SLAC-PUB-13744

A HIERARCHICAL CONTROL ARCHITECTURE FOR A PEBB-BASED ILC MARX MODULATOR*

K. Macken[ξ], C. Burkhart, R. Larsen, M.N. Nguyen, and J. Olsen
SLAC National Accelerator Laboratory
2575 Sand Hill Road
Menlo Park, CA 94025, USA

Abstract

The idea of building power conversion systems around Power Electronic Building Blocks (PEBBs) was initiated by the U.S. Office of Naval Research in the mid 1990s. A PEBB-based design approach is advantageous in terms of power density, modularity, reliability, and serviceability. It is obvious that this approach has much appeal for pulsed power conversion including the International Linear Collider (ILC) klystron modulator application. A hierarchical control architecture has the inherent capability to support the integration of PEBBs. This has already been successfully demonstrated in a number of industrial applications in the recent past. This paper outlines the underlying concepts of a hierarchical control architecture for a PEBB-based Marx-topology ILC klystron modulator. The control in PEBB-based power conversion systems can be functionally partitioned into (three) hierarchical layers; system layer, application layer, and PEBB layer. This has been adopted here. Based on such a hierarchical partition, the interfaces are clearly identified and defined and, consequently, are easily characterised. A conceptual design of the hardware manager, executing low-level hardware oriented tasks, is detailed. In addition, the idea of prognostics is briefly discussed.

I. INTRODUCTION

The PEBB concept as envisioned by the U.S. Navy aimed at simplifying power converter design and assembly through the use of intelligent and reconfigurable standard building blocks with well defined functionality and interfaces [1]. The underlying idea was to build power converters in much the same way as personal computers. Reducing the complexity of the design process directly translates into a reduction of design effort and associated costs.

Motivated by the desire to ease serviceability and achieve high availability, a PEBB-oriented approach is found to be attractive compared to anything else in its day to design an ILC Marx modulator.

The PEBB concept is a platform-based approach. An efficient platform-based approach requires not only the definition of basic building blocks with defined functionality and interfaces, but also standardisation of the control architecture.

A PEBB can be seen as a defined set of hardware that integrates power semiconductor devices, gate drivers, protection, and other components. Sufficient intelligence needs to be embedded into a PEBB so that it is capable of processing electric power with a defined set of characteristics.

PEBBs are interconnected to build a system as illustrated schematically in Figure 1. The integration of these blocks, irrespective of how they are configured together, necessitates the support by the overall control architecture in order to produce the desired system behaviour and performance. Prior work in this area [2] demonstrated that a hierarchical control architecture has the inherent capability to support the integration of PEBBs.

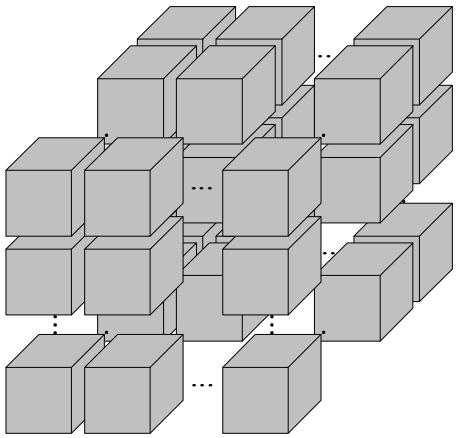

Figure 1. Illustration of the PEBB concept. Standardised building blocks (represented here as cubes) with well defined functionality and interfaces are interconnected in series and/or parallel to construct a power converter for a specific application. The number of blocks required scales with desired power and voltage ratings. All blocks may have identical functionality, but also a combination of blocks with different functionality is possible as well. The arrangement of the blocks determines the overall dimension, i.e. length, width, and height, of the converter.

It was discussed in [3] to design the control architecture of PEBB systems based on a hierarchically layered model or a

* Work supported by the U.S. Department of Energy under contract DE-AC02-76SF00515
ξ email: kmacken@slac.stanford.edu

so-called Open System Architecture (OSA) model by analogy to the Open Systems Interconnection (OSI) model as shown in Figure 2. The OSI model is an open model that allows the integration of devices from various manufacturers into one single application. The OSA model partitions control into functional layers which can be abstracted from each other. Information is exchanged between layers.

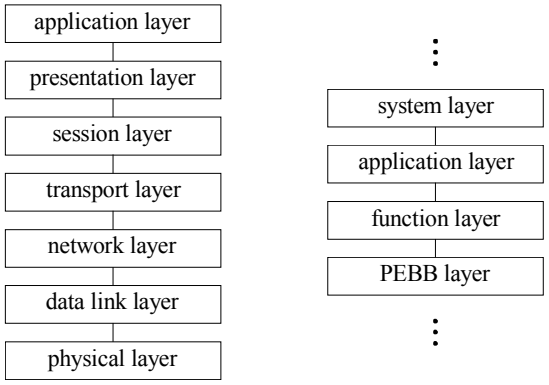

Figure 2. Analogy between OSI model (left) and OSA model (right). A control architecture can be partitioned into four layers using a hierarchical concept. It is important to recognise that the model is not restricted to four layers. There can be more than four layers or even less.

Using this concept of hierarchical layers it is possible to standardise the control architecture through standardisation of interfaces between layers.

This paper describes the concepts of a hierarchical control architecture for a PEBB-based Marx-topology ILC klystron modulator. In section II, the functional partitioning of the control of the modulator into hierarchical layers is discussed. Section III follows with an interface characterisation. A conceptual design of the hardware manager is presented in section IV. The idea of prognostics is briefly explained in section V. Section VI concludes the discussion.

II. HIERARCHICAL PARTITIONING

A. Review of Basic Concepts

The design of hierarchically layered control architectures for PEBB-based systems has been a significant theme of research. The basic concepts related to a hierarchically layered control architecture design approach have been described thoroughly in [4-6] and are summarised below.

A layered model as described above can be used to hierarchically partition control authority in PEBB-based systems. One way to hierarchically partition the control authority in PEBB-based power conversion systems is illustrated in Figure 3. The partitioning is based on functionality. The control authority is partitioned between three hierarchical layers; system layer, application layer, and PEBB layer. The resultant three-layer control architecture is composed of a hardware manager, an application manager, and a system level controller.

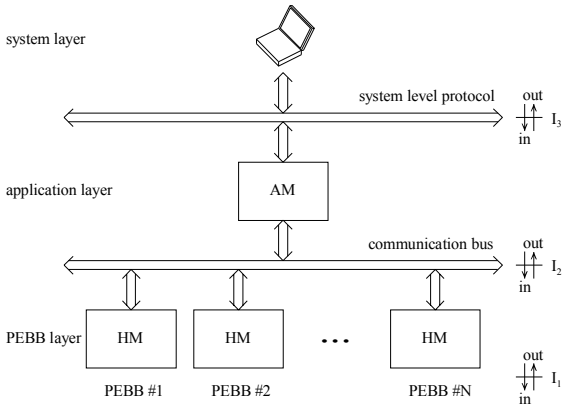

Figure 3. Illustration of a control architecture for a PEBB-based system partitioned between three hierarchical layers. Hardware managers (HMs) execute low-level hardware-oriented tasks at the PEBB layer. An application manager handles higher-level application-oriented tasks at the application layer. A system level controller deals with system control and monitoring functions at the system layer. I_1, I_2, and I_3 denote interfaces 1, 2, and 3, respectively. Arrows indicate direction of data flow (i.e. either incoming or outgoing) across each interface.

1) Hardware Manager: The hardware manager is a controller designed at the PEBB layer in the hierarchy described in Figure 2. The hardware manager is application independent. It masks all PEBB specific control tasks and makes them invisible to the applications manager. The hardware manager is an integral part of the PEBB adding intelligence to it. It handles (fast) low-level hardware related control tasks such as PWM generation, signal sensing and A/D conversion, protection, local control, and communication.

2) Application Manager: The application manager is a controller designed at the application layer. The application manager is hardware independent. It handles (slow) higher-level application related control. Typical control tasks include system initialisation, monitoring and protection, higher level control, reconfiguration, and diagnostics and prognostics.

3) System Level Controller: The system level controller is a controller designed at the system layer. The system level controller performs high-level control. Typical tasks include responding to user commands, coordinating performances between converters, and monitoring system execution.

In a hierarchical control architecture, high flexibility and modularity can be achieved through standardisation of the interfaces. It is particularly noteworthy that this approach has the advantage that the lower layers of the PEBB design can remain (quasi) consistent from application to

application. Customization for a particular application is at the higher layers (e.g. application layer).

The interfaces become clearly identified and defined by the partitioning. For the three-layer hierarchy presented above, the first interface is within the PEBB, between the power electronics hardware and the hardware manager. The second interface is between the hardware manager and application manager. The third interfaces the application manager to the system level controller. It is now easy to characterise the interfaces. This will be discussed in more detail in the next section.

B. Three-Layer Control Hierarchy

Figure 4 shows the tailored control architecture for a PEBB-based ILC Marx modulator based on a hierarchical partitioning as discussed above.

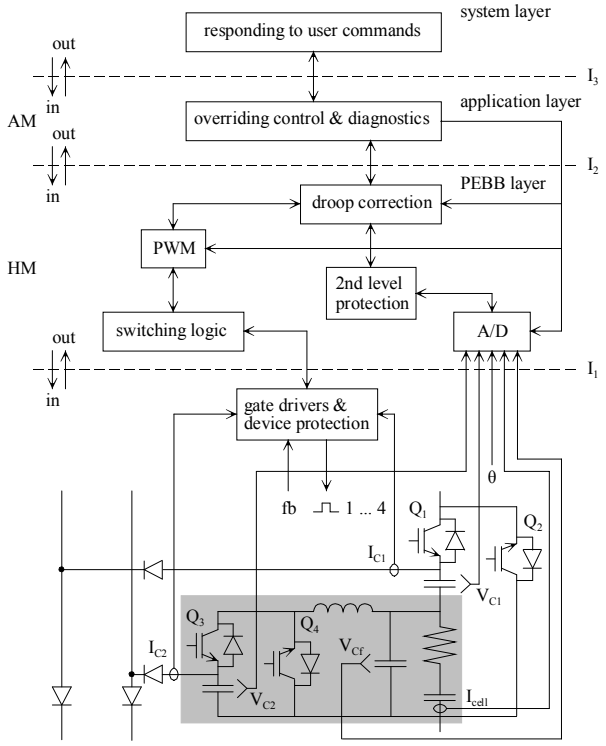

Figure 4. Control architecture of a PEBB-based ILC Marx modulator showing hierarchical layers and interfaces. The hardware manager at the PEBB layer and the application manager at the application layer are indicated.

A PEBB-based Marx modulator is built around *N* Marx cell PEBBs. A Marx cell PEBB is composed of a main cell and a correction cell. The correction cell, nested at the bottom of the main cell through an output filter (indicated by the shaded area in Figure 4), is PWM regulated to compensate for the capacitor voltage droop on the main cell during its discharge, maintaining the combined output voltage of both cells within the specified flatness tolerance. Each Marx cell PEBB is controlled by a hardware manager at the PEBB layer. One layer higher up the hierarchy, an application manager controls the *N*-cell Marx modulator.

III. INTERFACE CHARACTERISTICS

In the previous section it was shown how the control architecture of the PEBB-based ILC Marx modulator is partitioned into three hierarchical layers with clearly identified and defined interfaces. Once the interfaces are identified and defined, they can be easily characterised as shown in [7]. An interface characterisation is instrumental in designing the various hierarchical controllers.

The signals that are communicated through an interface can be classified based on their type as follows:

- control (C)
- measurement/diagnostics (M)
- protection (P)
- settings (S)

The characteristics of interface 1 (I_1), as defined in Figure 4, are summarised in Table 1. The characteristics of interfaces 2 (I_2) and 3 (I_3) are omitted in the discussion presented here for brevity.

There are four incoming signals transferred across interface 1, i.e. $V_{ge,Q1}$, $V_{ge,Q2}$, $V_{ge,Q3}$, and $V_{ge,Q4}$, where V_{ge} refers to gate-emitter voltage and Q_1, Q_2, Q_3, and Q_4 refer to main discharge IGBT, main charge IGBT, correction discharge IGBT, and correction charge IGBT, respectively, as indicated in Figure 4. They are all control signals. Their use is to turn on and off the IGBTs.

There are 15 outgoing signals communicated across interface 1. The collector-emitter voltages $V_{ce(sat),Q1}$, $V_{ce(sat),Q2}$, $V_{ce(sat),Q3}$, and $V_{ce(sat),Q4}$ are measurement signals and are used to prognosticate the aging of the IGBT. The underlying concepts of prognostics will be discussed briefly in section V. The measurement signals $V_{ge,Q1}$, $V_{ge,Q2}$, $V_{ge,Q3}$, and $V_{ge,Q4}$ are used for diagnostic purposes of the gate drivers. The sensed charge currents through main capacitor I_{C1} and correction capacitor I_{C2} are communicated through interface 1 for two purposes. First, the sensed currents are measurement signals used to evaluate the aging of the capacitances. Second, the sensed currents are protection signals used to detect charging faults. The voltages V_{C1}, V_{C2}, and V_{Cf} are the voltages across main capacitor, correction capacitor, and filter capacitor, respectively. All three are control signals. They are used for the droop correction. In addition, V_{C1} and V_{C2} are also measurement signals. They are used to evaluate the capacitances. The cell current I_{cell} is a protection signal to detect an overcurrent condition. Finally, the heatsink temperature θ is also a protection signal used for overtemperature detection.

Table I lists the required analog-to-digital (A/D) resolution, sampling frequency f_s, and required bandwidth associated with the various signals as well.

The resolution of the A/D converter has to be chosen such that the analog equivalent of the A/D converter least significant bit (LSB) q_{adc} is less than the allowed variation of

Table 1. Characterisation of interface 1.

	type	signal	resolution [bits]	f_s [kHz]	bandwidth [Mb/s]
in	C	$V_{ge,Q1}$	-	-	0.00041
	C	$V_{ge,Q2}$	-	-	0.00041
	C	$V_{ge,Q3}$	-	-	3.26
	C	$V_{ge,Q4}$	-	-	3.26
out	M	$V_{ce(sat),Q1}$	8	10	0.04
	M	$V_{ce(sat),Q2}$	8	10	0.04
	M	$V_{ce(sat),Q3}$	8	10	0.04
	M	$V_{ce(sat),Q4}$	8	10	0.04
	M	$V_{ge,Q1}$	8	10	0.04
	M	$V_{ge,Q2}$	8	10	0.04
	M	$V_{ge,Q3}$	8	10	0.04
	M	$V_{ge,Q4}$	8	10	0.04
	M, P	I_{C1}	8	10	0.04
	M, P	I_{C2}	8	10	0.04
	C, M	V_{C1}	8	40	0.16
	C, M	V_{C2}	8	40	0.16
	C	V_{Cf}	8	40	0.16
	P	I_{cell}	8	10	0.04
	P	θ	8	10	0.04

the (scaled) sensed variable Δx

$$q_{adc} \le H_x \Delta x \tag{1}$$

where H_x is gain of the sensor and q_{adc} can be expressed as follows

$$q_{adc} = \frac{FSR}{2^{N_{adc}}} \tag{2}$$

where FSR is full scale range of the A/D converter and N_{adc} is resolution of the A/D converter. For a 12-bit A/D converter, an error of one LSB is 1/4096 of the full signal range. Substituting (2) into (1) and rearranging, yields the following expression for the required resolution of the A/D converter

$$N_{adc} \ge \text{ceil}\left[\log_2\left(\frac{FSR}{H_x \Delta x}\right)\right] \tag{3}$$

where the ceil function gives the smallest integer value greater than or equal to the argument.

The required bandwidth B for an analog signal is obtained as follows

$$B = \frac{1}{\pi t_r} \tag{4}$$

where rise time t_r can be expressed as

$$t_r \le \frac{1}{2^{N_{pwm}} f_{sw}}. \tag{5}$$

Combining (4) and (5) yields the following expression

$$B = \frac{2^{N_{pwm}} f_{sw}}{\pi} \tag{6}$$

where f_{sw} is switching frequency and N_{pwm} is required PWM resolution calculated as follows

$$N_{pwm} \ge \text{ceil}\left[N_{adc} + \log_2\left(\frac{1}{D}\frac{V_o H_v}{FSR}\right)\right] \tag{7}$$

where D is duty cycle, V_o is output voltage, and H_v is gain of voltage sensor. The latter equation is obtained assuming that in order to avoid limit cycles, the resolution of the PWM signal should be such that the change in the output voltage due to one LSB change of the duty ratio is smaller than one LSB of the A/D converter.

The minimum bandwidth required to transmit a digital signal is

$$B = \frac{R_b}{2} \tag{8}$$

where binary pulse rate (or bit rate) R_b in bits per second is given by

$$R_b = Nf_s \tag{9}$$

where N is number of bits per sample.

IV. CONCEPTUAL DESIGN OF HARDWARE MANAGER

The block diagram of the hardware manager is shown in Figure 5. It consists of a field programmable gate array (FPGA), 2 Mb RAM for storing data, boot control, indicator LEDs, various A/D converters, an universal series bus (USB), and a small form-factor pluggable (SFP).

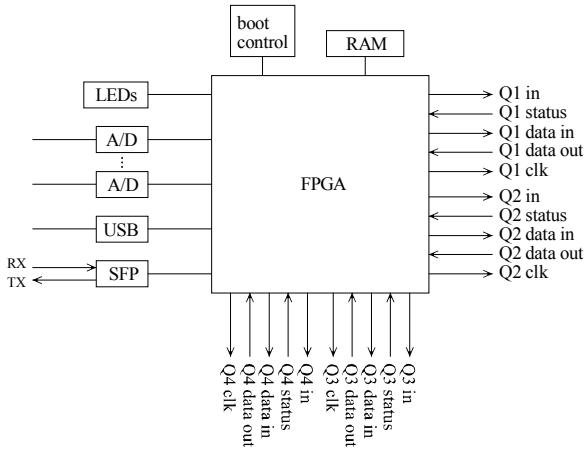

Figure 5. Block diagram of hardware manager.

The boot control configures the FPGA upon system boot-up. It consists of a 2Mb PROM for the primary boot image and a 2Mb flash memory to hold a secondary boot image. The secondary boot image can be downloaded through the FPGA over ethernet. On system power-up, the FPGA will always boot from the primary image. The A/D converters sample various of the measured variables as indicated in Figure 4 and listed in Table 1 such as voltages V_{C1}, V_{C2}, and V_{Cf}. One A/D converter is dedicated to digitise the output of a temperature sensor. This sensor is configured to monitor the on-board temperature, two remote temperatures, and the supply voltage. The USB allows the hardware manager to be connected directly to a computer for prototyping purposes if desired. The SFP interfaces the hardware manager to a Gb ethernet link.

The hardware manager receives clock and control input data from the application manager and sends data to it over ethernet. Further, the hardware manager sends clock and data signals to the gate drivers and receives data and status signals from them as illustrated in Figure 5. The gate drivers are equipped with opto-couplers to provide sufficient isolation.

The format of the serial data packet sent from the gate drivers to the hardware manager is schematically illustrated in Figure 6. The first field in the packet is a start bit followed by address field, data field, and error check field. The data field consists of N bytes of data. Each byte of data has 8 bits of actual data preceded by a logic 1. This insures that there will be a logic 1 at least every 9 bits of data transmission. A string of more than 9 logic 0's will indicate a synchronisation gap. A parity bit is added to every packet for error checking purposes. A similar format is used to encode the data sent from the hardware manager to the gate drivers.

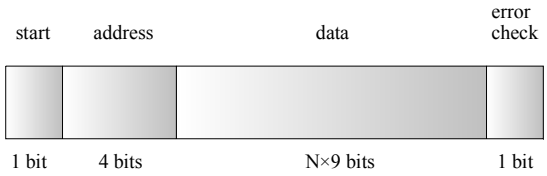

Figure 6. Serial data format.

V. DIAGNOSTICS, PROGNOSTICS, AND HEALTH MANAGEMENT

Diagnostics is defined as the process of detecting and isolating failures/anomalies in the operation and performance of a system or a component. Prognostics refers to the process of assessing the extent of performance degradation, such as deviation from an expected normal operating condition, and then predicting the future state of reliability based on actual and past monitored health conditions. The added capability to make appropriate decisions about maintenance actions based on diagnostics/prognostics information, available resources, and operational demand is referred to as health management.

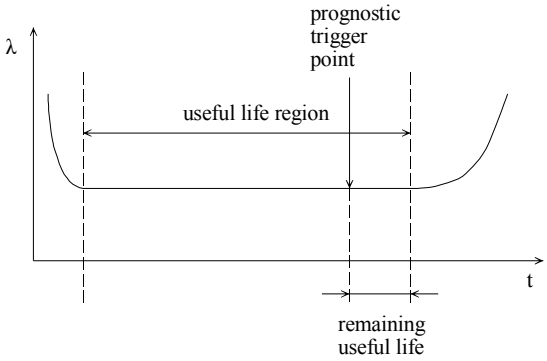

Figure 7. Illustration of precursor monitoring approach to prognostics.

One approach to prognostics is through monitoring and reasoning of parameters that are precursors to impending failures [8,9]. In a precursor monitoring approach, a failure can be predicted by correlating the change in the monitored precursor parameter with the impending failure. Different failure mechanisms are associated with different precursor parameters (i.e. failure signatures). An impending failure is

indicated when the monitored value is greater than a preset reference value. At a calibrated time or so-called prognostic distance, it will trigger before the onset of the end-of-life region, as shown in Figure 7.

A failure mode, mechanisms, and effects analysis (FMMEA) can be used to list critical failure mechanisms and identify the associated failure precursor parameters to be monitored.

An example of a precursor parameter is the collector-emitter saturation voltage of an IGTB [10]. The failure of bond wires causes a change in either the contact resistance or the internal current distribution, such that it can be identified by monitoring the collector-emitter saturation voltage. Besides the collector-emitter saturation voltage of the IGBTs, other precursor parameters have been listed and discussed in section III.

Assessing the health of a system provides information that can be used to provide warnings in advance of catastrophic failure and predict when maintenance should be scheduled based upon the evidence of need.

Each hardware manager, in the control architecture detailed in section II, communicates the monitored precursor parameters to the application manager. The application manager eventually interprets the received data and evaluates the health of the corresponding individual cells. It also decides if an "unhealthy" cell needs to be taken out of order and coordinates the reconfiguration of the Marx modulator.

VI. CONCLUSIONS

This paper has presented the concepts of a hierarchical control architecture for a PEBB-based ILC Marx modulator. The functional partitioning of the control of the modulator into (three) hierarchical layers; system layer, application layer, and PEBB layer was discussed. An interface characterisation was given. A conceptual design of the hardware manager was presented. The idea of prognostics was briefly explained.

REFERENCES

[1] T. Ericsen, "Power Electronic Building Blocks - A systematic approach to power electronics," in *Proc. Power Engineering Society Summer Meeting*, Seattle, WA, July 16-20, 2000, pp. 1216-1218.

[2] T. Ericsen, Y. Khersonsky, P. Schugart, and P. Steimer, "PEBB - Power Electronics Building Blocks, from concept to reality," in *Proc. International Conference on Power Electronics, Machines and Drives*, Dublin, Ireland, Apr. 4-6, 2006, pp. 12-16.

[3] F. Wang, S. Rosado, and D. Boroyevich, "Open modular power electronics building blocks for utility power system controller applications," in *Proc. IEEE Power Electronics Specialists Conference*, Acapulco, Mexico, June 15-19, 2003, pp. 1792-1797.

[4] I. Celanovic, I. Milosavljevic, D. Boroyevich, R. Cooley, and J. Guo, "A new distributed digital controller for the next generation of power electronics building blocks," in *Proc. IEEE Applied Power Electronics Conference and Exposition*, New Orleans, LA, Feb. 6-10, 2000, pp. 889-894.

[5] I. Celanovic, N. Celanovic, I. Milosavljevic, D. Boroyevich, and R. Cooley, "A new control architecture for future distributed power electronics systems," in *Proc. IEEE Power Electronics Specialists Conference*, Galway, Ireland, June 18-23, 2000, pp. 113-118.

[6] G.W. Francis, R.P. Burgos, I. Celanovic, F. Wang, and D. Boroyevich, "A universal controller for distributed control of power electronics systems in electric ships," in *Proc. American Control Conference*, Portland, OR, June, 8-10, 2005, pp. 1999-2004.

[7] S. Rosado, F. Wang, D. Boroyevich, and R. Wachal, "Control interface characterization of power electronics building blocks (PEBB) in utility power system applications," in *Proc. IEEE Power Engineering Society General Meeting*, Toronto, ON, July, 13-17, 2003, pp. 1350-1355.

[8] N.M. Vichare and M.G. Pecht, "Prognostics and health management of electronics," *IEEE Transactions on Components and Packaging Technologies*, vol. 29, pp. 222-229, Mar. 2006.

[9] C.Y. Yin, H. Lu, M. Musallam, C. Bailey, and C.M. Johnson, "A prognostic assessment method for power electronics modules," *in Proc. Electronics System-Integration Technology Conference*, London, UK, Sept. 1-4, 2008, pp. 1353-1358.

[10] N. Patil, D. Das, K. Goebel, and M. Pecht, "Failure precursors for insulated gate bipolar transistors (IGBTs)," *in Proc. International Seminar on Power Semiconductors*, Prague, Czech Republic, Aug. 27-29, 2008, pp. 107-112.