Original research article

Development of In-Motion Wireless Power Transfer Test Bed Platform for Wireless Electric Vehicle Charger

Watchara Amasiri* and Wanchai Pijitrojana
Department of Electrical and Computer Engineering, Faculty of Engineering, Thammasat University Rangsit Campus, Khlong Luang, Pathum Thani 12120, Thailand

Received 13 September 2016; Received in revised form 17 March 2017
Accepted 21 March 2017; Available online 30 June 2017

ABSTRACT

This paper describes a new method to model and measure in-motion wireless power transfer for a wireless electric vehicle charger to experimentally reinforce understanding of principles, characteristics, operation, communication and control of an in-motion wireless power transfer system. It uses a real-time target executable called 'external mode' in MATLAB/Simulink with Arduino and Wi-Fi-Shield, inexpensive and open source hardware. With an I/O real-time controller, real-time signal acquisition and real-time parameter tuning capabilities, the system can be advantageously used for experimenting, modeling, communicating and measuring. Step by step of how the proposed system may be designed for experimenting with a wireless electric vehicle charger is included. Relevant hardware usually found in the electronic market are also indicated.

Keywords: real-time target executable test bed; wireless electric vehicle charger; wireless power transfer test bed

Introduction

WIRELESS POWER TRANSFER plays an important role in charging electric vehicles (EV). Different wireless power charger structures for EVs have been proposed [1-5]. The most popular solution is fixed wireless charging, which requires EVs to be motionless for an inconveniently long time. John M. Miller et al. (2014) [1] presented experimental results on power smoothing of in-motion wireless EV charging. Six high-frequency coils, on-off sequencing controller, controllable power level, and connectivity between vehicle and grid site were included. Amateur radio equipment for experimenting with WPT has also been proposed. Sun-han Hwang et al., (2014) [6], presented an amateur radio transceiver ICOM IC-718, amateur radio linear amplifier Ameriton AL-811, and amateur radio automatic antenna tuner MFJ-993b as power source, high power amplifier, and impedance matching L-network, respectively. Power output could be achieved up to 220 watts with efficiency up to 95%, using cheap experimental equipment. System architecture was flexible and suitable for implementation. Many techniques for wireless power transfer efficiency enhancement have been proposed [7-8]. Nam Yoon Kim et al., (2012) [7], evaluated the design of a software-based

***Corresponding author:** w.a.th@ieee.org **doi:** 10.14456/tijsat.2017.15

wireless power transfer with closed-loop and open-loop control algorithm implemented in MATLAB&Simulink with data acquisition card to control source voltage to maintain the wireless power transfer efficiency.

A real-time target executable function, termed an external mode in MATLAB & Simulink, was selected as an essential part of software development for modeling and experimental monitoring. Fig. 1 shows a conceptual diagram of the real-time executable function in this system.

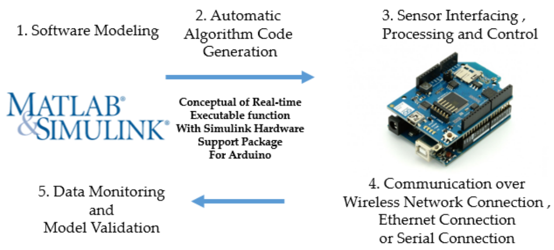


Fig. 1. Conceptual diagram of real-time executable function with Simulink hardware support package for Arduino.

External mode used a Simulink coder to dynamically link algorithm codes with I/O driver codes generated from the I/O block. It operated with the Simulink support package for Arduino hardware, automatically deploying algorithm code from Simulink to Arduino. This facilitated modeling of experimental systems run on inexpensive platforms. The Arduino Wi-Fi Shield was chosen to monitor and validate experimental results by wireless network connection. In this paper, software for modeling, measuring and controlling was built on an Arduino MEGA2560 with an Arduino Wi-Fi Shield and written in Simulink, using capabilities for real-time target executable and Simulink Coder for algorithm code generation. WPT experimentation and efficiency improvement techniques can be used through these platforms.

Test Bed Architecture

Software environment

Simulink is a block-based diagram environment for simulation and Model-Based Design (MBD). It also has automatic code generation, continuous testing and verification of embedded systems [9]. Simulink also provides a graphical editor, customizable block libraries, and solvers for modeling and simulating dynamic systems. Simulation and experimental results may be exported from Simulink for further analysis in MATLAB.

In 2013, the MathWorks Simulink Team published a Simulink Support Package for Arduino Hardware to develop and simulate algorithms running standalone on Arduino. This package also contains library blocks for configuring and accessing Arduino peripherals and accessing through Arduino Ethernet Shield and Wi-Fi Shield; there is also real-time parameter tuning and signal monitoring and automatic code generation for Arduino. This package is supported by most Arduino models including UNO, Mega 2560, Mega ADK, Leonardo, Micro, Mini, Pro and others. Table 1 lists supported functions in the Simulink support package for Arduino hardware and some relevant applications.

Hardware Environment

The Arduino MEGA 2560 is an essential part of this system. It is an open source microcontroller board based on the ATmega2560, which is a high-performance Atmel 8-bit AVR RISC-based microcontroller. The Arduino Wi-Fi Shield [10] is required to provide the wireless network connection needed for real-time parameter tuning and real-time signal monitoring in external mode. The Arduino Wi-Fi shield is an add-on peripheral compatible with most Arduino platforms, real-time signal acquisition, and real-time parameter tuning on Simulink.

Table 1. List of Supported Functions in Simulink Support Package for Arduino Hardware.

Arduino Peripheral	Simulink Supported	Example Application
Digital I/O	Yes	coil sequencing control
PWM	Yes	motor speed control
Serial communication	Yes	direct digital synthesis (DDS) module interface
TCP/IP communication	Yes	instrument interface
UDP communication	Yes	instrument interface
Wi-Fi Shield[a]	Yes	running external mode
Ethernet Shield[a]	Yes	running external mode

[a] official shields are recommended to be used with Simulink support package.

With analog to digital conversion (ADC) on the ATmega2560, analog voltage can be read. The Simulink support package for Arduino hardware provides library blocks for using this function.

The Simulink Support Package for Arduino Hardware also provides serial communication, Ethernet communication and Wi-Fi communication for commanding and controlling other equipment.

WPT Efficiency Measurement

This section recommends a methodology of measuring the power efficiency in WPT for EV.WPT functional cascade in [2] is applied to this system. It is well known that efficiency measuring in WPT needs to consider 1) HF power source, 2) WPT transmitting coil, 3) WPT receiving coil and 4) load. Overall efficiency is measured by checking the voltage and current of each separate section. Fig. 2 shows the WPT functional cascade accordingly.

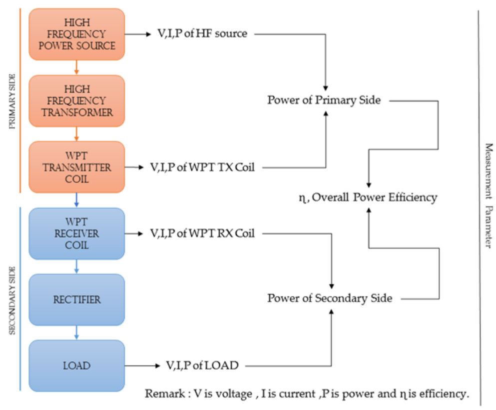


Fig. 2. WPT functional cascade used in this proposed system.

Voltage and current are measured using voltage and current sensing modules to be readable by the microcontroller. In this paper, a voltage transformer and a simple operational amplifier circuit measured voltage. The hall effect current sensor with over current fault output measured in each section. Voltage and current sensor output had a positive slope proportional to voltage and current flow. This measurement technique is versatile, and is applicable to other sections of the proposed system.

When the voltage and current in each section are measured, overall efficiency η, can be calculated by (1)

$$\eta = \eta_{sec} \times \eta_{pri} \tag{1}$$

where

$$\eta_{sec} = \frac{P_L}{P_{RX}} \tag{2}$$

$$\eta_{pri} = \frac{P_{TX}}{P_S} \tag{3}$$

in which, P_L, P_{RX}, P_{TX} and P_S are load power, WPT RX coil power, WPT TX coil power, and HF source power, respectively.

Design Procedure

System modeling, model deployment

and evaluation are described below:

1) Experimental design: design experiments that are possible with the system architecture. Activities should depend on hardware range, experimental purpose, and teaching goals.

2) Building the Model: build the model determined by block diagram with predefined library blocks in Simulink describing system behavior.

3) Validate the Model: generate input signals with source library blocks in Simulink or predefined input data in MATLAB, simulating dynamic behavior by choosing a fixed step solver in Simulink. Then validate the model algorithm by analyzing results as the simulation runs.

4) In the Real world: configure input and output behavior of Arduino peripherals, access real input signal and control output behavior with library blocks in the Simulink support package for Arduino hardware. Deploy the model to Arduino in external mode. Monitor and analyze the model behavior with real-time capabilities. Validate and tune interesting parameters with real-time parameter tuning.

5) Test and evaluate experimentally: create a test bed user interface, evaluate the implemented experimental model and algorithm performance.

When the design procedure goal point has been achieved, Arduino may work on a standalone basis without signal monitoring or parameter tuning capabilities. To do so, select simulation mode in Simulink as normal mode.

Examples

In this section, the authors discuss and show an example for which the proposed system is generally suitable and also those where architecture can be used. We start by simple experimentation such as EV motion control and coil sequencing control.

EV motion control and coil sequencing control for the grid site are necessary for the in-motion wireless power transfer test bed. This makes it possible to conveniently operate the entire test bed system simultaneously. The implemented models of moving control for EV and coil sequencing control for the grid site are described separately.

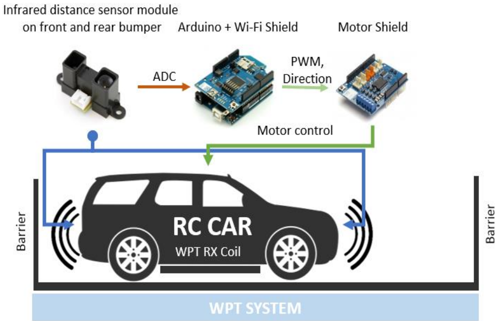
EV Motion Control

The whole system of controlling the electric vehicle movement must be considered. The procedure outlined in Section 3 and the following steps are necessary:

Step 1) Design an experiment compatible with the test bed system architecture. Fig. 3 shows the EV motion control system architecture (a) and pseudocode for an EV motion control algorithm (b).

Infrared distance sensors are attached to the front and rear EV bumpers to sense distance between the EV and barrier. A motor driver circuit is also attached to control radio-controlled car movement.

Step 2) Determine the designed model in Simulink. Fig. 4 shows the block diagram of the designed system.



(a) EV motion control system architecture

```
Infinite loop
    looking for control command from Simulink.
    which command?
        which direction?
            forward; control motor shield
            backward; control motor shield
        read front/rear sensor values
        is EV at the end of test bed?
            move EV; control motor shield
            stop EV;  control motor shield
End
```

(b) Pseudo code for EV motion control algorithm

Fig. 3. EV motion control system architecture.

Step 2) Determine the designed model in Simulink. Fig. 4 shows the block diagram of the designed system.

In this step, a 1-D lookup table, compare to constant, switch, and logic operator library blocks are used to model the system. Inputs of this model are front and rear bumper distance sensors signals, accelerator signal, key switch signal and direction signals. Alarm distance between bumper and barrier is set to 50 centimeters and can be changed by altering the value in the constant comparison library block.

Step 3) Validate the model by generating an input signal and observing the output signal behavior. The block diagram built in step 2 is combined with subsystem blocks. Fig. 5 shows the signal builder connected to the implemented EV motion control model.

In Fig. 6 (a), a ten second accelerator signal is created, increasing from 0 to 100 percent (blue line) and key switch signal turned on at two seconds (orange line).

Results show a PWM duty signal stable when the system started until two seconds later. PWM duty increased depending on accelerator signal level.

In Fig. 6 (b), front bumper distance sensor data (blue line) decreased to less than 50 centimeters at four to five seconds. PWM duty behavior decreased to 0 to stop EV motion, because front bumper and barrier were too close. Results indicate that the proposed model may be used to control EV movement.

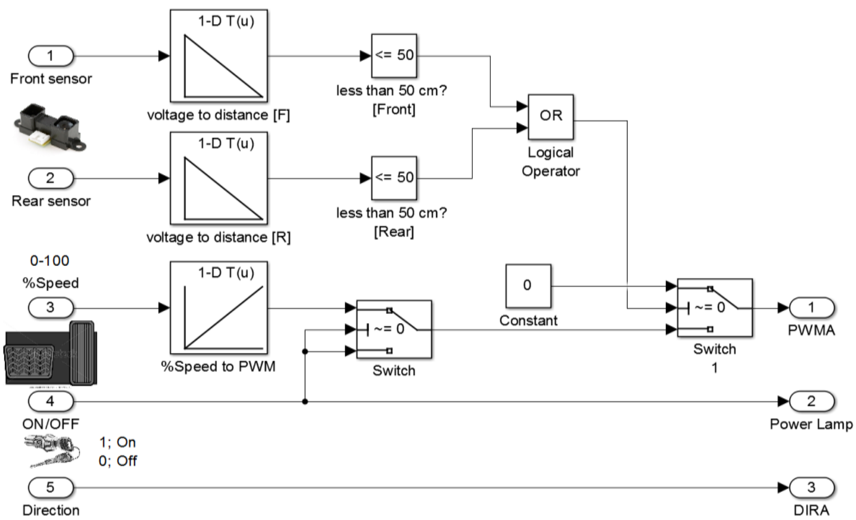


Fig. 4. EV motion control block diagram in Simulink.

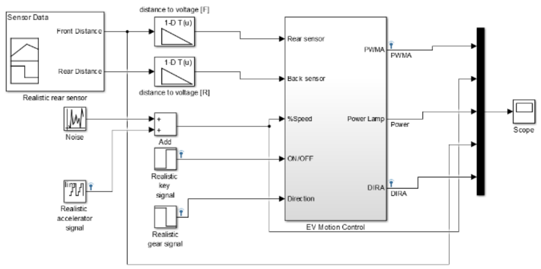
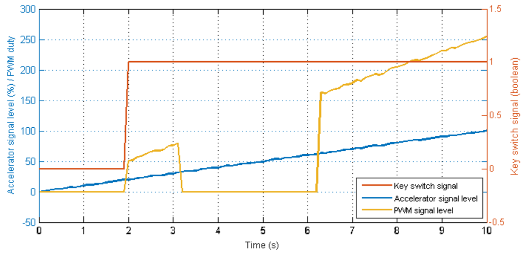
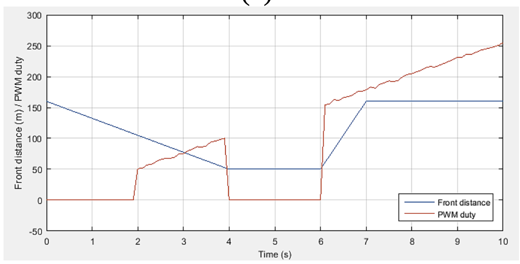


Fig. 5. Signal builder and implemented EV motion control model.



(a)



(b)

Fig. 6. Generated input and output signal behavior comparison of EV motion control model.

Step 4) Configure input and output behavior of Arduino peripherals deployed to Arduino, validating and tuning the parameters. In this step, analog input, constant, PWM, display and digital output libraries blocks provided by Simulink were added as shown in Fig. 7.

dashboard library blocks and evaluate the implemented model. Fig. 8 shows the implemented EV motion control in Simulink.

In this step, knob, toggle switch, and slider switch were added to work as an accelerator, key switch, and directional selector, respectively.

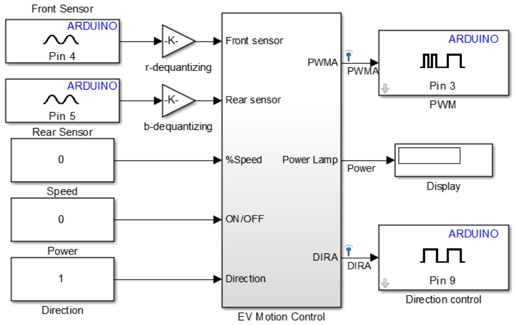


Fig. 7. EV motion control subsystem connected to Arduino peripherals library blocks.

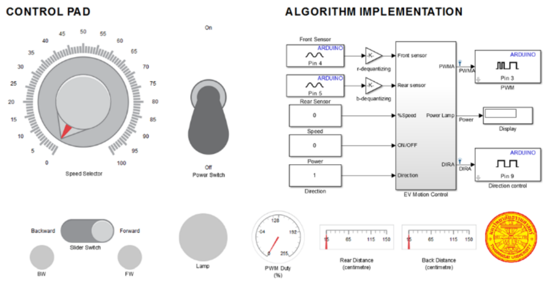


Fig. 8. Implemented EV Motion Control in Simulink.

Coil sequencing control

Coil sequencing control was necessary to prevent the wireless electric vehicle charger from heating metal objects near the transmitting coil and to enhance the overall efficiency while the EV is not in the charging range. To model the coil sequencing control using the procedure outlined in Section 3, these steps are followed:

Step 1) Design the experiment to be compatible with the test bed system architecture. Fig. 9 shows the coil sequencing control.

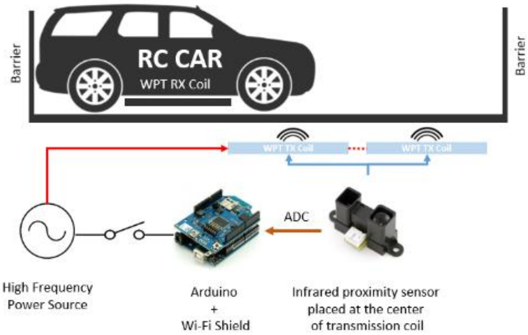


Fig. 9. Coil sequencing control system architecture.

Two infrared distance sensors were embedded in the transmitting coil center to sense distance between WPT TX and WPT RX coils. The Arduino was connected to the HF power source to control both WPT TX coils separately.

Step 2) The designed model was determined in Simulink. In this step, a 1-D lookup table and compare to constant library blocks are used to model the system. System input was the coil proximity sensor voltage and the output coil sequencing logic. Fig. 10 shows a block diagram of the designed system.

Step 3) The model was validated by generating a realistic input signal and observing the output signal behavior. The block diagram built in step 2 was combined with subsystem blocks. Fig. 11 shows the signal builder connected to the implemented coil sequencing control model.

In Fig. 12 (a), a ten second distance sensor signal was added with noise (red and blue lines). The signal was decreased and increased from two to four and five to eight seconds. Fig. 12 (b) shows that, when each distance signal level was lower than 30 cm, coil sequencing logic changed to activate each WPT TX coil.

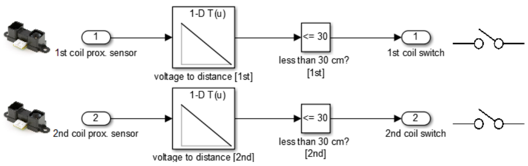


Fig. 10. Coil sequencing controller model in Simulink.

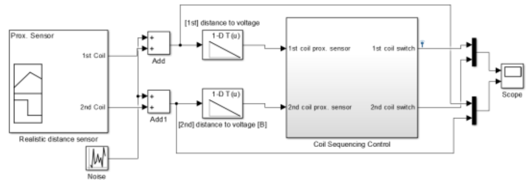
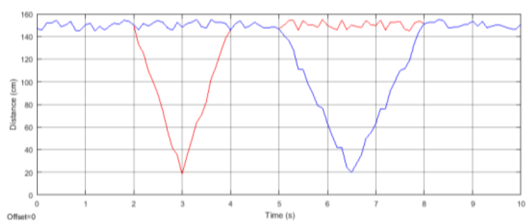


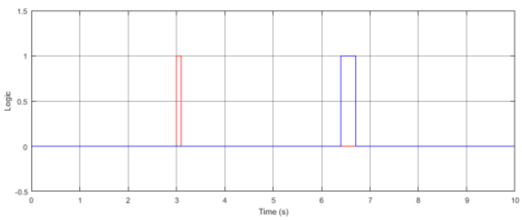
Fig. 11. Signal builder and implemented coil sequencing control model.

Step 4) Input and output behaviors of Arduino peripherals were configured by the Simulink hardware support package for Arduino library blocks, and deployed to Arduino for validating and tuning the parameters. Analog input and digital output library blocks provided by Simulink were added as shown in Fig. 13. This model was validated by varying the proximity sensor distance, then observing the behavior of the coil sequencing logic signal.

Step 5) The user interface created with dashboard library blocks and implemented model was evaluated. Fig. 14 shows the implemented model with user interface in Simulink.



(a)



(b)

Fig. 12. Generated input and output signal behavior comparison of coil sequencing control model.

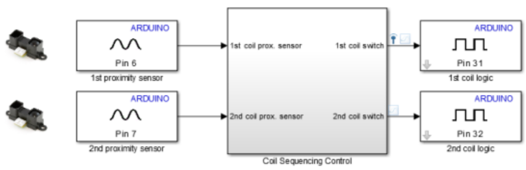


Fig. 13. Coil sequencing control model connected with Arduino peripherals library blocks.

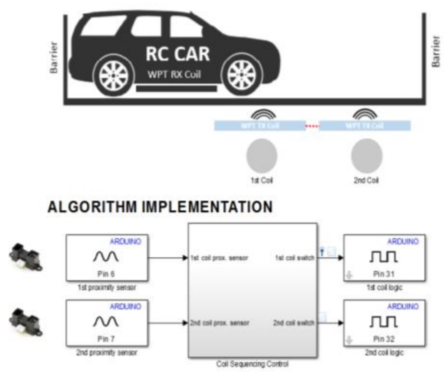


Fig. 14. Implemented coil sequencing control in Simulink.

In this step, lamp library blocks were added to display the coil sequencing logic status.

Real-time Power Efficiency Monitoring System

The proposed WPT efficiency measuring method in section 2.3 and the proposed example model in Sections 4.1 and 4.2 were integrated to operate in a power efficiency monitoring system. Voltage and current sensors were attached to an HF power source and transmitting coil on the primary side and receiving coil and load on the secondary side to sense the voltage and current as described.

To model a real-time power efficiency monitoring system using the procedure outlined in Section 3, the following steps are necessary:

Step 1) Design the experiment to operate with the test bed system architecture. Fig. 15 shows the real-time power efficiency monitoring system architecture.

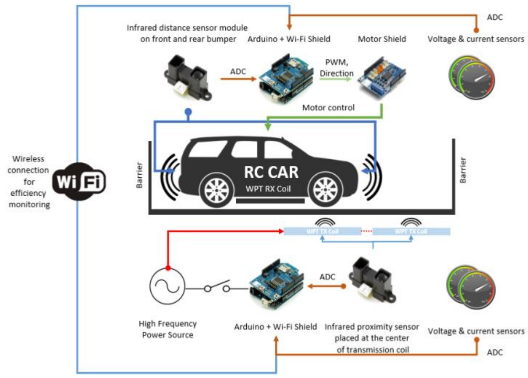


Fig. 15. Real-time power efficiency monitoring system architecture.

The Arduino and Wi-Fi shields on both sides were linked by a wireless network connection to work with the real-time executable function in Simulink.

Step 2) The designed model was determined in Simulink. The 1-D lookup table was used. System input was the voltage signal from the voltage and current sensors. Fig. 16 shows a block diagram of the power efficiency monitoring system in Simulink.

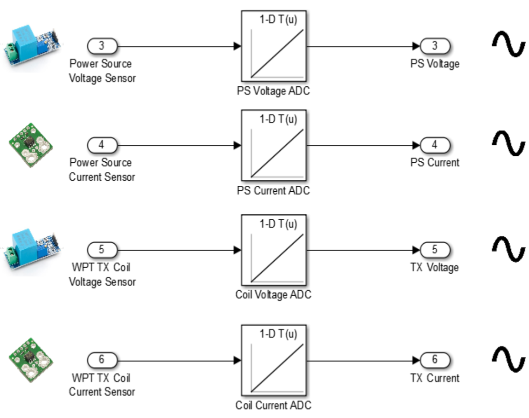


Fig. 16. Power efficiency monitoring model.

Step 3) Validate the model by generating an input signal and observing the output signal behavior. The block diagram built in step 2 was combined with subsystem blocks. Fig. 17 shows the signal builder connected to the power efficiency monitoring model.

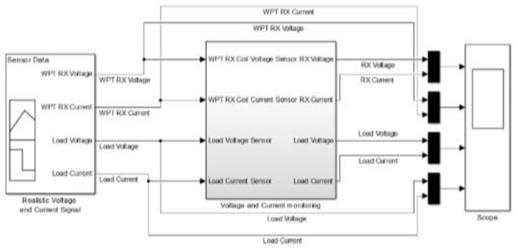
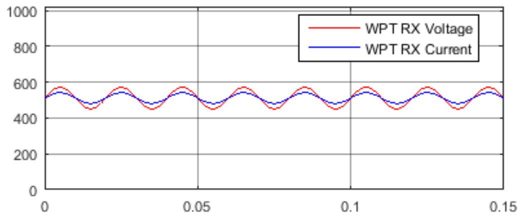


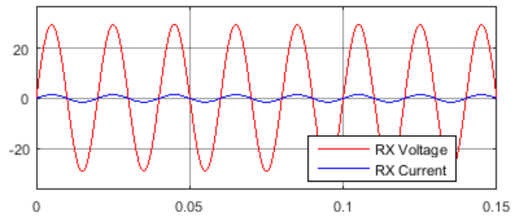
Fig. 17. Signal Builder and implemented Power efficiency monitoring model.

In Figs. 18-19 (a), voltage and current sensor signals were created for the WPT RX coil and load, respectively. The signal riding on DC was offset to imitate current and voltage sensor behavior. Figs. 18-19 (b) contain results after the generated signal passed through the implemented power efficiency monitoring model to read voltage and current levels.

Step 4) Configure input and output behaviors of Arduino peripherals, deploy to Arduino, validate and tune the parameters on both sides. Analog input and display libraries blocks provided by Simulink were added as shown in Fig. 20 with voltage and current sensors connected to Arduino to validate the model.

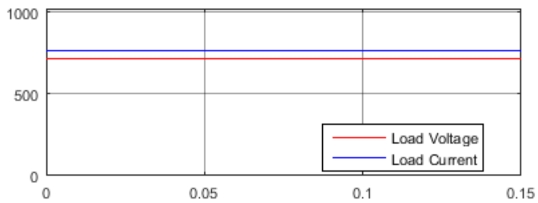


(a)

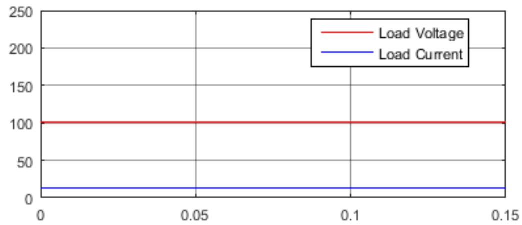


(b)

Fig. 18. Voltage and current sensor signals (a) compared to voltage and sensors signals after analog to digital conversion (b).



(a)



(b)

Fig. 19. Voltage and current sensor signals (a) compared to voltage and sensors signals.

behavior (b) after analog to digital conversion

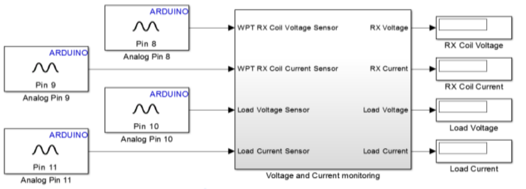
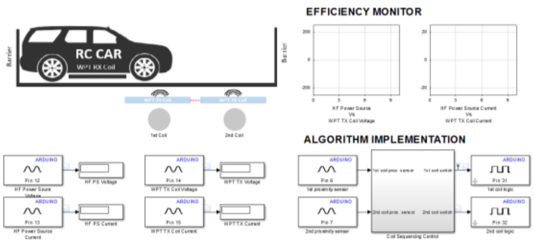
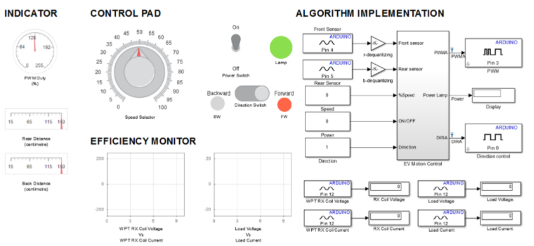


Fig. 20. Power efficiency monitoring model connected with Arduino peripherals library blocks.

Step 5) Create the user interface with dashboard library blocks and evaluate the implemented model. Fig 21. (a-b) shows the implemented real-time power efficiency monitoring system on the primary and secondary sides, respectively.



(a)



(b)

Fig. 21. Implemented real-time power efficiency monitoring system on primary side (a) and secondary side (b).

In this step, a dashboard scope was added to the primary and secondary sides for real-time monitoring of voltage and current.

Conclusion and Future Works

A new method has been outlined and illustrated for implementing, modeling, and measuring in-motion wireless power transfer for a wireless electric vehicle charger. The goal is to increase understanding of implementation, characteristics, operation, communication, and control of in-motion wireless power transfer systems by using MATLAB/Simulink and low cost hardware such as Arduino to implement the test bed system.

In this instance, authors can easily make a user-friendly interface test bed system. Interfacing with many sensors, such as voltage and current sensor or proximity sensor, is easily achievable enabling real-time parameter tuning of the control model. This approach may be applied to create a complex control system for in-motion wireless electric vehicle charger such as aswitching controller or power flow control system with feedback to improve the charging efficiency.

Acknowledgements

This work was supported by the Higher Education Research Promotion, National Research University Project of Thailand, Office of the Higher Education Commission, the Graduate Scholarship and Thesis Research Funding Project from Faculty of Engineering, Thammasat University.

References

[1] Miller JM, Onar OC, White C, Campbell S, Coomer C, Seiber L, Sepe R, Steyerl A. Demonstrating dynamic wireless charging of an electric vehicle: The benefit of electrochemical capacitor smoothing. IEEE Power Electronics Magazine;1:12-24

[2] Miller JM, Onar OC, Chinthavali M. Primary-Side power flow control of wireless power transfer for electric

vehicle charging. IEEE Journal of Emerging and Selected Topics in Power Electronics;3:147-62

[3] Li S, Mi CC. Wireless power transfer for electric vehicle applications. IEEE Journal of Emerging and Selected Topics in Power Electronics;3:4-17

[4] Chung YD, Lee CY, Kang H K, Park YG. Design consideration and efficiency comparison of wireless power transfer with HTS and cooled copper antennas for electric vehicle. IEEE Transactions on Applied Superconductivity;25:1-5

[5] Choi SY, Gu BW, Jeong SY, Rim CT. Advances in wireless power transfer systems for roadway-powered electric vehicles. IEEE Journal of Emerging and Selected Topics in Power Electronics;3:18-36

[6] Sun-han Hwang, Yong-ho Son, Byung-Jun Jang. Software-based wireless power transfer platform for power control experimentation. 2014 IEEE Wireless Power Transfer Conference (WPTC):80-83

[7] Nam Yoon Kim, Ki Young Kim, Young-Ho Ryu, Jinsung Choi, Dong-Zo Kim, Changwook Yoon, Yun-Kwon Park, Sangwook Kwon. Automated adaptive frequency tracking system for efficient mid-range wireless power transfer via magnetic resonance coupling. 2012 42nd European Microwave Conference:221-224

[8] Seung-Hwan Lee, Lorenz, R.D., Development and Validation of Model for 95%-Efficiency 220-W Wireless Power Transfer Over a 30-cm Air Gap. IEEE Industry Applications;47:2495-2504

[9] Simulation and Model-Based Design [Internet]. [Cited 2016 Feb 29]. Available from : http://www.mathworks.com/products/simulink/

[10] Arduino - ArduinoWiFiShield101 [Internet]. [cited 2016 Feb 05]. Available from: https://www.arduino.cc/en/Main/ArduinoWiFiShield101.