

Development of an Electronic Control Unit for PMSM Drives in Automotive Applications

Burak Soner, Arif İçke, Ahmet C. Mert, Umut Başaran, Serkan T. İmram and İlker Şahin

AVL Research and Engineering, Istanbul, Turkey
burak.soner@avl.com

Abstract

This paper presents the development and design of an Electronic Control Unit for permanent magnet synchronous motor (PMSM) drives for automotive applications, specifically for electrical traction drives of hybrid/electric vehicles. Hardware design of the electric motor control unit (EMCU) complies with the automotive hardware design standards. The software architecture of the EMCU is an AUTOSAR compliant real-time operating system (RTOS) based structure which enables easy integration of different function components and ensures safe hard real-time operation. Together with the RTOS on the 32-bit main microcontroller, an 8-bit support microcontroller is used to implement the functional safety requirements derived from analysis according to the ISO26262 automotive standard. The modular software structure also decouples the algorithm and interface development stages, making both hardware and software development processes less error-prone. To test the EMCU, a generic vector control scheme is implemented on the PMSM. Experimental results presented in the paper are acquired from tests on a full-sized ETAS HiL simulator setup consisting of a real-time PC and a configurable FPGA-based PMSM emulator board.

1. Introduction

As a result of the fast depletion of fossil resources and significant environmental damage caused by pollutants emitted by conventional fossil-fueled vehicles, emission regulations all over the globe are getting tighter and tighter. As a solution to this, the automotive industry has introduced the electrification of conventional traction systems into vehicle design [1]. Ranging from hybridization of conventional vehicles to developing full-electric vehicles, this solution pushed the automotive industry to merge with another huge field of study: electric drives and safe embedded systems that control them.

The implementation of control strategies in the digital domain is often done on embedded systems consisting of microcontroller units (MCU) and digital signal processors [2, 3]. The idea of using FPGAs is also a consideration due to the fact that FPGAs reduce processing and peripheral operation times significantly, moving the digital system closer to the concept of “real-time” [4]. Due to the high number and variety of tasks that need to be accomplished and the safety considerations associated with them in present day vehicles, automotive embedded systems need to be safe hard real-time systems [5, 6]. The main standard which governs this criterion for road vehicles is the ISO26262 [7]. Complying with this standard creates requirements that need to be met by both the hardware and

software. In terms of hardware, automotive-specific components and architectures are used. To satisfy the hard real-time criterion in terms of software, a generic software architecture for automotive embedded systems has been proposed. The AUTOSAR architecture allows for a modular and diagnosable embedded software which creates a real-time environment (RTE) for control tasks. [8]

The electric machines that these hard real-time embedded systems control are usually either induction machines or synchronous machines [9]. Permanent magnet synchronous motors (PMSM) are very frequently used in electrical traction systems thanks to their high efficiency, torque density, low maintenance and instrumentation needs and also ease of control [10]. Vector control strategies have been extensively developed for PMSM drives [11, 12].

To decrease the effort and cost of testing embedded systems and control algorithms running on them, hardware in the loop (HiL) setups are used [13, 14]. HiL setups consist of an interface to the hardware and a RTOS based computer which emulates the rest of the system and employs a graphical user interface for the tester. This vastly reduces testing and verification effort both in terms of time and money, since, expensive and hard-to-debug hardware components that are susceptible to damage, like motors, injectors, inverters etc., are thoroughly modeled and emulated by easily accessible software.

This paper presents the total design and development of an automotive-grade electronic motor control unit (EMCU) for PMSM drives starting from hardware design up to the HiL testing and application stage. Experimental results for the application of a PMSM vector control scheme on the EMCU are also presented. The hardware design of the electronic board is in compliance with the automotive standards and is based on one main MCU, one support MCU and the related peripherals used in control applications. The software architecture is an AUTOSAR based real time operating system (RTOS) structure which is modular, reliable and configurable. The dual MCU structure has been employed to implement functions to meet the safety requirements put forth by the analysis according to the ISO26262 standard. A vector control algorithm for the PMSM has been implemented on the EMCU. Performance tests for the application have been conducted in real-time on an ETAS HiL setup consisting of a real-time PC and an FPGA-based PMSM emulator board.

2. EMCU System Overview

The EMCU is the control board of a PMSM inverter, which also encloses a power board consisting of IGBTs and DC link capacitors. The main task of the EMCU is to accomplish torque and speed control of a PMSM, which is designed to be

connected to the traction system of a hybrid/electric vehicle. The main reasons why PMSMs are widely used in traction systems are their high efficiency and the well-developed control and drive systems for PMSMs.

2.1. Control System Overview

As shown in Fig. 1, in order to implement torque and speed control for a PMSM, the EMCU needs various information from the DC/DC converter, inverter and PMSM. It also needs to know the state of the ignition switch of the vehicle. Some of this information is conveyed to the EMCU via the CAN-bus or serial protocols like the serial peripheral interface (SPI), while some of other parts of that information are received via analog and/or digital inputs. A resolver interface is also included on the EMCU to get position information from the PMSM. EMCU outputs PWM signals that are used to drive the gates of the IGBTs of the 3-phase inverter. Utilizing a vector control scheme, the EMCU applies speed and torque control to the PMSM.

2.2. Safety Architecture and Automotive Standards

The EMCU contains two MCUs that communicate with each other and implement certain functional safety tasks. These tasks include monitoring of system voltages, monitoring the task sequencer of the main MCU and watchdog duties, which are derived from functional safety requirements resulting from analysis according to ISO26262. These tasks are defined jointly by the hardware and software design since they are very low-level in terms of system hierarchy. Also, to comply with other automotive standards, hardware design and component choice needs to be done accordingly.

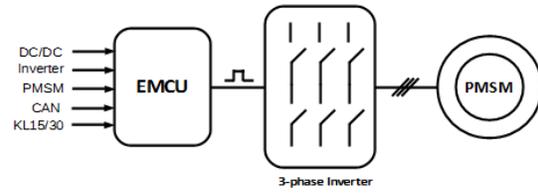


Fig1. Block diagram of the EMCU-PMSM system

3. Hardware Design

The block diagram of the EMCU hardware is illustrated in Fig. 2. The EMCU comprises of a 32-bit main controller, an 8-bit support MCU, two power supply units (PSU1 and PSU2), isolators, a complex programmable logic device (CPLD), and interfaces for resolver, sensor and CAN communication.

The power supply unit PSU1 which supplies majority of the components on the EMCU has an integrated Buck converter which converts a battery voltage of 12V to 5.5V with high efficiency. A stable and low noise voltage supply is a must-have for sensitive circuits like the sensor interfaces and the analog-digital-converters (ADC) of the MCUs to minimize measurement noises. This is accomplished by the integrated low-dropout regulators (LDO) of the PSU1 connected to the output of the Buck converter. These LDOs provide supply voltages ranging from 1.3 V to 5.0 V. The power supply unit PSU2 is simply an LDO that supplies only the CPLD and a part of the isolators. Since the current consumption of the CPLD and the isolators is low, the power efficiency of the PSU2 is not a major concern.

With the total control software embedded in its flash

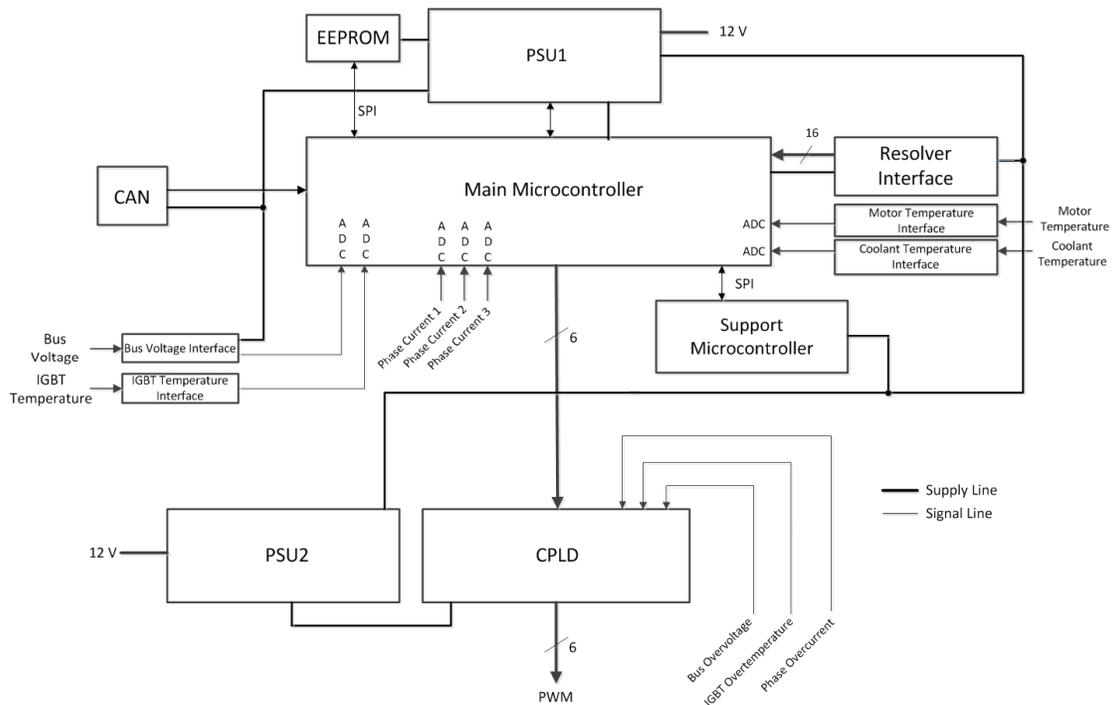


Fig2. Block diagram of the EMCU hardware



Fig 3. Photograph of the EMCU

memory, the main MCU runs the motor control algorithms to generate the PWM signals that are buffered by the CPLD and are sent to the gate driver of the inverter mounted on a separate printed circuit board (PCB). The CPLD deactivates the PWM signals in case of errors like over-temperature, over-current or over-voltage in the inverter. Owing to its simple logic gate based structure, CPLD responds much faster to these errors than the MCU. To guarantee a flawless operation of the system, the support MCU monitors not only the supply voltages but also the operational integrity by a question/answer structure that it shares with the main MCU over a shared SPI bus, generating a system reset in case of an error. The supply voltages are also monitored by the PSU1. Moreover, the PSU1 has a simple integrated watchdog which monitors the main MCU which needs to send a periodic falling edge trigger signal to the watchdog input of the PSU1. Failing to do so results in a system reset generated by the PSU1.

The resolver interface is a 10-bit to 16-bit ADC with two analog inputs. It incorporates a sine wave generator with a maximum programmable frequency of 20 kHz connected to the primary windings of a resolver mounted in the PMSM. Using its two secondary windings configured at 90 degrees from each other, the sine wave generated by the resolver interface is amplitude modulated with the sine and cosine of the mechanical angle of the rotor [15]. These two orthogonal signals are then fed to the analog inputs of the resolver interface, digitized by the ADCs of the resolver interface to calculate the rotor position and speed. The digital outputs of the resolver are directly connected to the general purpose input of the MCU.

Although the EMCU and the inverter are mounted on separate PCBs, large inverter currents might cause ground shifts and the interferences stemming from the inverter PCB can be coupled to the EMCU via the wiring and the connectors, negatively affecting system performance. To alleviate this problem two separate ground domains have been implemented on the PCB of the EMCU. To establish the signal transfer between those two ground domains, digital and analog isolators preventing ground coupling are used.

Except the connectors, all components of the EMCU comply with the automotive industry standards, AEC-Q100, AEC-Q101 and AEC-Q200 [16]. A picture of the EMCU is shown in Fig 3.

4. Software Architecture and PMSM Vector Control

The modular software architecture of the EMCU consists of three parts: the application software layer (ASW), the basic software layer (BSW) and the hardware abstraction layer

(HAL). ASW is processor-independent and consists of high level algorithms developed for the control and monitoring of the physical system, including the vector control algorithm for PMSM. The BSW consists of an AUTOSAR based RTOS and processor-specific peripheral drivers that wrap and simplify the low-level processor commands. The HAL maps the high-level ASW commands to the BSW functions, acting as a translator between the two.

4.1. Modular Software Architecture

In order to operate as a reliable hard real-time system, the EMCU software utilizes an AUTOSAR based RTOS as its lowest level scheduler. To make the software modular and configurable, the HAL which acts as a communication layer between the BSW and the ASW was used. Configurability and modularity of embedded software is especially important in automotive software development since the control software needs to be safe and in some means reconfigurable. With the HAL, the ASW development and the processor-specific BSW development is decoupled, thus fulfilling the above mentioned criteria.

AUTOSAR also brings a standardized diagnostics scheme and easy integration of separate application components. The “Diagnostics Event Manager” is specifically developed for automotive ECUs and is thoroughly documented in AUTOSAR specifications. Different functionalities like regenerative braking strategies, acceleration pedal position monitoring etc. can be developed in nuclear fashion and integrated into the ASW. The HAL reads the high level commands coming from the ASW (e.g. PWM duty cycle values) and translates those into the low-level commands “understandable” by the processor peripheral drivers in the BSW (e.g. comparator register values for the PWM timers to toggle the digital outputs of the PWM pins).

The RTOS initializes the processor, utilizes the basic software drivers for the processor peripherals and schedules everything the processor does. It generates multiple fixed-time “tasks” (also called interrupts) in which the ASW control algorithms can run. The fact that the timing of these tasks are strictly fixed enables developers to discretize the continuous time control algorithms and implement them on the MCU with a sampling time of their choice. In automotive embedded systems, due to the software safety standards, there are also monitoring functionalities that require relatively slow tasks with sampling times in the range of 1-100 ms, while software components like the current control loop run inside tasks with smaller sampling times, typically 100us.

4.2. Vector Control of PMSM

The schematic overview of a vector control based ASW is depicted in Fig 4. Depending on the type of the reference signal coming from the CAN network, the reference signal is considered either as a speed or torque set-point. After being processed by a temperature-based torque derating function, the limited or derated torque or speed set-points are evaluated by the current set-point assignment subsystem in which the direct and quadrant axes current references are determined by motor-specific pre-defined maps with inputs of torque reference. Current references in d-q frames are then directed to the current controller which involves two proportional-integral controllers for control of the current in each axis. Voltage references

determined by the current controller are transformed into stationary alpha-beta reference frame to be processed by the duty cycle assignment block to generate Space Vector Modulation based PWM waveforms for each phase. The current control loop is closed with the current and position feedback measured from the plant. Measured three phase current signals are then transformed into rotating reference frame to be used as the feedback by the current controller which could be considered as the core part of the vector control loop.

5. HiL Testing and Results

HiL testing is one of the increasingly used methods for the development and testing of automotive systems [17]. Within this approach, the hardware to be tested is surrounded with a test harness, where the other system components are either modeled or emulated by the HiL platform. This RTE ensures high-fidelity results with high resolution in terms of both sampling time and accuracy of emulated measurement signals. For the testing of the EMCU, a HiL test rig was designed (Fig. 5). The EMCU is connected to other vehicle components via harnesses as if it were running on a real vehicle. A host PC is responsible for both the calibration of the EMCU parameters and the configuration and test execution on the HiL system. The HiL test-rig consists of the following components:

- CAN module which emulates vehicle communication network,
- FPGA based electric drive simulation board that simulates all typical signals required for testing inverter and electric motors,
- Real-time PC, which is connected to host PC via LAN, running the C-code of the plant model
- 12 V Power supply to activate the EMCU board.

Plots for a sample test case experimented on the HiL test rig are depicted in Fig. 6. The test case consist of a speed ramp up with the torque set-point kept constant at 100 Nm while the mechanical speed of the virtual dynamometer of the real-time electric motor model is gradually increased to 7000 rpm. Depending on the torque reference and measured mechanical speed, d-q axes current references are instantaneously extracted out of pre-defined tables while the closed-loop current controller

produces d-q voltage signals to achieve those current set-points. It should be noted that as the speed reaches approx. 5000 rpm, it becomes difficult to maintain 100 Nm torque outcome and the measured torque starts to decay. This is due to full utilization of the DC link voltage caused by the expected phenomenon of back-EMF voltage which expands proportionally with the rotational speed. Experiments conducted on the test rig proved that the overall EMCU device operates with satisfactory performance fulfilling automotive related safety requirements.

6. Conclusions

Development and testing of an automotive-grade EMCU for PMSM drives was presented in this paper. Utilizing an electronic hardware design that complies with automotive hardware and safety standards, an AUTOSAR based RTOS runs on the main MCU, ensuring a safe hard RTE for control algorithms. The BSW-HAL-ASW structure of the software enables decoupled development of the control algorithm and the board specific BSW functions. A vector control algorithm for PMSM has been implemented in the application layer. An experimental setup consisting of a full-size ETAS HiL simulator, a real-time PC, monitoring environment and an FPGA-based PMSM emulation board was used to obtain results for the performance of the EMCU on the vector control algorithm. The results show EMCU successfully implements speed and torque control and complies with automotive standards.

7. Acknowledgements

This research was supported by the TEYDEB 1501 funding scheme of the Scientific and Technological Research Council of Turkey (TUBITAK).

8. References

- [1] A. Emadi, Y.J. Lee, K. Rajashekara, "Power Electronics and Motor Drives in Electric, Hybrid Electric, and Plug-In Hybrid Electric Vehicles", *IEEE Trans. Ind. Electron.*, vol. 55, no. 6, pp. 2237-2245, Jun, 2008.

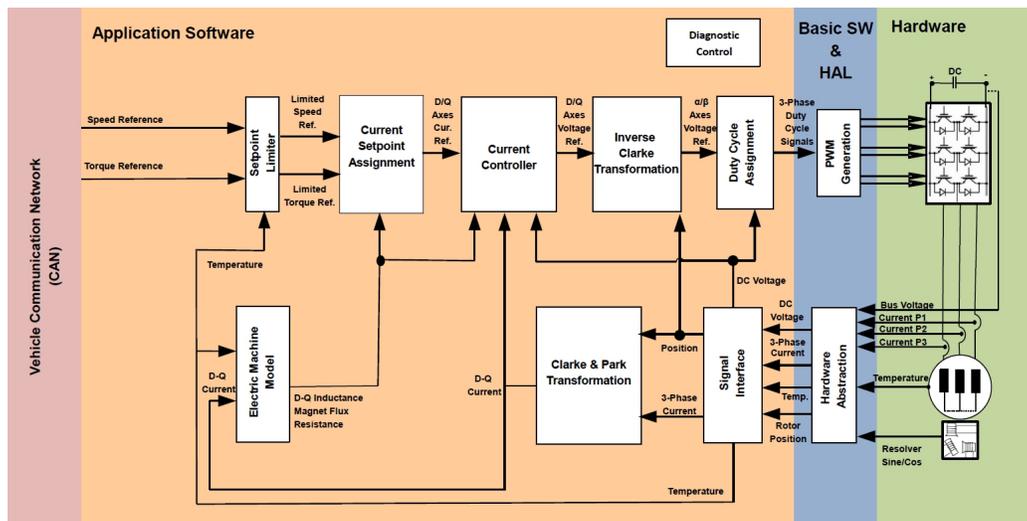


Fig 4. Block diagram of the vector control system

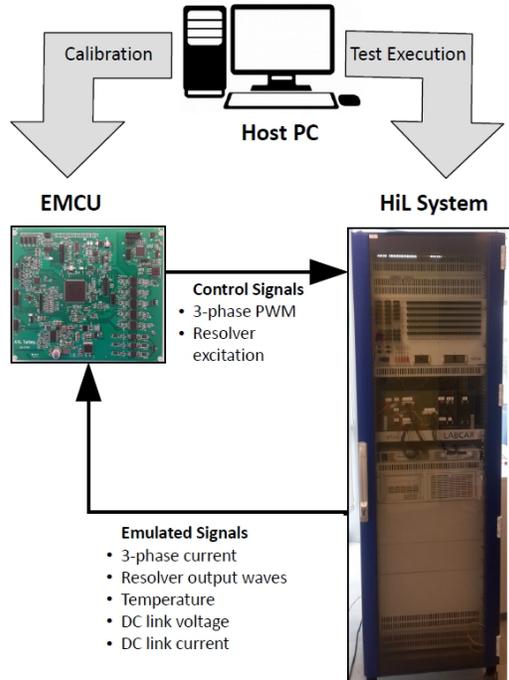


Fig 5. HiL Setup

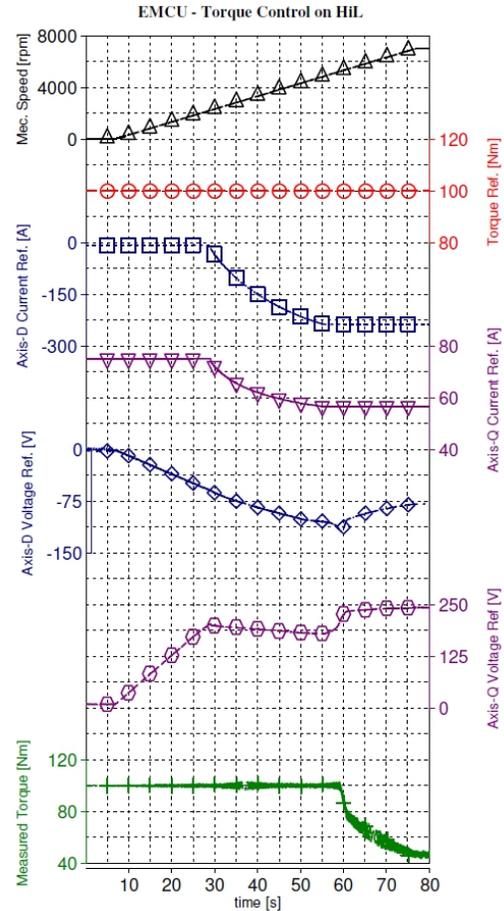


Fig 6. Experimental results for torque control

[2] O. Larses, "Architecting and Modeling Automotive Embedded Systems", PhD. thesis, Dept. of Machine Design, KTH, Stockholm, Sweden, 2005.

[3] G. Leen, D. Heffernan, "Expanding Automotive Embedded Systems", *Computer*, vol. 35, no. 1, pp. 88-93, Jan, 2002.

[4] F. Salewski, S. Kowalewski, "Hardware/Software Design Considerations for Automotive Embedded Systems", *IEEE Trans. Ind. Inform.*, vol. 4, no. 3, pp. 156-163, Aug, 2008.

[5] D. Chen, R. Johansson, H. Lönn, Y. Papadopoulos, A. Sandberg, F. Törner, M. Törngren, "Modelling Support for Design of Safety-Critical Automotive Embedded Systems", *Computer, Safety, Reliability, and Security, SAFECOMP '08*, UK, 2008, pp. 72-85.

[6] M.G. Harbour, M.H. Klein, J.P. Lehoczky, "Timing Analysis for Fixed-Priority Scheduling of Hard Real-Time Systems", *IEEE Trans. Software. Eng.*, vol. 20, no. 1, pp. 13-28, Jan, 1994.

[7] *Road Vehicles – Functional Safety*, ISO26262, 2011.

[8] AUTOSAR (2011, Nov 23). *Demonstration of Integration of AUTOSAR into a MM/T/HMI ECU (v1)* [Web]. Available: http://www.autosar.org/fileadmin/files/releases/AUTOSAR_TR_IntegrationintoMMTHMIECU.pdf

[9] J. Puranen, "Induction Motor Versus Permanent Magnet Synchronous Motor In Motion Control Applications a Comparative Study", PhD thesis, Dept. of Elec. Eng., LUT, Lappeenranta, Finland, 2006.

[10] F. Demmelmayr, M. Troyer, M. Schroedl, "Advantages of PM-machines Compared to Induction Machines in Terms of Efficiency and Sensorless Control in Traction Applications", *IECON 2011 – 37th Annual Conference on IEEE Industrial Electronics Society*, pp. 2762-2768, Melbourne, VIC, 2011.

[11] P. Pillay, R. Krishnan, "Modeling, Simulation, and Analysis of Permanent-Magnet Motor Drives. I. The Permanent-Magnet Synchronous Motor Drive", *IEEE Trans. Ind. Applicat.*, vol. 25, no. 2, pp. 265-273, Mar/Apr, 1989.

[12] L. Zhong, M.F. Rahman, W.Y. Hu, K.W. Lim, "Analysis of Direct Torque Control in Permanent Magnet Synchronous Motor Drives", *IEEE Trans. Pow. Elec.*, vol. 12, no. 3, pp. 528-536, May, 1997.

[13] P. Baracos, G. Murere, W. Jin, "Enabling PC-Based HiL Simulation for Automotive Applications", in *IEMDC 2001 – IEEE International Electric Machines and Drives Conference*. Cambridge, MA, 2001, pp. 721-729.

[14] S. Raman, N. Sivashankar, W. Milam, W. Stuart, N. Sabi, "Design and Implementation of HiL Simulators for Powertrain Control System Software Development", in *American Control Conference, 1999*, San Diego, CA, 1999, pp. 709-713, vol. 1.

[15] Automotive Electronics Council, AEC Documents, August 2013, <http://www.aecouncil.com/AECDocuments.html>

[16] Texas Instruments, Application report SPRA605, 2000

[17] B. Lu, X. Wu, H. Figueroa, A. Monti, "A Low-Cost Real-Time Hardware-in-the-Loop Testing Approach of Power Electronics Controls", *IEEE Trans. Ind. Electron.*, vol. 54, no. 2, pp. 919-931, Apr, 2007.