International Journal of Mechanical, Industrial and Aerospace Sciences
ISSN: 2517-9950
Vol:9, No:6, 2015

Implementation of Conceptual Real-Time Embedded Functional Design via Drive-by-Wire ECU Development

A. Ukaew, C. Chauypen

***Abstract*—**Design concepts of real-time embedded system can be realized initially by introducing novel design approaches. In this literature, model based design approach and in-the-loop testing were employed early in the conceptual and preliminary phase to formulate design requirements and perform quick real-time verification. The design and analysis methodology includes simulation analysis, model based testing, and in-the-loop testing. The design of conceptual drive-by-wire, or DBW, algorithm for electronic control unit, or ECU, was presented to demonstrate the conceptual design process, analysis, and functionality evaluation. The concepts of DBW ECU function can be implemented in the vehicle system to improve electric vehicle, or EV, conversion drivability. However, within a new development process, conceptual ECU functions and parameters are needed to be evaluated. As a result, the testing system was employed to support conceptual DBW ECU functions evaluation. For the current setup, the system components were consisted of actual DBW ECU hardware, electric vehicle models, and control area network or CAN protocol. The vehicle models and CAN bus interface were both implemented as real-time applications where ECU and CAN protocol functionality were verified according to the design requirements. The proposed system could potentially benefit in performing rapid real-time analysis of design parameters for conceptual system or software algorithm development.

***Keywords*—**Drive-by-wire ECU, in-the-loop testing, model-based design, real-time embedded system.

I. Introduction

An embedded system is a self-contain computer hardware and software module, which contains input, data processing and output components. It is designed to perform specific functions. Often, the embedded system is required to work in real-time manner where the executed time is specified. Real-time deadline can be fast or slow depend upon system functionality for a design requirement [12]-[16].

Once a product design is assigned, its design architecture is structurally planned to accommodate the design intent or customer need. Architecting involves selecting the process, identifying parameters, defining interface, performing feasibility analysis and formulate requirements [19], [20]. The design phases of real-time embedded system include conceptual design, preliminary design, critical design, production, and operation or maintenance phases [12], [18]. The well-known design architecture for a product contained a safety critical embedded system is called *v-model* process as presented in Fig. 1 [12]. This type of design architecture is normally founded in automotive control system development. In current literature, both the conceptual and preliminary design phases of this design process are emphasized.

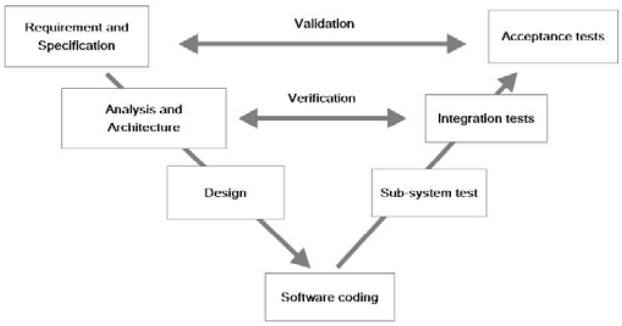

Fig. 1 The design process architecture known as *v-model* for a safety critical embedded system [14], [16]

In the conceptual design phase, feasibility is normally performed to explore, study, and analyze ideas and possible design. Within the process, the design requirements and objective are defined. Moreover, a development approach, schedule and design constraints are proposed [12]. Also, system architecture, including concepts, hardware components, and software components, are defined. On the other hand, the preliminary design could involve analysis, simulation, or early prototype testing [17]. During these design process, block diagram, schematic diagram, and configuration are also prepared. Furthermore, basic system requirement may be identified or estimated. Thus, analysis and design verification of requirement are mandated to be performed at this stage [13], [15].

II. Modified V-Model for Conceptual Design

In this literature, model based design approach and in-the-loop test were proposed as design architecture to accommodate *v-model* process in conceptual and primary design phases as shown in Fig. 2 for the safety critical system. The design of drive-by-wire ECU for electric vehicle conversion was chosen as a case study to demonstrate how proposed methodology can be employed. As a result, design requirement and objective were both defined in concepts and rapid real-time verification can be performed at early stage of the design process. Advantage and some trade-off are also

A. Ukaew is with Naresuan University, Phitsanulok, 65000 Thailand (phone: +66964231; fax: +66964004; e-mail: ananchaiu@nu.ac.th).
C. Chauypen is with Naresuan University, Phitsanulok, 65000 Thailand (e-mail: choopongc@gmail.com).

discussed for implementation of the proposed design architecture.

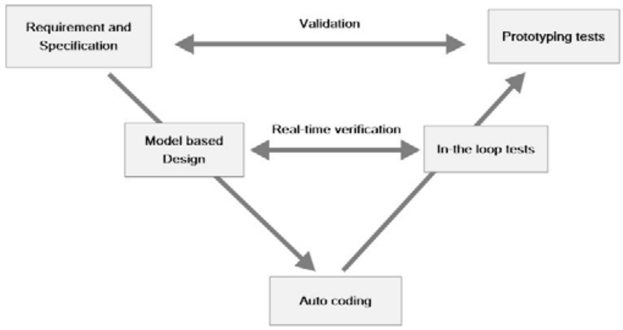

Fig. 2 *Modified v-model* design process architecture for conceptual and primary design phases

III. Design Background

Substitution of internal combustion engine parts with electric power train components (such as an electric motor, an inverter, and batteries) could alter the vehicle system characteristics significantly. This alteration could result in poor EV conversion performance or could cause safety and a reliability issue after the conversion is completed. As a result, drive-by-wire or DBW ECU is proposed to be implemented within electric vehicle or EV conversion system to synchronize EV propulsion dynamics and existing vehicle characteristics [8]. In addition, DBW functionality can be employed to improve EV drivability by providing power demand to the electric motor drive according to the driver preference. However, installation of the DBW ECU without appropriate evaluation could induce such system failures or component malfunctions due to unpredicted behaviors during actual driving situations. Therefore, during the initial development process, ECU functions are needed to be evaluated for the design requirements and safety beforehand [6], [10]. In addition, the real-time test system could be applied to predict electric vehicle conversion performance [2], [4].

In current DBW ECU development as shown in Fig. 3, DBW software functions were developed by mean of model-based design approach. Embedded ECU parameters and its CAN communication protocol were also determined. Then, the DBW functions were evaluated with simulation in virtual vehicle environment. For requirement-based test of the conceptual DBW ECU, real-time vehicle dynamics model was employed to evaluate the DBW ECU functionality and its network communication. In this case, conceptual ECU can be implemented and tested in the without the present of the actual EV prototype. The test system can also be implemented early where low fidelity requirement is required for initial evaluation of new ECU concepts. Furthermore, ECU parameters can be simultaneously adjusted by mean of the model-based development process.

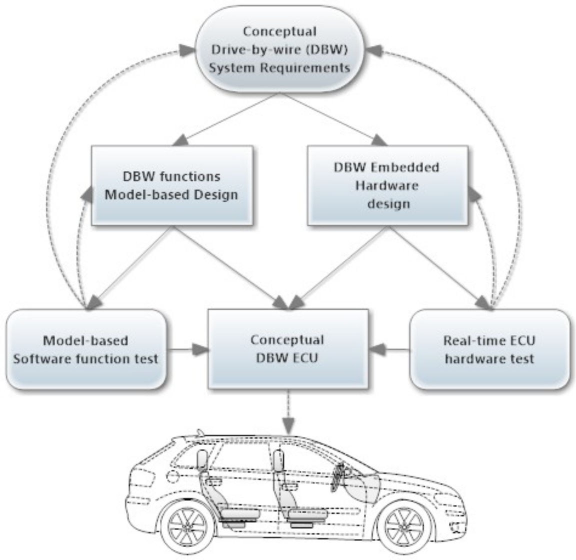

Fig. 3 Conceptual DBW ECU development for EV conversion

IV. Conceptual Drive-by-Wire Functions

The main function of conceptual drive-by-wire ECU developed by [8] is to determine power demand from the driver, through vehicle supervisory control ECU, based on the pedal ratio in percentage as shown in Fig. 4. As a result, torque setpoints or command for motor drive is calculated from the ratio power demand to motor speed. The command then is sent to the motor drive unit and updated on control area network or CAN bus with specified protocol configuration. The details of input and output commands for the ECU are presented in Fig. 5.

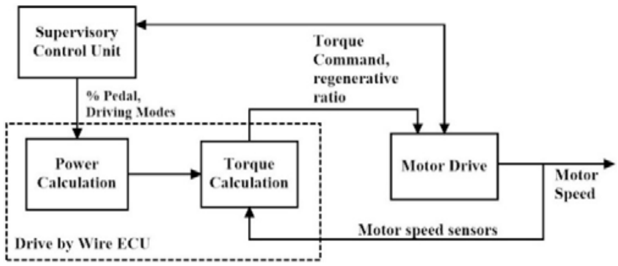

Fig. 4 Drive-by-wire ECU functions and signal connection to the supervisory ECU and the motor drive unit [8]

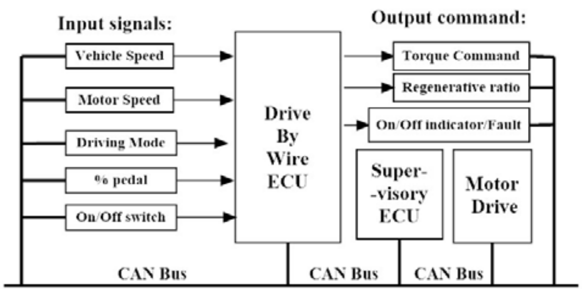

Fig. 5 Input and output signal flow of the drive-by-wire ECU with CAN bus interface [8]

Torque setpoints for all motor drive are determined from the power requested by the driver pressing on the pedal. Based on driver's preference or judgment, the vehicle could be handled in different road or traffic conditions. In order to meet driving demand shown in Fig. 6, adequate torque setpoints are required. Moreover, the driving modes introduced earlier need to be incorporated in the software function of the drive-by-wire ECU. In this initial development, linear conversion of power to torque command in Fig. 7 and driving schedule operation are defined and implemented in the software function. For simplicity purpose, power demand by the driver is linearly calibrated with accelerator pedal position ratio where full pressing on the pedal or one hundred percent pedal ratio is equivalent to the maximum demand for EV driving. This concept can be used as the basis for more complicated and optimized power demand algorithm, such as calibration with production pedal feel. Torque command (T) is calculated by dividing the power demand (P) with motor speed (ω) obtained from CAN bus. For vehicle reversing and decelerating operation, the toque command is defined as negative values.

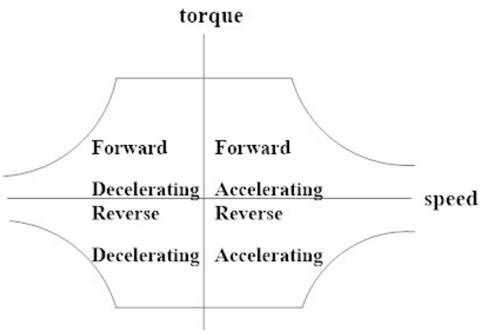

Fig. 6 Operation of electric propulsion for converted EV driving [7]

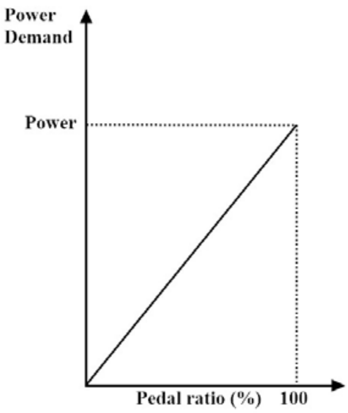

Fig. 7 The pedal ratio is linear proportioned to the power demand [8]

V. EV Conversion Modeling

Virtual EV conversion modeling employed in the current test was mainly based on [4] physical based models where all mathematical models for EV dynamic system were generated from engineering principles and theories [4]. The core models are consisted of electric vehicle traction, motor, battery, and EV power flow models. Some model parameters, such as motor efficiencies and vehicle moment of inertias, were reasonably estimated.

For EV traction modeling, forces acting on the vehicle governed the traction equation as seen in Fig. 8. Those forces comprised of tractive force (F_{te}), rolling resistance force (F_{rr}), aerodynamic force (F_{ad}), lateral acceleration force (F_{al}), wheel acceleration force (F_{wa}), and hill climbing force (F_{hc}) or the component force of vehicle weight.

The relation is shown in (1) where traction needs to overcome the load is equal to five other forces;

$$F_{te} = F_{rr} + F_{ad} + F_{hc} + F_{la} + F_{wa} \tag{1}$$

The electric motor is installed to replace the internal combustion engine or ICE in providing the torque to drive the wheels with the efficiency of approximately 90 percent. Traction model is divided into two phases due to the motor traction characteristics. The electric motor is operated at a maximum torque at low speed and then the torque value is declined with constant motor power. The point where the torque characteristic starts to change is known as motor critical speed.

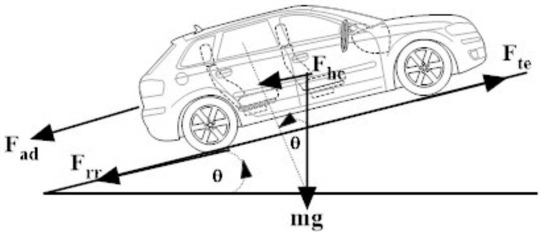

Fig. 8 Force components involved in the electric vehicle traction [4], [10]

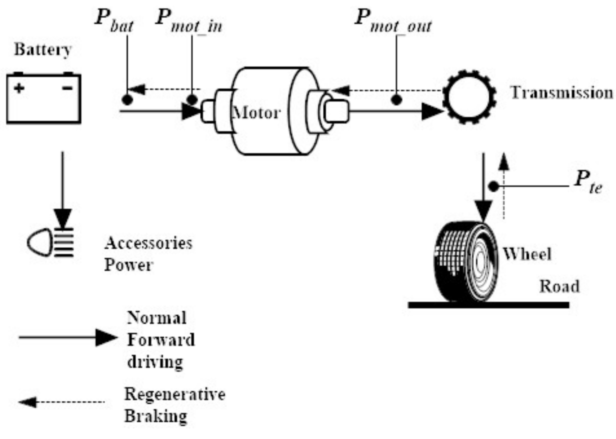

Fig. 9 The diagram shows power flow in/out components within EV system for both normal forward driving and regenerative operations [4]

Battery dynamic behavior can also affect EV performance and autonomy or range. Lithium ion battery type will be chosen as the power source for power source. Its open circuit voltage is known to be constant until 80 percent depth of discharge (*DOD*).

As shown in Fig. 9, the schematic diagram represents power flow model, which is transient forward facing type, to represent power flow within the EV system [4]. This particular modeling method is preferred for low fidelity model based simulation and testing since it can reduce complexity and errors within the test process.

VI. Model Based Software Function Test

Evaluation of drive-by-wire software requirement can be performed at the early design stage by mean of simulation method. Mathematical models of the ECU function model, a vehicle dynamic model, and related components were integrated as shown in Fig. 10. The detail of EV conversion components system modeling and drive-by-wire function models are explained in details by [8]. The integration of models allows the system to be simulated and evaluate the requirement early on [1], [3], [10]. Mathwork/Simulink was chosen to build the drive-by-wire function models due to its versatility in modeling under visual environment [1], [3]. Simulation analysis was performed to evaluate the drive-by-wire functions against the design requirement. The DBW design requirement in this case is to produce torque command for the EV motor drive. DBW functional analysis can be further realized by examination of EV response based on the driving test profile.

VII. Simulation Analysis of Concepts

After the models were constructed, simulation then could be performed. EV driving was simulated for all parking, driving, and reversing driving schedules as shown in Fig. 11. Torque commands generated from each driving schedule were observed. As a result, design requirement of the DBW function was met where torque commands could be produced within quadrants drive for electric propulsion.

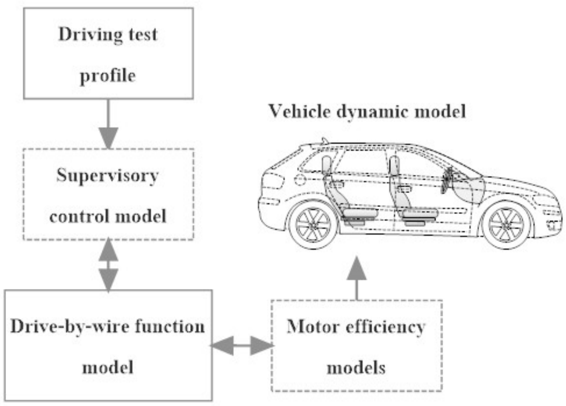

Fig. 10 Mathematical models integration concept of Drive-by-wire function models and simulated environment models

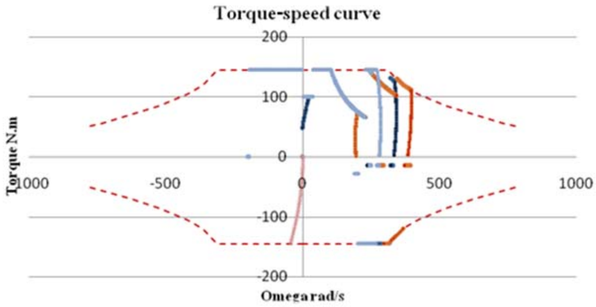

Fig. 11 Drive-by-wire function simulation to produce torque commands in quadrants drive as evaluation of design requirement

By inserting the driving test profile as the input pattern in the simulation, vehicle velocity transient, torque dynamics, and power demand from driver were obtained. As shown in Fig. 12. Vehicle velocity was seen gradually increase even in full press duration since converted EV carries such heavy battery packs and other components. Motor torque could be seen to follow the driving pattern as expected. In addition, power demand values shown in Fig. 12 were increased according to valid DBW functions design.

VIII. ECU Hardware Function Test

In the current development, the implemented hardware and software components were consisted of real DBW ECU hardware, virtual vehicle dynamics models, and vehicle network protocol or CAN protocol. The details of HIL system specifications and CAN hardware are listed in Table I.

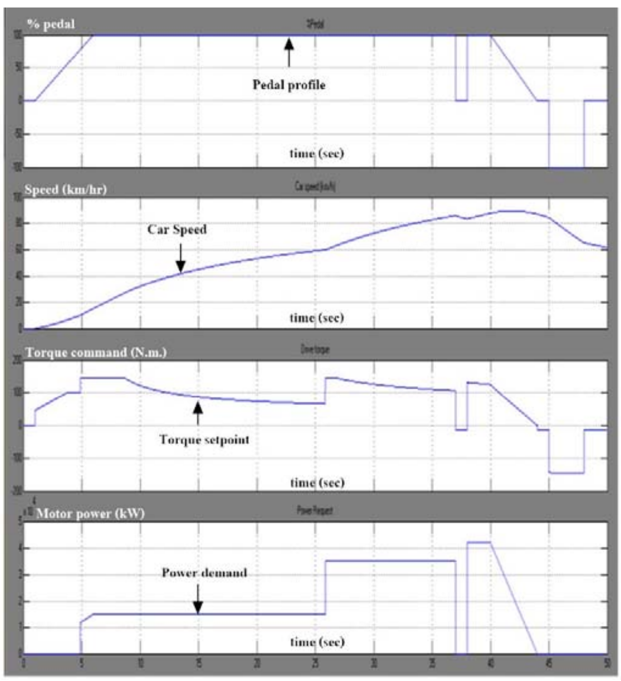

Fig. 12 Drive-by-wire functions simulation with driving test profile as the input and vehicle speed in kilometer per hours [km/hr], torque dynamics in newton-meter [N.m.], power demand in kilowatts [kW], and time in seconds [sec]

TABLE I
DETAILS OF TEST SYSTEM SPECIFICATION

Components	Specification
drive-by-wire ECU	real-time rapid prototyping board CPU : ARM Cortex-M4 32bits 168 MHz RAM : 8 Mb CPU : ARM Cortex-M4 32 bits 168 MHz RAM : 8 Mb
vehicle dynamic and driving profile real-time applications	real-time processor board CPU : ARM Cortex-M4 32bits 168 MHz RAM : 8 Mb CPU : ARM Cortex-M4 32bits 168 MHz RAM : 8 Mb
real-time platform	Mathwork/Simulink real-time workshop
interface	CAN bus 2.0 (high speed) baud rate: 500 kbaud
physical connection	CAN: DB9 connector
power supply	12V terminal
protocol sampling time	10 ms

MHz = megahertz, Mb = megabytes, kbaud = kilobaud V = volt, ms = millisecond

The current test system is affordable and capable of handling low fidelity virtual environmental models (vehicle dynamics, supervisory) as shown in Fig. 13. The test system is similar to Lab-Car test system proposed by [5] which is comprised with close loop testing capability. By applying rest-bust simulation, driving test profile model could be inserted in CAN bus communication. The actual hardware DBW ECU (the block with solid color) is interfaced to virtual environment models though CAN bus interface. Motor efficiency models and vehicle dynamic models receive torque command to determine torque value for EV driveline, and calculated vehicle velocity. The motor speed signal was fed back into the actual DBW ECU. All virtual environment models, including real-time blocks, were built using Mathwork/Simulink program.

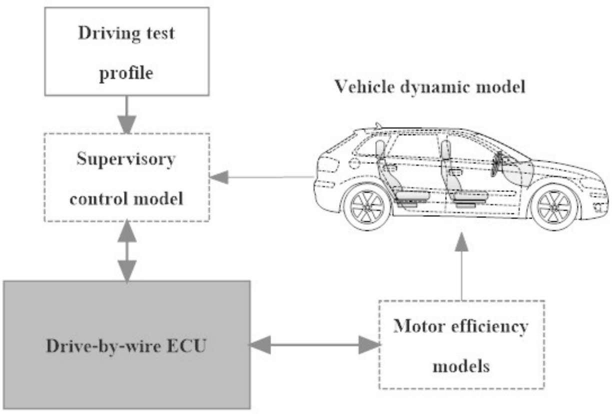

Fig. 13 The test system integration concept for drive-by-wire ECU requirement based and performance test

The actual components setup is shown in Fig. 14. The setup test for the DBW ECU consisted of software and hardware components, which were interfaced with PC contained Simulink real-time workshop module. The DBW ECU and real-time application hardware interface were connected by CAN bus interface cables and connectors.

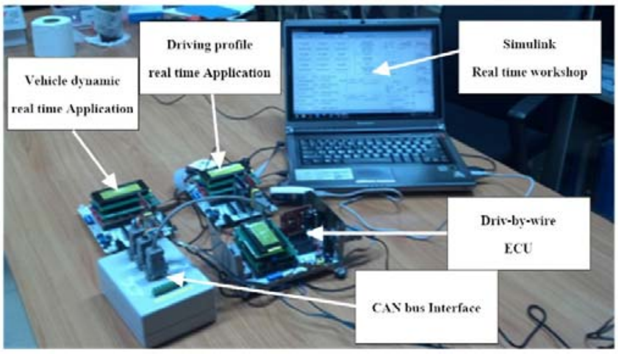

Fig. 14 Real components setup of close loop system consists of two real-time module cards, DBW ECU, and PC with Mathwork/Simulink real-time application

The real-time environment models consist of the vehicle dynamic model, and the driving profile model as shown in Fig. 15. Models were downloaded into each processor board as real-time applications. The actual DBW ECU was connected within the test loop. In real EV operation, fault tolerant protocol of DBW ECU was managed by the supervisory control ECU where supervisory ECU would monitor the lost DBW ECU signals. When that situation is occurred, supervisory ECU would activate EV limp-home (emergency safety driving) mode instantaneously.

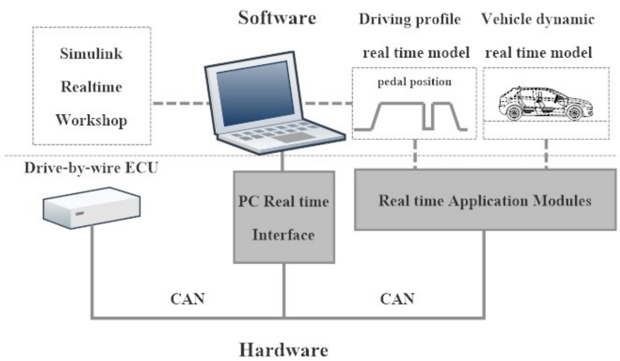

Fig. 15 Hardware component integration setup for ECU functional tests

IX. ECU Function Test Analysis

The EV models and the CAN bus protocol were both implemented as real-time application where ECU parameters and CAN messages could be monitored and evaluated according to requirements for drive-by-wire ECU. The ECU requirements are similar to those found in model-based software evaluation where the implemented ECU is required to produce torque command for the EV electric propulsion system. The DBW ECU performance was also evaluated by examining the vehicle velocity response to driver command as shown in Fig. 16.

The driving profile chosen consisted of periods of no press on the pedal, partial press on the pedal, full press on the pedal, hard press/depress the pedals and partial release on the pedal, and then complete release on the pedal. The total driving period was 60 seconds.

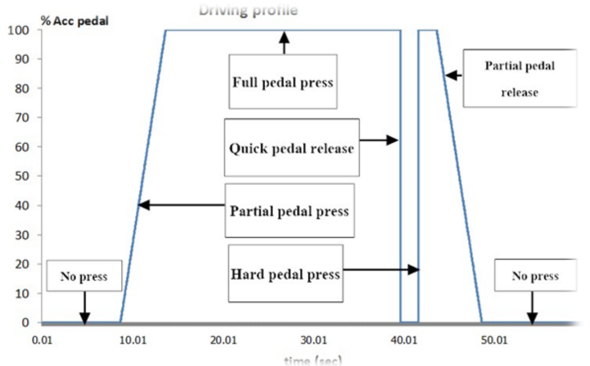

Fig. 16 Driving profile test profile based on accelerator pedal position

Based on the driving profile, the test results indicated that power demand (Fig. 17) rose and fluctuated during the hard press/depress period then declined as the pedal was depressed, consistent with the driving pattern. As for the torque-speed curve, torque command required for the motor drive was observed where torque setpoints produced by DBW ECU rose instantaneously within torque limit of 145 Nm. as shown in Fig. 18. The torque command values were also fluctuated as well during transient period, and then declined as the accelerator pedal was depressed. This torque characteristic ensured that DBW ECU can provide torque command for motor drive upon the driver request.

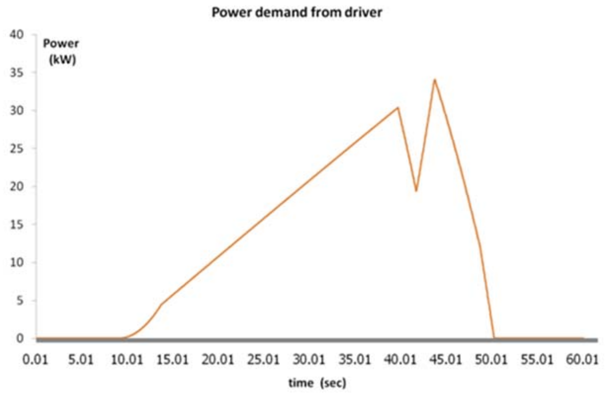

Fig. 17 Power demand from driver based on driving profile

In addition, EV performance could be also predicted with the current DBW ECU implementation. The vehicle velocity in Fig. 19 shows that converted EV could accelerate from 0 to 100 km/hr within 26.83 seconds. This sluggish acceleration behavior is caused by the high payload of EV conversion, approximately 1.7 tons in gross vehicle weight. Therefore, future development in reducing the electric vehicle conversion weight is considered to improve the performance. On the other hand, this hypothetical EV could produce maximum speed as much as 147 km/hr for this particular driving pattern.

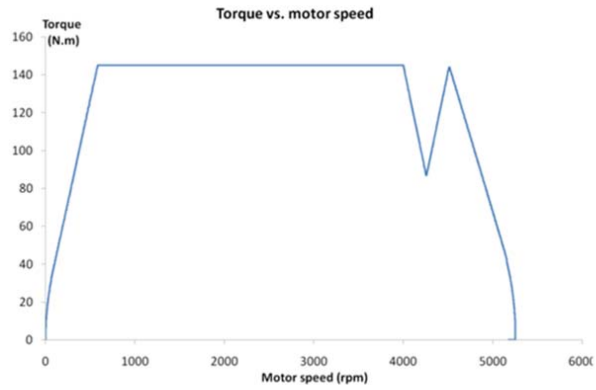

Fig. 18 Torque-motor speed curve obtained from electric propulsion

The proposed system was implemented to validate the functional concepts of DBW algorithm embedded in the ECU unit where low fidelity environmental models were needed. It is purposely to avoid the risk of model inconsistency problem. Nevertheless, it can be performed as a low cost and flexible real-time validation manner.

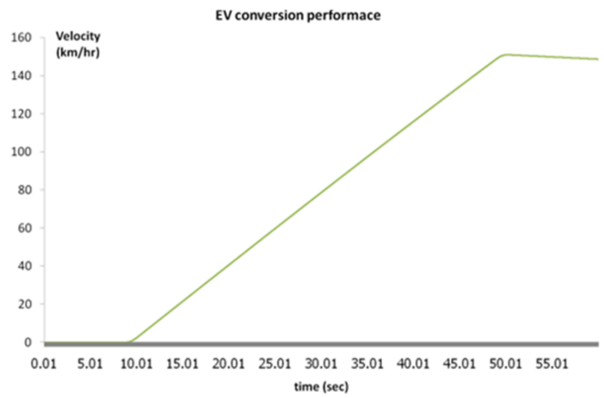

Fig. 19 EV velocity based on driving profile

Thus, the design architecture enabled conceptual development to become more affordable for initial design phase. The design feasibility and analysis could be performed early to verify whether basic requirements are met. Also, design decision can be made to adjust or terminate at this point by mean of evaluation or analysis. Nevertheless, when critical design phase or more complex ECUs are expected [22], [24], design verification must be conducted with the full test system of hardware in-the-loop (HIL) configuration [23]. Implementation of HIL system would require high performance computing capability and various hardware inputs and outputs or I/O ports [6], [10]. The standard testing methodology is also needed for the real ECU production and the actual prototype is also required to validate the ECU functionality and reliability [9], [11]. Design dependability or fail-safe test for safety critical system, such as fault tolerance or redundant test, are suited with HIL configuration or laboratory bench test [21], [24]. The test facility would require higher cost to build depending on functional complexity of system and functionality.

X. Conclusions

The design process of conceptual real-time embedded system can be improved by implementing model based design and in-the-loop test approaches. The design requirement can be verified early and quick real-time simulation and analysis can be performed. Design feasibility and evaluation may be employed at this early stage. The conceptual design of drive-by-wire ECU for electric vehicle conversion demonstrated the advantage for initial proofing of ECU concepts. Model based design approach, in-the-loop testing, and real-time testing were employed to ensure that the conceptual DBW functions and algorithm met the initial design requirement. The proposed practice and analysis illustrate potential benefits in conceptual design and evaluation of system functionalities.

Acknowledgment

The author would like to acknowledge the support of research under corporative electric vehicle research and development program from; Faculty of Engineering, Naresuan University, Cluster and Program Management Office (CPMO), National Science and Technology Development Agency (NSTDA), and Electric Generation Authority of Thailand (EGAT).

References

[1] Y. Choi, Y. Kim, H. Moon, and Y. Son, "Model based design and real-time simulation of the electric bike using RT-LAB and Simulink," *SAE Technical Paper* 2013-01-0110, 2013, *doi*:10.4271/2013-01-0110.

[2] A. Himmler, K. Lamberg, and M. Beine, "Hardware-in-the-loop Testing in the context of ISO 26262," *SAE Technical Paper* 2012-01-0035, 2012, *doi*: 10.4271/2012-01-0035.

[3] J. King, and D. Nelson, "Model-based design of a plug-in hybrid electric vehicle control strategy," *SAE Technical Paper* 2013-01-1753, 2013, *doi*: 10.4271/2013-01-1753.

[4] J. Larminie, and J. Lowry, *Electric Vehicle Technology Explained*. John Wiley & Sons, Ltd, UK, ISBN 0-470-85163-5, 2003, ch. 7.

[5] L. Michaels, S. Pagerit, A. Rousseau, P. Sharer, "Model-based systems engineering and control system development via virtual hardware-in-the-loop simulation," *SAE Technical Paper* 2010-01-2325, 2010, *doi*: 10.4271/2010-01-2325.

[6] S. Mueller, and P. Langjahr, "Efficient testing and cost awareness: low-cost versus HIL-system," *SAE Technical Paper* 2012-01-0933, 2012, *doi*: 10.4271/2012-01-0933.

[7] K. Patil, M. Muli, and Z. Zhu, "Model-based development and production implementation of motor drive controller for hybrid electric vehicle," *SAE Technical Paper* 2013-01-0158, 2013, *doi*: 10.4271/2013-01-0158.

[8] A. Ukaew, "Model based system design of conceptual drive-by-wire ECU functions for electric vehicle conversion," *SAE Int. J. Passeng. Cars – Electron. Electr. Syst.* 6(2):411-418, 2013, *doi*: 10.4271/2013-01-0426.

[9] G. Vandi, D. Moro, F. Ponti, R. Parenti, "Vehicle dynamics modeling for real-time simulation," *SAE Technical Paper* 2013-24-0144, 2013, *doi*: 10.4271/2013-24-0144.

[10] P. Vuli, M. Badalament, and V. Jaikamal, "Maximizing test asset re-use across MiL, SiL, and HiL development platforms," *SAE Technical Paper* 2010-01-0660, 2010, *doi*: 10.4271/2010-01-0660.

[11] M. Yeaton, "Managing the challenges of automotive embedded software development using model-based methods for design and specification," *SAE Technical Paper* 2004-01-0720, 2004, *doi*: 10.4271/2004-01-0720.

[12] K. R. Fowler, *What Every Engineer Should Know about Developing Real-Time Embedded Product*. CRC Press, Boca Raton FL 2008, ch. 1-3.

[13] I. Lee, J. Y-T. Leung, S. H. Son, *Handbook of Real-Time and Embedded Systems*. Chapman & Hall/CRC Boca Raton FL 2008, pp.2.1-2.16.

[14] R. William. *Real-Time Systems Development*. Butterworth-Heinemann, Burlington MA 2006, ch. 1-3.

[15] A. S. Vincentelli, H. Zeng, M. D. Natale, P. Marwedel, *Embedded System Development from Functional Models to Implementations*. Springer-Science, NY 2013, pp. 8-16.

[16] P. A. Laplante, *Real-Time System Design and Analysis*. 3rd Ed., IEEE Press, Wiley-Interscience, NJ 2004, ch 1,5.

[17] K. Perko, R. Kocik, R. Hamouche, A. Trost, "A modelling-based methodology for evaluating the performance of a real-time embedded control system," *J. Simulation Modelling Practice and Theory*, vol. 19-7, pp. 1594-1612, Aug. 2011.

[18] M. Törngren, O. Redell, "A modelling framework to support the design and analysis of distributed real-time control systems," *J. Microprocessors and Microsystems*, vol. 24-2, pp. 81-93, April 2000.

[19] M. Marinoni, T. Facchinetti, G. Buttazzo, G. Franchino, "An embedded real-time system for Autonomous Flight control," *Proc. ANIPLA, Int. Co. Meth. Emerging Tech. Automation*, 2006.

[20] U. Ali, M. B. Malik, "Hardware/software co-design of a real-time kernel based tracking system," *J. Sys. Architecture*, vol. 56-8, pp. 317-326, Aug. 2010.

[21] F. Afonso, C. Silva, A.Tavares, S. Montenegro, "Application-level fault tolerance in real-time embedded systems," *J. Industrial Embedded Sys.*, pp. 126-133, doi: 10.1109/SIES.2008.4577690, June 2008.

[22] D.C. Schmidt, "Middleware techniques and optimizations for real-time, embedded systems," *Proc. 12th Int. Sym. Sys. Synthesis*, pp. 12-16, doi: 10.1109/ISSS.1999.814254, Nov. 1999.

[23] M. S. A. Trab, M. Brockway, S. Counsell, R. M. Hierons, "Testing Real-Time Embedded Systems using Timed Automata based approaches," *J. Sys. Software*, vol. 86-5, pp. 1209-1223, May 2013.

[24] O. Redell, J. El-khoury, M. Törngren, "The AIDA toolset for design and implementation analysis of distributed real-time control systems," *J. Microprocessors and Microsystems*, vol. 28-4, pp. 163-182, May 2004.