Embedded ECU Emulation Platform for Safe Adaptive Front-Light System Control Algorithm Development

Jiae Youn , Men Di Yin , Jeonghun Cho , and Daejin Park

School of Electronics Engineering, Kyungpook National University, 80 Daehak-ro, Bukgu, Daegu, Republic of Korea

Abstract. Safety-conscious design and implementation of automotive electronics is now being important in safe driving under severe electromagnetic disturbance. In this paper, we introduce our experience and development framework to develop a safe electronic control unit (ECU) automotive controller algorithm using hardware-in-the-loop (HILS) based on Matlab/Simulink models, which are connected to ECU hardware. We propose a design flow of Matlab/Simulink based on model-based algorithm implementation of ECU controller and chip level ECU function emulation methodology using dSPACE's MicroAutoBox^{TM}, which is an off the shelf embedded system for ECU software-hardware emulation. As a case study, we shows that fast design and implementation of the specific ECU hardware, which is integrated with custom-designed its controller algorithm for adaptive front-light system (AFLS) is easily evaluated by using the proposed development framework.

1 INTRODUCTION AND MOTIVATION

Nowadays, recent advance of design automation methodology and infrastructure for the rapid development of electronic devices has led to considerable growing number of electronic control units (ECUs) based on complex on-chip hardware-software cooperation, which are integrated in the automobiles as main controllers.

There are more than 100 ECUs in automotive, including more than 3 million software lines are embedded to control electronic systems. Considering the complexity of development and testing overhead to evaluate the complicated operations ECU, we adopted the model based development (MBD) [1] by which the hardware-software is partitioned into Matlab/Simulink-based soft building blocks and on-board emulation based hardware blocks.

This method can overcome the shortcomings such as difficult to modify the hardware blocks and to verify the designed functions by using traditional development approaches through manual programing in hardware level.

Even more important, MBD is utilized as a hardware-independent method, which can be done even if hardware does not exist. We adopted design methodology in the integration of soft-driven model verification, code generation and rapid prototyping validation. Our recent efforts show that the proposed methodology enables successful design and implementation the custom-designed AFLS controller ahead of schedule as shown in Figure 1. which shows the overall of ECU development simulation.

2 PROPOSED METHODOLOGY

The AFLS (Adaptive Front Light system) has been used as one of the driver assistance system for safety driving [2]. AFLS provides visibility during nighttime driving, by lightening hidden area such as blind spot in corner.

Original AFLS function controls the angle of lights and turns off the adaptive control mode to prevent mal-functions under receiving the disturbance such as external noise injection on ECU. To improve the side-effects in this exception case, ECU developers will try to modify hardware data-path and its software algorithm to control AFLS more precisely in emergency status. We focus to reduce these iterative design efforts to modify and improve the pre-designed building blocks including hardware and software.

As a case study, we show the development experience of ECU hardware modification and on-board or on-chip software emulation by using the proposed ECU emulation framework. The successfully designed ECU, which continuously control headlamp by LUT (Lookup Table) of the backup AFLS presented in previous work [3], could be easily evaluated by the proposed ECU emulation platform based on the cooperation of software-driven Matlab/Simulink models and hardware-level target board. The backup AFLS is monitoring AFLS status real-time. If main AFLS enters into an abnormal condition, the headlamp is automatically intervened by the backup AFLS.

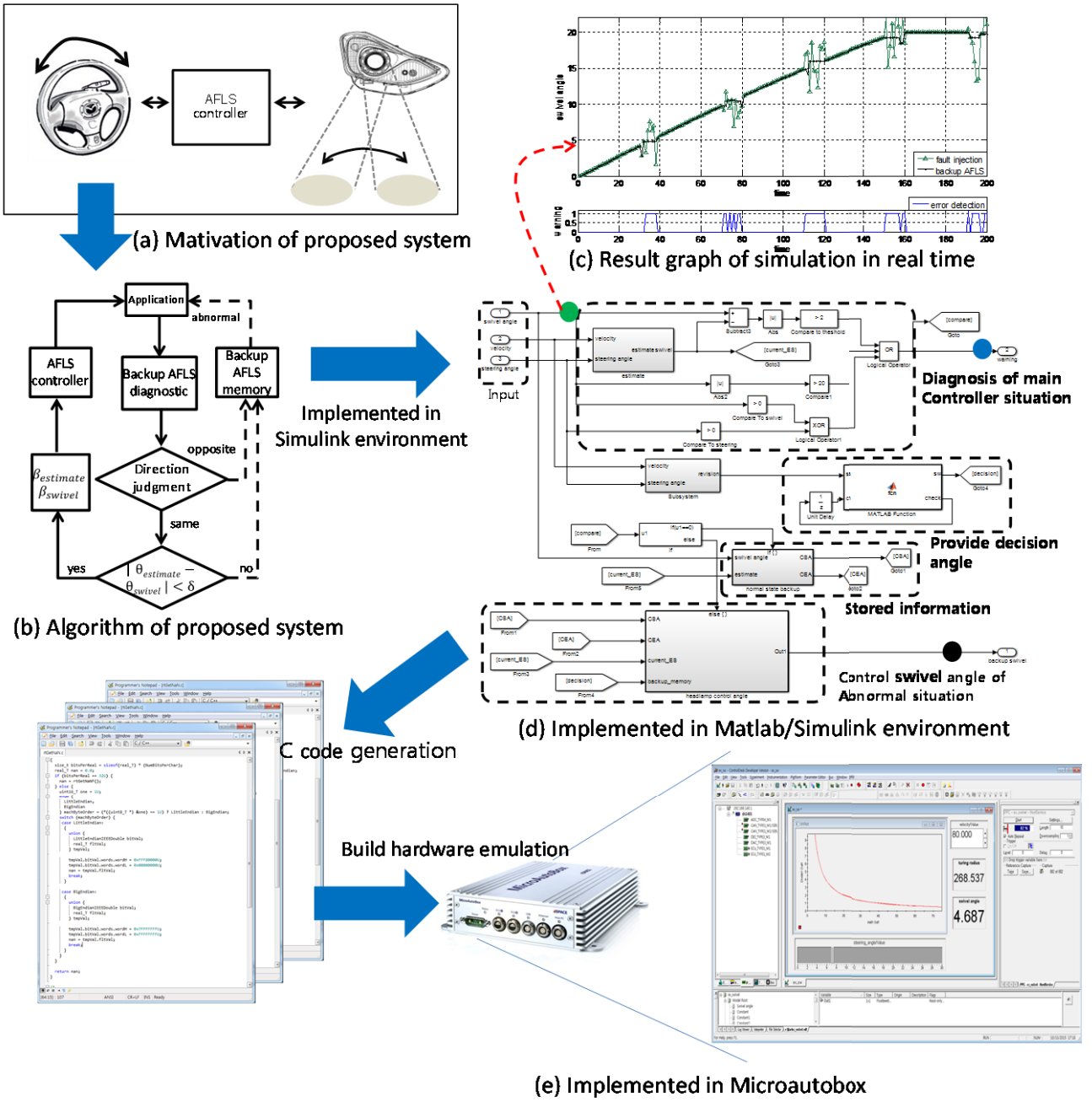

Figure 1. Overall of ECU development

Figure 1. show the overall of ECU development simulation. Figure 1. (a) show the motivation of proposed method. Figure 1. (b) shows propose backup AFLS algorithm. Then we implemented in Matlab/Simulink environment which is shown in model based design of Figure 1. (c). The requirement functions of AFLS can be easily modeled and evaluated by the methodology of model-in-the-loops (MIL) test.

After MIL test, we can easily hardware-in-the-loops (HIL) simulation and test which shows Figure 1. (e). In order to perform HIL test, we must generate target-specific firmware using Simulink configuration, which can be downloaded and executed in runtime on ECU chip. After the code generation is successfully build, we can HIL test using MicroAutoBox.

2.1 Model-In-the-Loop Simulation

The user defined AFLS control systems and its algorithm are developed using the proposed emulation method, which is based on Simulink-based model. An ECU designer in automotive company can easily modify the internals of AFLS systems by implementing user defined AFLS in soft-driven model level and compare result of simulation on the hard-driven target board level.

When AFLS is under fault situation, conventional AFLS control algorithm turns off the system modes. As a case study, we assume that an ECU designer tries to modify the functionality of the target-specific backup AFLS to monitor the status of the main AFLS controller and improve the safety of AFLs control operations. Newly designed AFLS in Figure 1. (d) follows the

proposed model development in which the external environment such as steering angle, vehicle speed and swivel angle of main controller is virtually simulated required in order to diagnosis the main AFLS controller. This study assumes the maximum steering angle and swivel angle of -35° ~ 35° and -20° ~ 20 as a case study.

When main AFLS controller is abnormal situation, Backup AFLS can continuously control headlamp using LUT corresponding to steering angle range and information of normal situation in main controller. If main controller is normal situation, main controller status is saved in buffer. When main controller is abnormal situation, we use backup memory within steering angle range. Headlamp is statically controlled by LUT corresponding to steering angle range. So we use both LUT and backup information to control headlamp.

To validate the performance of user defined backup AFLS, abnormal error conditions is virtually injected into main AFLS controller, which is implemented by Matlab/Simulink model. The envrionmental disturbance such as noise can be easily described by using the components in Matlab/Simulink model and retargeted into the ECU emulation board.

Figure 1. (c) show the runtime waveform in the designed AFLS model in case which the wave signals

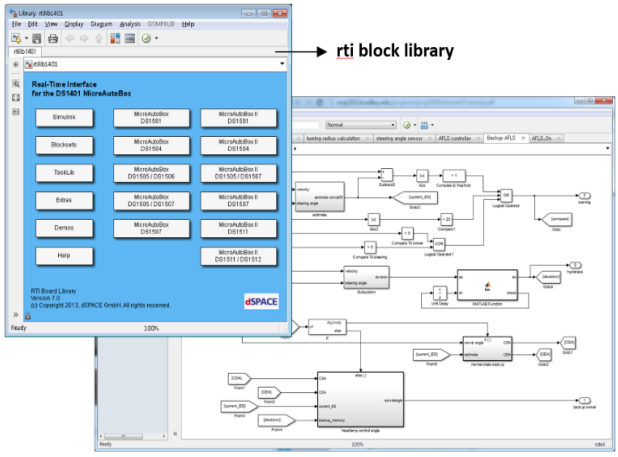

Figure 2. Using rti block labrary

shows swivel angle according to time. The green line shows the result of an abonormal AFLS controller. When error is detected by backup AFLS, the headlamp is controlled by backup memory. The black line shows the waveform of swivel angle controlled by backup memory. The second graph shows the main controller status. Result of simulation is evaluated by using scope block or plot compile in real-time.

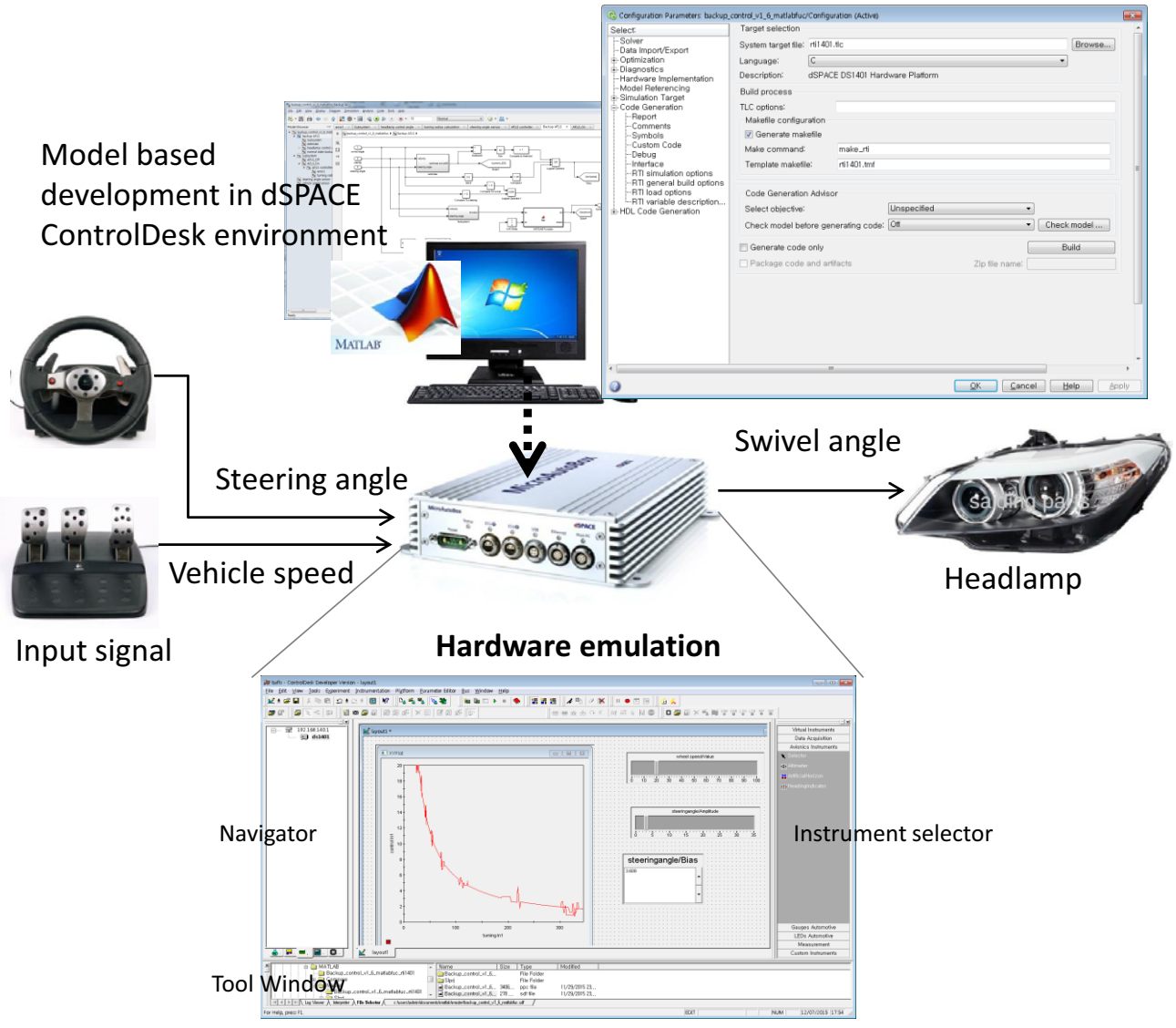

Figure 3. Hardware in the loop environment

2.2 Hardware-In-the-Loop Simulation

In our case study, the hardware-in-the-loop simulation (HIL) is implemented by using dSPACE MicroAutoBox™ environment [4-5]. The controller and its algorithm in ECU chip has been designed and simulated using Matlab/Simulink block. As shown in Figure 2. In order to control input/output signal in MicroAutoBox, user must use rti block library.

Figure 3. shows HILS environment used in modeling and simulating custom-designed AFLS. Simulink-based AFLS controller can be build in MicroAutoBox [6]. As shown in head of Figure 3, Simulink-driven model implementation is successfully compiled into MicroAutoBox. After building simulink models, simulator-specific configuration files, such as *ppc* file and *sdf* file, are generated in ControlDesk and downloaded into the target hardware, which is configured with user-defined parameters.

ControlDesk software enables to control the MicroAutoBox from PC in real time. By using the ControlDesk, we could easily control parameters of user-designed model in simulink.

We can check that output of the model simulation according to input. In first step we create layout file using instrument selector. When layout is generated in ControlDesk, designer needs to located the instrumented variables in the variable browser and then probes by using the instruments block [7]. We use layout file to check result of algorithm testing, which is shown in bottom of Figure 3. We check the swivel angle using *plotterarray* block which shows swivel angle corresponding to the turning radius used in proposed algorithm. Then, we check the result of swivel angle corresponding to velocity which is changed using instrument block. So we can easily control parameters and test algorithm.

MicroAutoBox is used as a real-time system for prototyping user-designed hardware and its algorithms test such as chassis control, body control and advanced driver assistance system [8]. MicroAutoBox is offer to adapt the I/O and signal condition using ControlDesk. The previously verified design can be integrated in MicroAutoBox, and parameters various is easily changed in real-time. So user can easily hardware in the loop simulation. We can check output signal according to input using ControlDesk. Which has graphics user interface, and then we can check in real-signal using MicroAutoBox Zero Insertion Force (ZIF) connector.

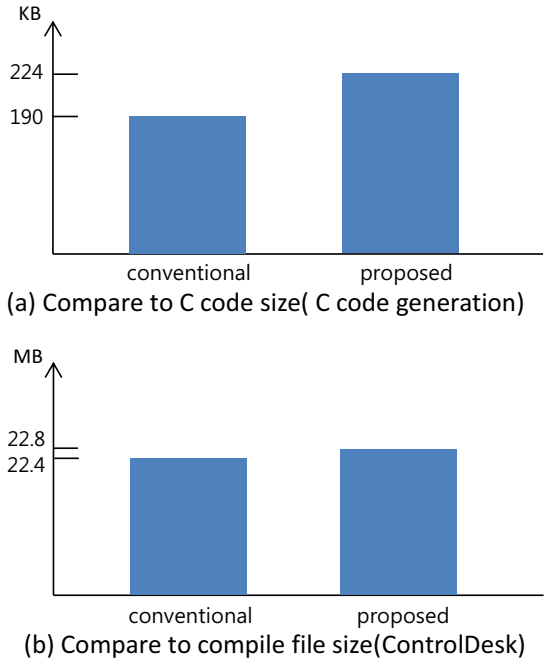

Figure 4. Hardware overhead comparison

2.3 Design case of Backup AFLS

AFLS is one of advanced driver assistance system (ADAS). AFLS improves visibility of driver during night-time driving. However, when conventional AFLS is abnormal situation, AFLS turns off the control mode and just performs fundamental lighting without movement of the lamp [9].

Backup AFLS monitor main AFLS status real-time and receives information such as velocity, steering angle, and current swivel angle according to main controller by BCM and main AFLS. Backup AFLS determines the error status saves the main AFLS information of normal status into backup memory, such as swivel angle corresponding to steering angle range.

If backup AFLS detect abnormal situation, backup AFLS read backup memory and LUT in order to continuously control the headlamp. If it is normal situation, saves the current swivel angle and information. If main AFLS enters abnormal state, backup AFLS notifies that the main controller is under abnormal situation. The backup AFLS monitor AFLS status in real-time and automatically intervened the main AFLS to control the system operation in emergency situation.

So we had to design a newly-designed backup AFLS based on fail-safety, which requires design efforts and implementation costs. We chose the HILS methodology to reduce the complexity and design time cost.

Our initial results in designing backup AFLS have been presented [3]. By injecting abnormal simulation through the Matlab/Simulink models, diagnosis functions of main AFLS fault situation have been successfully evaluated. In addition, the HILS evaluation, which is connected through the hardware-link between the PC0side Matlab/Simulink models and ECU-level emulation, results in more confidence for functions of the designed building blocks.

By using this methodology, Figure 4. show result of code size. Compile code size of proposed system is about 18 percent increase from conventional AFLS. However, hardware code size is just about 2 percent increase from conventional system.

3 CONCLUSION

This paper introduces our experience in terms of development framework to develop user-designed ECUs including hardware data-path and its software algorithm. We verified the implemented algorithm based on MIL test and HIL test by using Matlab/Simulink and ControlDesk toolbox in MicroAutoBox. We adopted Matlab/Simulink-based HILS methodology to verify and test in design step,

which leads to the reduced time and cost in modeling, simulation and validating the user-defined AFLS hardware and software. As a case study, we evaluated using the proposed methodology for model design procedures to effectively compensate for the control error that may occur in AFLS ECU. We successfully could compare check result of our soft-driven implementation with real ECU hardware.

References

1. J. Ghidella and J. Friedman, "Model-based design streamlines development of body electronics systems," 2005
2. D. Kim, "Study of optimized tuning in full afls head lamps," in proceedings of the FISITA 2012 World Automotive Congress. Springer, 1719 (2013)
3. J. A. Youn, M. D. Yin, J. Cho, D. Park, *GCCE* (2015)
4. Wang, Lingchang, "Development of a Hardware-In-the-Loop Simulator for Battery Management Systems," The Ohio State University, 2014
5. Gomez, Martin, "Hardware-In-the-loop simulation," Embedded Systems Programming **14**, 38 (2001)
6. M. H. Horter, C. Stiller, *Intelligent Vehicle Symposium*, 299 (2009)
7. N. Quijano, K. Passino, *A tutorial introduction to control systems development and implementation with dSPACE* (2002)
8. "http://www.dspace.com/en/pub/home/products/hw/micautob.cfm.
9. R. Ma, *Automotive adaptive front-lighting system reference design, Texas Instruments* (2013)