Design of a Real-time Simulator Capable of Hardware-in-the-loop Simulation for an Automated Collision Prevention (ACoP) System for an Autonomous Electrical Vehicle

Idrees Alzahid, Fnu Qinggele, and Yong-Kyu Jung
{Alzahid001, qinggele001, and jung002}@gannon.edu
Electrical and Computer Engineering, Gannon University, PA, USA

Abstract

A real-time simulator is an essential means for swiftly designing and verifying embedded time-sensitive and/or real-time applications especially in automotive and transportation. A cost-effective real-time simulator platform for research in automotive electronics has been developed and performed both hardware-/software-in-the-loop (HIL/SIL) simulations of an automated collision prevention (ACoP) system. The multi-core real-time simulator platform was developed for intuitively and swiftly managing different complexity levels and design scales while satisfying real-time constrains with sufficient accuracy of the parallel HIL simulation of models and/or hardware components. The real-time simulator platform was evaluated with the ACoP system integrated to the C/VHDL models of the electric vehicle. We evaluated the HIL simulation with 50 μs real-time clock resolutions. The HIL simulation achieved 2.5x faster acceleration and deceleration of engine (i.e., motor) than the SIL simulation while maintaining 0.3% of the HIL simulation difference of the speed overshoot and undershoot compared with an ideal Simulink simulation.

I. Introduction

Contemporary transportation systems, such as automobiles and locomotives, are necessary to employ embedded electronics and computing systems for satisfying the rapidly increasing complexity and accuracy of demanding requirements. In addition, research and engineering society has been challenged to convey such swiftly evolving transportation systems under the tight time-to-market pressure. Real-time simulation-based prototyping is one of the proven solutions for embedded system developers in automotive industry. Therefore, a real-time simulator with hardware-in-the-loop (HIL) simulation [1, 2, 3] capability must address the seamless integration of various complex subsystems and the accurate real-time simulation capability with existing and developing hardware and software modules while continuously supporting developers with an intuitive but precise design refinement and effective evaluation.

Various sensors have been employed for electronics and computing parts of automotive and transportation [4]. Cameras are popular for visual identification of the objects surrounding a vehicle. The GPS systems installed in a vehicle are aimed for navigating and positioning purposes. In particular, an automobile is one of the high potential candidates for applying Internet-of-Things (IoTs), which gather various forms of information from numerous types of sensors via wireless communication and process the information on embedded or application specific processing engines [5].

Efficient and perceptive HIL simulations are generally require swift integration, flexible extension, input/output configuration, precise execution, and systematic verification. Since the HIL simulation permits models of a part of the system to be simulated in real time with the actual hardware of the remainder of the system, developers can promptly evaluate hardware and software subsystems of the electric vehicles (EVs), including a control strategy, various I/O interfaces, different signaling, and signal conditioning. Application-specific real-time simulators, therefore, were developed for different applications, including electric control units in EVs [6], fuel cells in hybrid EVs [7], and electric and hydraulic systems in avionics [8].

As the demand for real-time simulators increases in industry, significant growth in the number of RT-Sims has been evident during the last decade in academia [9]. Unlike industry, academic version of real-time simulators [10, 11] are expected to embrace specific features including meaningful and relevant experience without being limited by laboratory equipment, user-friendly interfaces with increasing sophistication, the flexibility for continuous expansion, and preferred cost-effectiveness. We have developed a cost-effective academic real-time simulator (RT-Sim) with the HIL capability in order to successfully utilize the RT-Sim in academia, especially for different disciplines including Embedded Systems, and Communications [12]. In particular, the developed RT-Sim with the HIL simulation is beneficial to successfully perform automotive electronics research including an automated collision prevention (ACoP) system, which comprises of the hardware and software subsystems of a wireless sensor system (WiSS). Section 2 describes the architecture and operation of the ACoP. Section 3 expresses evaluation of the ACoP integrated to an EV model via a flexible wireless interface module for the rapid HIL simulations. The evaluation results and

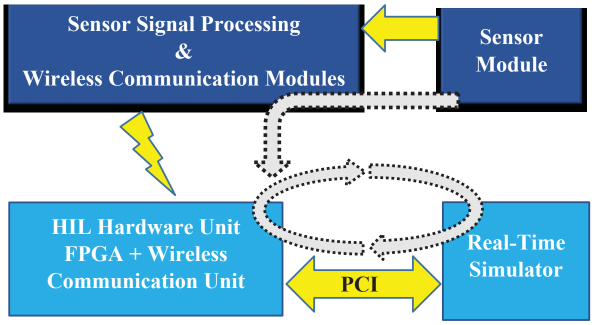

Figure 1. A Block Diagram of the Wireless Sensor System (WiSS) integrated for a Hardware-in-the-loop (HIL) Real-time Simulation

analysis of the WiSS on the EV are also described in Section 3. Section 4 depicts the conclusions.

II. Architecture and Operation of a HIL Real-time Simulation for Wireless Sensor System (WiSS)

Figure 1 illustrates a block diagram of the WiSS integrated for the HIL real-time simulation architecture and operation. The WiSS consists of a sensor module and a sensor signal processing and wireless communication module. The presented WiSS was implemented with ultrasonic sensors [13] for detecting and measuring objects, an Arduino board [14] for processing sensor signals, and transmitting information to and receiving it from the flexible wireless interface module consisting of a wireless integrated field-programmable gate array (FPGA) [15] for the accurate and swift HIL simulations. In order to establish a cost-effective wireless communication network, a pair of Xbee modules [16] is employed between the WiSS and the flexible wireless interface module.

As seen in Figure 1, the flexible wireless interface module is integrated to RT-Sim via PCI [17] with wires and to the sensor signal processing and communication module without wires. The HIL simulation with WiSS is an extended form of the HIL real-time simulation. A traditional HIL simulation can be found between the EV models running in the RT-Sim and a time-critical subsystem implemented in FPGA running on the interface module. The interface module encompasses a cost-effective wireless capability in order to provide a viable means for extending various sensor systems without further physical hardware modification of the current FPGA-based interface module for the presented HIL simulation. Packets of information generated by the sensor signal processing module are delivered through the wireless channels established between the sensor signal processing module and the interface module.

III. Design of An Automated Collision Prevention (ACoP) system with WiSS

The presented WiSS is designed for researching an ACoP system of an EV. The ACoP is an emergency driving assistant system that can provide a safety means for drivers and passengers in the EV under uncontrollable and/or sudden disrupted situations on the road. The ACoP system can automatically detect distance of any objects surrounding the vehicle. In addition, the ACoP system is capable of generating a warning alarm for the driver as well as possibly taking control over the EV to avoid collision between the EV and any other objects including other vehicles, unmovable objects, such as trees, street lights, and so on, as well as pedestrians. Therefore, the ACoP system can not only slow down and stop the EV, but also change the direction of the EV in order to prevent any collisions.

The ACoP system can consist of a hardware and software subsystems including the proposed WiSS. The WiSS is to detect objects and measure the distance to the objects. A hardware subsystem of the WiSS consists of (1) an ultra-sonic sensor and driving system, (2) a pair of wireless communication system (i.e., Xbee 2.4 GHz), (3) an interface control logic written in VHDL programmed to the FPGA module. A software subsystem includes Arduino script for operating the ultra-sonic sensor driving system. After the hardware and software partitioning and implementation of the subsystems, the WiSS was integrated and verified for exercising a top-down verification method and for rapid prototyping of

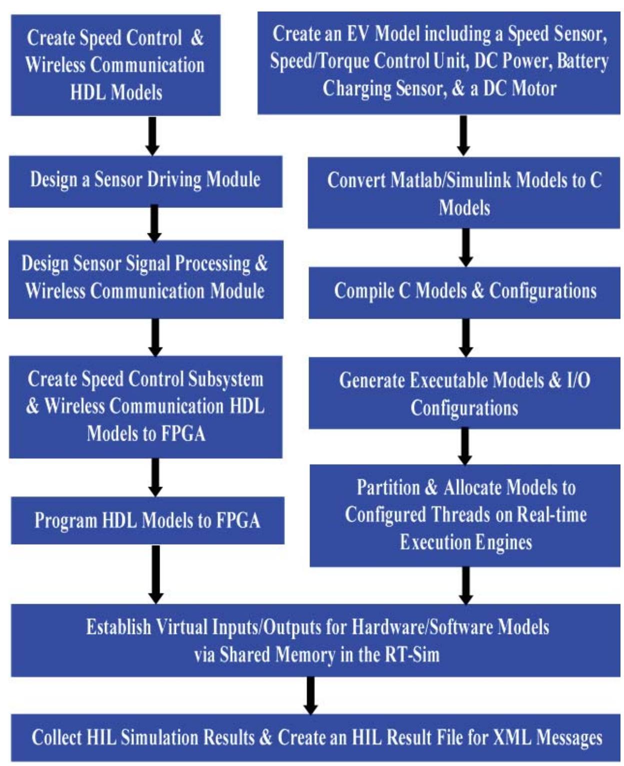

Figure 2. The WiSS Design and Operational Procedures for the HIL Real-time Simulations

the WiSS with the HIL real-time simulations. In order to expedite the HIL simulations, a series of the verification scenarios was developed and applied for developing a comprehensive ACoP algorithm that dynamically reflects various information on road size, road condition, driving direction, driving speed limit, and other information including weather and temperature via several expanded sensor and processing systems as a part of an IoT in future enhancement.

Figure 2 illustrates the WiSS design and operational procedure for the HIL simulation performed. The complete Simulink closed-loop EV model was designed. The EV consists of subsystems for sensing speed of EV and charging level of the battery, a control unit for controlling speed and torque of the DC motor, a battery for supplying DC power, and a DC motor for driving. The EV specifications includes a 25 horsepower, four quadrant operation wound DC motor, which is designed to execute in discrete time with a sampling rate of 1 micro-second. The speed control unit determines the speed of the armature. Inputs of the speed control unit are the desired speed in RPM, and the armature speed in RPM. Outputs of the speed control unit are the speed changes and armature currents. The speed control unit controls the armature current and prevents the current from surpassing the rated armature current.

The speed control unit receives the armature current, armature speed, speed change, and change in armature current, and generates the PWM pulses to set the armature voltage to the desired voltage in order to achieve the desired armature speed. The speed control unit also generates the control values for determining the PWM pulses. The DC Motor is powered by a 30 volt battery with a linear load torque. The DC Motor comprises a DC-DC converter connected to the PWM pulses to provide the desired armature voltage. I/Os of the motor are the torque load, PWM pulses, and battery voltage as the inputs and the armature current, armature speed, armature voltage, and the field voltage as the outputs.

The sensor module employs a ping ultra-sonic distance sensor to detect and measure the distance between the EV and any objects. The accuracy of the sensor employed is to measure the distance within 3 meters, which is about 3.3 yards. In addition, the sensor module is sufficiently durable for the outdoor usage. Since the sensor needs to interface to a microcontroller for further processing of the sensor outputs, an ATmega328 installed on an Arduino Uno, which is one the most popular for motor controlling applications, was selected. Analog and digital I/Os of the Arduino development board are used to integrate a cost-effective wireless device, such as an XBee S2 module. Therefore, a sensor signal captured and transmitted from the sensor module is processed by the microcontroller, which is also integrated to the wireless device.

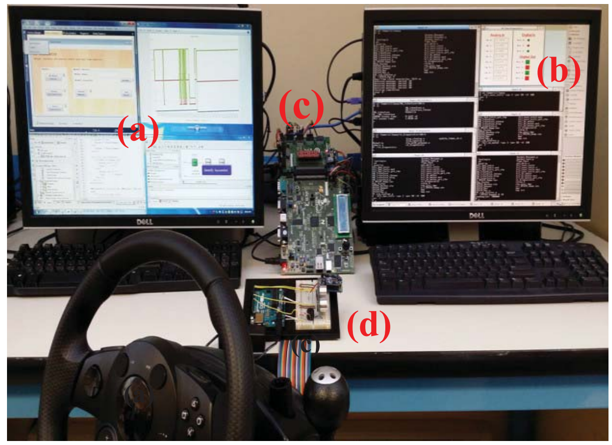

Figure 3. A WiSS Evaluation Setup for the HIL Real-time Simulations; (a) a console for user interface, (b) the RT-Sim developed, (c) the FPGA-based wireless interface module for HIL simulations, and (d) a prototype of the WiSS

The XBee device with its adapter offers an intuitive means of integration to the microcontroller in the Arduino Uno. The XBee device operating with 2.4 GHz clock establishes a wireless communication channel capable of interconnecting within 120 meters (i.e., 400 feet). A serial communication channel is established by the XBee devices between the WiSS and the FPGA-based interface module, which is attached to the RT-Sim for performing HIL simulations. A wireless communication HDL model was developed and programmed to the FPGA (i.e., Xilinx Spartan 3E). The inputs and outputs of the XBee devices are interconnected to I/Os of the FPGA.

IV. Evaluation of the ACoP System via the HIL Real-time Simulations

Figure 3 illustrates an overview of the HIL simulation setup for the WiSS. The RT-Sim we developed consists of three primary modules—(a) a console for user interface, (b) the RT-Sim for HIL simulations, and (c) an FPGA-based hardware interface module with wire/wireless connections for extended HIL simulations for WiSS. In particular, the FPGA-based hardware interface module with wireless capability provides additional flexibility to expand parallel and distributed HIL simulations without sacrificing a number of threads running in the RT-Sim. The WiSS consisting of the sensor and wireless signal processing modules is shown in Figure 3 (d).

A series of the WiSS evaluations has been developed based on the EV running on a highway. In order to develop the practical evaluation scenarios, we need to determine a few thresholds for identifying normal, warning, and extreme situations. According to

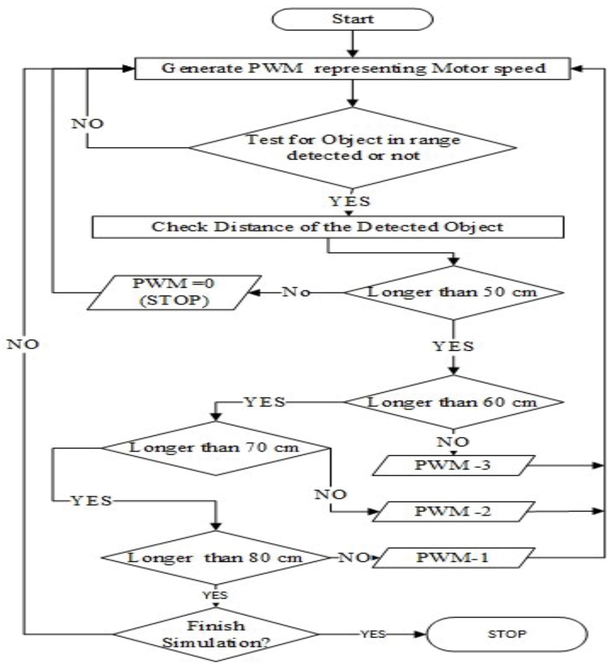

Figure 4. An Electric Vehicle Speed Controlling Flows based on a series of the evaluation scenarios for the WiSS

the federal highway administration, the width of a freeway road is 3.6 meters [18]. An average width (i.e., approximately 1.6 meters) of vehicles is used for our evaluations. The first warning threshold was determined by the distance between a vehicle and the EV equipped with the WiSS approaching the side or rear-end of the vehicle or vice versa. The distance identified for the warning is 80 cm. The other distances, such as the second warning and the extreme situation, are set in every 10 cm range. We assumed the EV can be driven faster than another vehicle can and the EV speed is 70 mph, which means the vehicle runs approximately 31.29 meters per second.

Figure 4 illustrates the different zones identified for the evaluations. In order to control the speed of the motor engine of the EV, four different acceleration and deceleration PWM pulse sequences are used as seen in Figure 4. Since the speed control depends on the initial speed, the primary PWM pulse sequences are accordingly modified. In addition, the PWM pulses are adjusted by receiving the feedback information transferred from the speed control unit running on the RT-Sim.

The HIL simulation period is 50 µs. The EV's maximum acceleration and deceleration cycles are measured as 3600 HIL simulation cycles. Thus, an acceleration of the motor engine can be completed within 180 ms. For instance, the EV driven in 70 mph moves 31.29 meter per second. The EV can move 0.156 centimeter per HIL simulation cycle, which is 50 µs. The sensor module was tested alone for evaluating accuracy of the distance measured. We obtain less than 2.89% error within 1 centimeter. For instance, about 3 distances measured over the same 100 distances were different by more than 1 centimeter. Therefore, our HIL simulation is sufficient in accuracy of the simulation.

In addition, the processed sensor data take a maximum of 5,355 µs for 100 cm and a minimum of 3,630 µs for 50 cm. We monitored packets delivered to the FPGA via XBee devices is an average of 100 bytes per packet. The XBee's data rate is 250 Kbps. Therefore, the wireless communication latency is 3,200 µs. We calibrated the maximum and minimum latencies of the sensor signals as 8,555 µs and 6,830 µs, respectively. Since there is another XBee wireless communication latency on the FPGA-based interface module, a range of the latencies of the WiSS is between 11,755 µs and 10,030 µs According to the signal latencies measured, the WiSS HIL simulations run between 200 and 235 HIL simulation cycles per sensor signal received from the sensor. Consequently, more than 200 requests from similar sensors can be processed concurrently by the presented RT-Sim with HIL simulations.

Figure 5 illustrates the WiSS HIL simulation results captured by the RT-Sim. An object was moved from the WiSS installed on the EV to measure the various distances. The measured distances reflect the sensor signals transmitted by the sensor module. The reactions of the EV are shown as three curves in Figure 5. The green curve represents actual speed (RPM) of the motor engine, the red curve shows tracking speed of the motor, and the blue curve illustrates final speed of the motor given at zero. The X-axis is the number of RPM samples

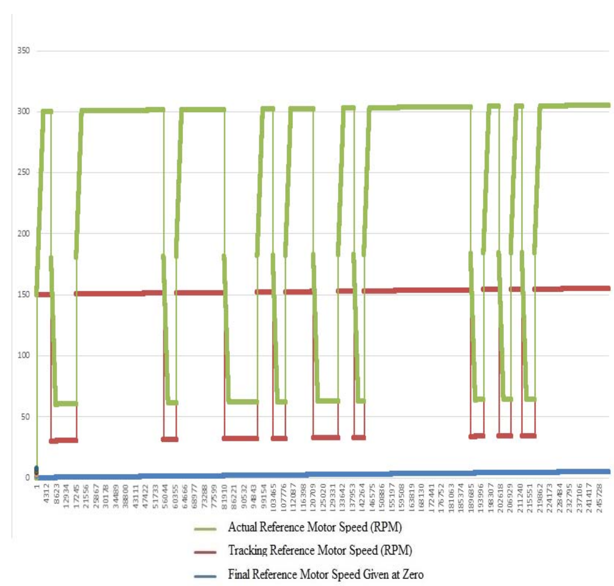

Figure 5. The WiSS HIL Simulation Results for Various Evaluation Scenarios; 250K samples are measured during the 5 second HIL simulation

collected during the 5-second HIL simulation. Total 250K samples were acquired. The Y-axis is the speed of the motor engine. The speeds (RPMs) were varied from 60 to 300 RPMs. The initial RPMs started at zero.

V. HIL real-time Simulations of the ACoP System with an Autonomous Electric Vehicle (AEV) Prototype

The ACoP system is implemented as an autonomous driving platform that is capable of performing wireless communication and data processing via a data processing and interface module for the HIL real-time simulations. The platform includes a four-wheel autonomous driving miniature vehicle assembled with a distance detector, a detection-angle controller, a motor driving module, and a wireless transmitter. Each of the sub-systems cooperates with each other under a microcontroller. The data processing and interface module consists of a wireless receiver and a processing terminal. The processing terminal handles graphical data and signal processing. All of the sub-systems in the platform are integrated to the real-time simulator.

The proposed platform operates autonomously while interacting with the real-time simulator via the wireless communication channel established between the platform and the real-time simulator via an FPGA-based HIL interface module. While the self-driving platform is moving on the testing ground, the distance detector scans the surrounding road-blocks and transfers the real-time driving circumstance to the DPI module. Then, it converts the data being collected to legible information and displays the information on the console monitor connected to the real-time simulator. At the same time, the light indicator on the FPGA-based HIL interface module blinks in accordance with specified rules to indicate successful data communication. The proposed concept of the platform-based HIL real-time simulation is to provide a viable and flexible means for swiftly designing and verifying important algorithms and components of the future automobiles in academic laboratories and design facilities.

The real-time simulation platform shown in Figure 1 has been evaluated for the HIL simulations with the ACoP system wirelessly connected to the FPGA module integrated to the two 14-bit output ADCs with 1.5 MHz sampling rate and 8 analog outputs (i.e., +/- 12 Volts), 3 digital outputs, 4 digital inputs (5 Volts), and a 12-bit resolution DAC with 6 analog outputs in the RTS via the voltage conversion units for various signaling standards including LVTTL, LVCMOS2/18, and TTL in the HIL module.

Three key models of an electrical vehicle—(1) DC motor with a battery, (2) current controller, and (3) speed controller—are designed as Simulink models, which are converted by the s-functions configured in RTS. In particular, the current controller model was modified for

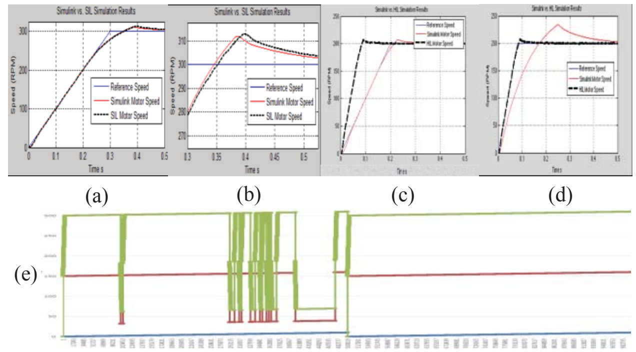

Figure 6. Engine (Motor) Acceleration HIL Simulation Results of an AEV Speed Controller: SIL vs Simulink (a) normal acceleration; (b) increased acceleration; & HIL vs Simulink (c) normal acceleration; (d) increased acceleration; and (e) ACoP HIL simulation.

the HIL simulation before distributing the executable codes of the associated models to three threads. The 0.625 MHz PWM signals are generated by a PWM generator implemented in the FPGA module.

The ACoP generates information on the objects detected and distances to the objects, processes the information with an Arduino board, and transmits the information via a 2.4 GHz operating Xbee adapter. The information is then received by another Xbee adapter installed to the FPGA module and passed to the ACoP algorithm implemented in HDL as an HW model for further processing with other HW models including the PWM generator.

As seen in Figure 6, the results of the Simulink, SIL real-time simulation, and HIL real-time simulation confirm that the RTS accurately simulated both of the HW/SW models. The simulation results are the reference speed and engine speed. The results, with a reference speed of 300 RPM, illustrate that the motor is able to achieve the desired speed within 0.3 seconds with a 10 RPM overshoot. These results are used as the baseline for the SIL and HIL evaluation of the real-time simulation. The overshoots measured in Simulink versus SIL and HIL simulations are 311 versus 312 and 205.8 versus 206.4. The percentage differences are 0.32% with SIL and 0.29% with HIL simulation. We discovered that the beginning of the SIL/HIL simulation between the virtual I/O and models are not synchronized in that the real-time simulation results are different within 0.1% of the Simulink simulation results. The results, however, still prove the accuracy of the real-time simulation. On the other hand, the acceleration times measured via the HIL and SIL simulation are more than 2.5 times different. Thus, the HIL real-time simulation proves to meet real-time constrains. Figure 4 (e) illustrates the engine speed control results of the ACoP HIL simulation. According to the distances detected from the objects, the engine (motor) speed decreased when the distance was within a warning zone and stopped when the distance crossed into the threshold of the collision but the driver did not reduce nor stop the vehicle.

The results of determining the access time of the virtual I/Os via the shared memory, PCI card, and PCI ADC in the RTS module were measured. An average of 0.1486 µs was for access data from/to the shared memory. The PCI access time measured was an average of 1.7889 µs, due to the bus used to access the PCI. The accessing ADC was identified as a critical path with an average of 13.2093 µs delay, which limits the faster model execution for the HIL simulation. The resolution of the real-time clock on the RTS is set to 10 µs. Since the most reliable time to consistently operate at the same timing interval was identified as 50 µs, the HIL execution on the RTS operates at a 50 µs for the real-time HIL simulation evaluation.

VI. Conclusions

A real-time simulation platform for automotive electronics is introduced for rapid and intuitive management, accurate simulation, and cost-efficient real-time environment for research and education. The real-time simulation platform is capable of executing both SIL and HIL simulations with sufficient accuracy in terms of real-time constrains and model operations. In addition, the real-time simulation provides a means to integrate existing design tools, such as Matlab/Simulink and FPGA-based platform to user developed HW/SW model configuration expansion, and simulation result management. A WiSS is presented as an extended hardware/software system for the hardware-in-the loop simulations of an autonomous electrical vehicle with the real-time simulator developed. In order to enhance the expandability of the HIL simulations, the FPGA-based interface module was upgraded with wireless connectivity. According to the analysis of the WiSS HIL simulations, up to 200 sensor systems can be simultaneously supported by the presented HIL simulation environment. The WiSS was successfully developed as a preliminary version of the ACoP system. A version of the ACoP system is successfully evaluated by utilizing external HW module wirelessly interfaced to the real-time simulation platform. In particular, the proposed real-time simulation efficiently schedules, distributes, interfaces, and simulates underlying SW and HW models and prototypes. Furthermore, the real-time simulation offers various wireless connections to users' hardware prototypes via a standardized wireless module. The presented real-time simulation platform evaluated the accurate (<0.3% error) HIL simulation and 2.5x faster operations with 50 µs real-time clock resolutions. The HIL simulation achieved 2.5x faster acceleration and deceleration of engine (i.e., motor) than the SIL simulation while maintaining the HIL simulation difference of the speed overshoot and undershoot compared with the Simulink simulations.

VII. References

[1] C. Dufour, S. Abourida, and J. Belanger, "Hardware-in-the-Loop Simulation of Power Drives with RT-LAB," IEEE PEDS pp. 1646-1651, 2005.

[2] R. McNeal and M. Belkhayat, "Standard Tools for Hardware-in-the-Loop (HIL) Modeling and Simulation," IEEE ESTS, pp. 130-137, 2007.

[3] O. Mohammed, N. Abed, and S. Ganu, "Real-Time Simulations of Electrical Machine Drives with Hardware-in-the-Loop," IEEE PESGM, pp. 1-6, 2007.

[4] P. Ranky, "Advanced Digital Automobile Sensor Applications," Sensor Reivew, Vol. 22, No. 3, pp. 213-217, 2002.

[5] D. Georgoulas and E. In-Motes, "A Real Time Application for Automobiles in Wireless Sensor Networks," MSR, Vol.3, No. 5, pp.158-166, 2011.

[6] L. Cheng and Z. Lipeng, "Hardware-in-the-Loop Simulation and Its Application in Electric Vehicle Development," IEEE VPPC, 2008.

[7] C. Dufour, J. Bélanger, T. Ishikawa, and K. Uemura, "Advances in Real-Time Simulation of Fuel Cell Hybrid Electric Vehicles," Proceedings of the 21st Electric Vehicle Symposium (EVS-21), April 2-6 2005.

[8] J. Casteres and T. Ramaherirariny, "Aircraft integration real-time simulator Modeling with AADL for architecture tradeoffs," IEEE DATE, 2009.

[9] P. Menghal and A. Jaya laxmi, "Real Time Simulation: A Novel Approach in Engineering Education," IEEE ECT, pp. 215-219, 2011.

[10] C. Dufour, C. Andrade and J. Bélanger, "Real-Time Simulation Technologies in Education: a Link to Modern Engineering Methods and Practices," ETE, 2010.

[11] S. Abourida, C. Dufour, J. Bélanger, and V. Lapointe, "Real-Time, PC-Based Simulator of Electric Systems and Drives," PST, pp. 1-6, 2003.

[12] T. Silloway, Y. Jung, D. Mackellar, F. Mak, "Design of a Real-Time Simulator for an Electric Vehicle," MSV, pp. 163-168, 2012.

[13] R. Dhole, V. Undre, C. Solanki, S. Pawale, "Smart Traffic Signal Using Ultrasonic Sensor," IEEE GCCEE, 2014.

[14] Jeremy Blum and Scott Fitzgerald, Exploring Arduino: Tools and Techniques for Engineering Wizardry, John Wiley & Sons, 2013.

[15] L. Junsong, et al, "FPGA Based Wireless Sensor Node with Customizable Event-Driven Architecture," JES, 2013.

[16] Telecommunications Weekly, "Digi Launches Next Generation XBee and XBee-PRO ZigBee Modules," 2010.

[17] Chan, Chris, "High-Performance PCI Card: Main/Lifestyle Edition," New Straits Times: 10. 2007.

[18] http://safety.fhwa.dot.gov/geometric/pubs/mitigationstrategies/chapter3/3_lanewidth.cfm, Website accessed on March 20th, 2015.