proposed. The control algorithm implemented on a dedicated microcontroller is first tested together with the plant model built in the VTB environment through PIL simulation, then tested using the same plant model on an RT-HIL simulation (RTVTB), and finally validated on the actual hardware. Results of PIL, RT-HIL, and hardware validation are given.

I. INTRODUCTION

Modern handheld electronic devices such as cellular phones, portable computers, PDA’s, camcorders, radios, etc., have fueled a great need for hi-tech rechargeable batteries [1]. The fuel-cell powered battery-charging station [2]-[4], as shown in Figure 1, provides an appealing solution to charging these batteries in the remote missions. In this system, a fuel cell stack as the power source charges up to three lithium-ion battery packs through separate buck converters. The sum of the maximum safe charging currents of the three batteries may exceed the maximum current available from the fuel cell stack. The power (current) available from the fuel cell stack is appropriately allocated among these batteries by actively controlling the power converters. A digital controller is used to coordinate these power converters. The charging currents and battery voltages are monitored and fed to the controller. This digital controller calculates the reference charging current and the corresponding duty cycle for each channel, and also outputs PWM switching signals to these buck converters.

In this application, the digital controller is realized by a microcontroller-based embedded control system [5]-[7]. The control algorithm is coded and then downloaded to a dedicated microcontroller system. Final construction of such a complex system may be costly while traditional simulation is often insufficient to exactly capture the control dynamics. One way to bridge the gap between simulation and final system construction are processor-in-the-loop (PIL) and real-time hardware-in-the-loop (RT-HIL) simulations [8]-[11]. This solution increases the realism of the simulation and provides access to hardware features that are currently not available in the software-only simulation. Prior to the final system construction, these simulations provide an approach to testing the actual control software running on a dedicated processor with a virtual prototype of the plant, where Real-Time Virtual Test Bed (RTVTB) [11] platform models the plant while the code generated for the control system runs on the real target hardware. In the PIL simulation, the simulation model in VTB communicates with the controller via a serial port, while in RT-HIL simulation, the simulation model in RTVTB and the control hardware communicate via a data acquisition card.

Figure 1. Block diagram of a fuel-cell powered battery-charging station.

The work reported in this paper is an important extension of the work in [4], where a real-time strategy for active power sharing in a fuel-cell powered battery-charging station was presented. The objective of this paper is to present approaches and results of PIL and real-time HIL simulations as well as hardware validation of embedded control implementation of the fuel-cell powered battery-charging station. The control algorithm implemented on a dedicated microcontroller is first tested with the plant model built in VTB through PIL, then tested using the same plant model on an RT-HIL simulation (RTVTB), and finally validated on the actual hardware. Results of PIL, RT-HIL, and hardware validation are given.
II. CONTROL ALGORITHM DEVELOPMENT

Although the active power sharing strategy has been developed and reported in [4], it is briefly reviewed in this section to give the readers an insight into the control algorithm. From Figure 1, it is clear that the electrical power from the fuel cell stack is distributed among three batteries, which can be expressed in (1).

\[ P_{fc} = \frac{P_1 + P_2 + P_3}{\eta_{filter} \cdot \eta_{buck}} \]  

(1)

where \( P_{fc} \) is the electrical power from the fuel cell stack, \( P_1, P_2, \) and \( P_3 \) are the electrical power to three charging channels respectively, \( \eta_{filter} \) is efficiency of the input filter, and \( \eta_{buck} \) is efficiency of the buck converter assuming that efficiency of each buck converter is identical and fixed. In practice, the power distribution among the batteries is realized by regulating the charging currents of the batteries. The following equation relates the current available from the fuel cell stack to three charging currents.

\[ I_{fc} = \frac{d_1 I_1 + d_2 I_2 + d_3 I_3}{\eta_{filter} \cdot \eta_{buck}} \]  

(2)

where \( I_{fc} \) is the current available from the fuel cell stack, \( I_1, I_2, \) and \( I_3 \) are the charging currents to three batteries respectively, and \( d_1, d_2, d_3 \) are, respectively, duty cycles of three buck converters and they have values between 0 and 1.

Since the fuel cell current is limited, the sum of the right hand side in (2) should be less than some value. Considering the variations of the voltages of the fuel cell stack and the batteries are not too large, the duty cycle will vary within a limited small range. Based on this, we can take the following expression as a criterion for power distribution.

\[ I_1 + I_2 + I_3 \leq I_{lim} \]  

(3)

where \( I_{lim} \) is a preset limit for the total charging current that can be estimated according to the current capacity of the fuel cell, the average duty cycle of buck converters and efficiencies of the input filter and buck converters.

The charge level of a battery can be expressed as follows.

\[ C(t_{end}) = C_0 \cdot SOC_0 + \int_{t_0}^{t_{end}} I \cdot dt \]  

(4)

where \( C_0 \) is the rated capacity of the battery, \( SOC_0 \) is the initial state-of-charge that represents the fractional charge in the battery, \( I \) is the charging current, \( t_{end} \) is the total charging time. The depth-of-discharge, defined as unity minus state-of-charge, indicates the charge needed to fill the battery. From (4), it is seen that the charge that the battery needs to get fully charged is the integral of the charging current over the total charging time. If DC currents of the same magnitude are applied to charge different batteries of the same capacity, the charging time will be approximately proportional to the depth-of-discharge. On the other hand, if we want all the batteries to become fully charged at the same time, the charging current of each battery will be proportional to its fraction of the total depth-of-discharge, as shown in (5) [3, 4].

\[ I_i = I_{lim} \cdot \frac{1 - SOC_i}{\sum_{i=1}^{3} (1 - SOC_i)}, i = 1, 2, 3 \]  

(5)

where \( I_i \) is the charging current of the \( i \)th battery, \( I_{lim} \) is the total available charging current, and \( SOC_i \) is the state-of-charge of the \( i \)th battery. It is clear that the current sharing strategy shown in (5) satisfies the criterion given in (3) and utilizes as much power of the fuel cell as possible.

In the above algorithm, the charging current varies with the state-of-charge. Since it is not practical to measure the state-of-charge directly, a method should be found to estimate it from the measured battery voltage and charging current. For lithium-ion batteries, an approximate relationship between the state-of-charge and open-circuit voltage can be found when the state-of-charge is not within the extreme range, i.e., if it is between 0.1 and 0.9. Here, the state-of-charge is estimated according to a linear relationship given in (6) [3, 4].

\[ SOC = \begin{cases} 0.9, & v_0 > v_1 \\ \frac{v_0 - a}{b} + c, & v_2 < v_0 < v_1 \\ 0.1, & v_0 < v_2 \end{cases} \]  

(6)

where \( a, b \) and \( c \) are constants and can be easily obtained through a series of experiments, \( v_0 \) is the battery open-circuit voltage that can be estimated online by current correction [4]. As shown in [3] and [4], this approximation does not affect the performance of the algorithm significantly.

III. EMBEDDED CONTROL IMPLEMENTATION

An Infineon microcontroller (C167CR-LM) was chosen for the embedded control system of the fuel-cell powered battery-charging station. Infineon C167CR-LM is a full-featured 16-bit single-chip CMOS microcontroller. It combines high CPU performance with high peripheral functionality and means for power reduction. The board can be adapted to a wide range of closed-loop applications due to its large number of I/O devices (111 I/O lines), a 10-bit resolution A/D converter (4 input channel) with a sampling time of 9.7 µs, 4 high-resolution PWM channels, 2 capture and compare units, and synchronous and asynchronous serial units. A 2 KB local memory is used to store programs and data.

Figure 2 shows the configuration of the microcontroller system that controls all three buck converters in the fuel-cell powered battery charging station. In each charging channel, a voltage chopper consisting of a MOSFET and a Schottky diode converts the voltage of the fuel cell stack to an appropriate lower voltage to charge the battery through a low-pass filter consisting of the power inductor \( L \) and the capacitor \( C \). A current transducer is used to sense the inductor current. A low-pass filter consisting of \( R_2 \) and \( C_2 \) filters the output signal from the current sensor. Resistors \( R_1 \) and \( R_2 \) form a voltage divider to measure the battery voltage. A capacitor \( C_1 \) is connected in parallel with \( R_2 \) to filter the ripple in the voltage. The measured charging current and voltage are input to the controller. The
microcontroller system produces a PWM signal to the buck converter to control the charging current or voltage. The buck converter can be shut off by a low pulse at a pin of the gate driver from the EN port of the controller when the charging process terminates or an over-voltage or over-current fault is detected. The microcontroller system has six input signals and six output signals.

Figure 2. Configuration of the microcontroller system that controls three buck converters.

A constant current/constant voltage (CC/CV) protocol is applied to charge the batteries. There are two regulation modes in the charging station: current mode and voltage mode. When the voltage at the output terminal of each charging channel is lower than the reference voltage, this charging channel will work under current mode. Once the output terminal voltage reaches the reference voltage, this charging channel will move to voltage mode and output a constant voltage at the terminal. A hysteresis allows this charging channel to switch back to current mode whenever the measured output terminal voltage drops under its low threshold that is a little bit lower than the reference voltage. The mechanism for the change of current mode and voltage mode is shown in Figure 3.

Figure 3. Hysteresis mechanism for the change between voltage mode and current mode.

The flow chart of the control algorithm running on the microcontroller is shown in Figure 4. The main routines in the control algorithm include initialization, circuit protection, total current decision, depth-of-discharge estimation, current sharing strategy, and compensation loop modules. The total available charging current $I_{\text{lim}}$ for current mode channels is calculated according to the fuel cell current limit specified by the user and the charging currents for voltage mode channels. The depth-of-discharge is estimated for each battery according to the measured battery voltages and currents, as shown in (6). The current sharing strategy module is to calculate the reference charging currents according to the estimated depth-of-discharge, and it is developed based on the current sharing algorithm which is shown in (5).

Figure 4. Flow chart of the control algorithm.

The compensation loops for current regulation and voltage regulation are respectively implemented as follows.

\[
d(n) = d(n-1) + k_{p,i} \cdot (I_{\text{ref},i}(n) - I_i(n)) + \sum_{k=0}^{\infty} (I_{\text{ref},i}(k) - I_i(k)) + k_{d,i} \cdot (I_i(n) - I_i(n-1))
\]

\[
d(n) = d(n-1) + k_{p,v} \cdot (V_{\text{ref}} - V_i(n))
\]

where $d(n)$ and $d(n-1)$ are duty cycles at the current and previous steps respectively, $I_i(n)$ and $V_i(n)$ are, respectively, the sampled current and voltage of the $i$th battery at the current step, $I_{\text{ref}}(n)$ is the reference current of the $i$th battery, and $V_{\text{ref}}$ is the reference voltage, $k_{p,i}$ and $k_{p,v}$ are current gain and voltage gain, $k_{i,i}$ is the current integral gain, and $k_{d,i}$ is the current derivative gain.

IV. PROCESSOR-IN-THE-LOOP AND REAL-TIME HARDWARE-IN-THE-LOOP SIMULATIONS

The system was tested in three ways: 1) with real digital controller operating in non-real-time handshaking mode with the simulated plant – PIL simulation, 2) with real digital controller running at full speed interacting with a real-time simulation model of the plant – RT-HIL simulation.
simulation, and 3) the fully assembled system with embedded digital controller running the actual hardware – hardware validation. This section presents the approaches and results of the first two ways: PIL and RT-HIL. Section V will discuss the hardware validation.

A. Processor-in-the-Loop Simulation Approach

The PIL simulation is the first step taken to test the control algorithm in the embedded controller. At this stage, the VTB simulation model interacts with the microcontroller via a serial connection. VTB is non-real-time simulation software. At each time step, VTB simulates the plant model for one sample interval and exports the system output to the controller. When the controller receives signals from the plant model, it executes the controller code for one sample interval. The controller returns its control signals 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.

This stage of the design process is useful in determining the parameters for control of the plant, but it does not take into account many variables involved in the interaction between the controller and the plant. Some of these variables include A/D converter accuracy and variations in the PWM signal provided by the controller (since all of this is being transmitted digitally through the serial ports). Also, at this point the VTB simulation requests the controller to step and waits for a control response before proceeding. This artificial synchronization can introduce errors. However, the real-time HIL simulation approach that will be discussed below takes into account these variables by interacting with the controller in the same manner the real hardware plant does, as well as providing a much more accurate representation of the speed of the control hardware’s effect on the control.

B. Real-Time Hardware-in-the-Loop Simulation Approach

In order to accurately test the performance of the control algorithm implemented on the microcontroller, a real-time HIL simulation is conducted with RTVTB.

The RTVTB is a hard real-time simulation environment for rapid prototyping of control systems [11], [12]. RTVTB is an affordable and versatile alternative to many available commercial real-time simulation systems that are costly and use proprietary hardware, because it is completely implemented with public domain software and off-the-shelf hardware. RTVTB is developed under Linux operating system. Linux is selected due to its low cost, high performance and support for hard real-time. In RTVTB, an open-source package, Real-Time Application Interface (RTAI) [13], is utilized to create the hard real-time tasks that RTVTB needs to become a hard real-time Linux simulation environment. RTAI is a kernel modification package of Linux that permits the handling of time-critical tasks. A real-time HIL simulation system requires I/O interfaces to hardware. In RTVTB, this is achieved by using Comedi [14], which is a freeware project that develops open-source device drivers for many different data acquisition (DAQ) cards. Figure 5 shows the architecture of the RTVTB. This architecture allows the user to perform a real-time HIL testing of a system that includes real hardware. This testing phase can be considered the very last step before final system integration. Through this process, the user can verify not only the algorithmic correctness but also its capability to meet the real-time constraints when running on the actual hardware.

Figure 5: Architecture of the RTVTB.
The microcontroller receives six analog signals from the RTVTB solver. These signals are converted from analog to digital through the 10-bit ADCs. When the microcontroller receives signals from the plant model, it executes the control algorithm described previously for one sample step. The result of the control calculation sets the duty cycles of PWM output signals. These signals are averaged by means of a custom interface [15]. This custom interface provides accurate integration of the controller signal without overloading the simulation platform. The interface, implemented in a FPGA device, returns the average values of the PWM duty cycles to the RTVTB simulator. At this point, one sample cycle of the simulation is completed and the plant model proceeds to the next sample interval. The process repeats and the simulation progresses.

C. PIL and Real-Time HIL Simulation Results

The processor-in-the-loop simulation was first conducted with the microcontroller and VTB. The initial states-of-charge of the batteries were 0.8, 0.75, and 0.7 respectively. The total available current was 2A. The results of the PIL simulation are shown in Figure 7. From Figure 7, it is seen that during current regulation mode the battery voltages were different and the charging currents varied with the voltages in real time. The battery with the lowest voltage (and thus the least charge) was charged with the highest current, and its voltage increased more rapidly than the others. The voltages of these batteries reached the reference voltage almost simultaneously and then their currents tapered. The small fluctuations in the currents and voltages were due to the fact that the microcontroller worked digitally and the minimum change of the duty cycle was about 0.1%. The controller parameters were tuned at this time and would be used in the next two tests.

The real-time hardware-in-the-loop system shown in Figure 6 was then tested. The initial conditions and control parameters were the same as those in the PIL simulation. The results of real-time simulation are shown in Figure 8. From Figure 8, it is seen that during current regulation mode the battery voltages were different and the charging currents varied with the voltages in real time. The battery with the lowest voltage (and thus the least charge) was charged with the highest current. It is seen that the fluctuations of the voltage and currents were bigger than those in the PIL simulation. This was because the inaccuracy of the A/D converters and variations in the PWM signals provided by the controller were accounted for in the real-time simulation. In addition, since the simulation was in real-time, the controller was no longer artificially synchronized with the plant simulation, thus creating a more accurate representation of the real performance of the complete system.
V. HARDWARE VALIDATION

Finally, the embedded control system was validated on real hardware. A prototype of the microcontroller-based fuel cell powered battery-charging station was built in the laboratory. The power source was an H Power D35 PEM fuel cell stack, which had a nominal power capacity of 35W and an open-circuit voltage of 24V. Three 4-cell Panasonic lithium-ion batteries were used. Figure 9 shows a photograph of the hardware, and Table I describe the components used in the experiment. In order to ensure that each charging current would never exceed the safe maximum charging current (that was 900mA for the batteries used in the experiment), the total charging current was set at 2.0 A. The initial open-circuit voltages were, respectively, 16.2V, 15V and 14.8V. Note that the initial conditions were a little different than those in the simulations. The experiment results are shown in Figure 10.

From Figure 10, it is seen that, during current regulation mode, all of the charging currents varied in real time with the battery voltages that were initially different. The battery with the lowest voltage was charged at the highest rate. As each battery approached full charged, its respective control mode was switched to constant voltage. This switch occurred at 1000s, 2300s, and 5000s, respectively for the three batteries. During constant voltage mode, the charging currents each slowly tapered off. The big fluctuations in the waveforms shown in Figure 10 are attributed to the fact that the duty cycles were affected by the noise introduced by the ground signal of the power circuit board and thus there were some errors between the desired duty cycle and the actual one. This could be improved by using an isolated control circuit.

From the experiment results, it is seen that the control algorithm running on the microcontroller worked well on the actual hardware. Experiment results also validate the PIL and RT-HIL simulations.
Figure 10. Experiment results of the embedded control system.

VI. CONCLUSION

This paper has presented approaches and results of processor-in-the-loop and real-time hardware-in-the-loop simulations as well as hardware validation of embedded control implementation of the fuel-cell powered battery-charging station. The control algorithm is described and the embedded control implementation is proposed. The control algorithm implemented on a dedicated microcontroller is first tested with the plant model built in VTB through PIL, then tested using the same plant model on an RT-HIL simulation (VTB), and finally validated on the actual hardware. Results of PIL, RT-HIL, and hardware validation are given.

Experiment results validate the PIL and RT-HIL simulations. It is shown that the PIL and RT-HIL simulations provide a good way to bridge the gap between simulation and final system construction.

ACKNOWLEDGEMENTS

This work was supported by the US Office of Naval Research under grants N00014-03-1-0952 and N00014-02-1-0623.

REFERENCES