Control Engineering Practice 59 (2017) 137–148

Contents lists available at ScienceDirect

Control Engineering Practice

journal homepage: www.elsevier.com/locate/conengprac

A framework for simulation-based engine-control unit inspection in manufacturing phase

Won K. Ham[a], Minsuk Ko[b], Sang C. Park[a,*]

[a] Department of Industrial Engineering, Ajou University San 5, Woncheon-dong, Yeongtong-gu, Suwon 443-749, Republic ofKorea
[b] Manufacturing Technology, Hyundai Kefico Dang-dong, Gunpo, Republic ofKorea

ARTICLE INFO

Keywords:
Electronic engine-control unit
Inspection
Manufacturing phase
Reverse engineering
Simulation

ABSTRACT

This paper proposes a framework for electronic engine-control unit (ECU) inspection in manufacturing phase. Although various methodologies have been developed for the ECU verification at the development phase, the ECU verification in the manufacturing phase has rarely been brought into focus. ECUs in the manufacturing phase need the verification process in the unified way of the ECU software and hardware components by three major causes: 1) ECU software revision, 2) incomplete installation of software, and 3) quality variation of hardware. For the effective ECU verification in the manufacturing phase, we propose a simulation-based ECU testing methodology. The proposed framework configures virtual vehicle environment to simulate an ECU using a “virtual engine system (VES) model” that specifies the operations of every ECU function during a simulation. The proposed framework employs a reverse engineering approach to identify the operation state transition of an ECU function by reference vehicle states from vehicle experimental data. The main objective of this paper is to design a VES model-based inspection system that simulates an ECU without software modification in brief time for set-up and execution. The proposed framework has been implemented and successfully applied to a Korean automotive company.

1. Introduction

1.1. Engine-control unit inspection

In the modern automotive industry, the control tasks of various functional aspects in an automotive engine management system are dependent upon an electronic engine-control unit (ECU). An ECU consists of two components: a chipset (the ECU hardware) and an embedded system (the ECU software). The chipset is physically connected to sensors (e.g., brake, camshaft, crankshaft, knock, and oxygen sensors) and actuators (e.g., the cooling fan, fuel pump, ignition coils, injectors, and throttle control valve), and the embedded system employs data received from the sensors to operate the actuators. Since an ECU controls overall systems of a vehicle, ECU failure can be hazardous to entire vehicle systems and human safety. Although ECU integrity has been a crucial issue in the automotive industry (Chen et al., 2013), the complex architecture of an ECU makes ECU quality assurance difficult and hinders vehicle system safety. To ensure ECU integrity, the International Standardization Organization (ISO) provided the standard V-Process concept that covers all of the stages in an ECU life cycle (Kafka, 2012), and this standard defines that verification processes from development to production phases are mandatory.

The ISO standard recommends thorough verification processes for chipset (ISO 26262-Part 5, 2011), embedded system (ISO 26262-Part 6, 2011), and integrated system (ISO 26262-Part 4, 2011) during the ECU development phase. This paper introduces practical methods that supports the verification process of each category as follows. For chipset verification, visual inspection inspects physical defects on a chipset, such as missing-component and open- and short-circuit (Huang & Pan, 2015); thermal testing gives environmental stresses of high temperatures and humidity to an ECU (Lecuyer, Fremont, Landesman, & Bahi, 2010); electromagnetic testing verifies cascading failures caused by electromagnetic interference among the assembled components (Kado, Nelson, & Taylor, 2013).

In the verification of an embedded system, a two-step process is proposed for testing each software module (the first step) and a software system that integrates validated modules (the second step). Methods for the first step investigate the input and output values of a module in operation environment generated by mathematical-models or testers (Arsie, Betta, Capriglione, Pietrosanto, & Sommella, 2014; Bringmann & Krämer, 2008), and detect errors and weakness in an implemented module (Pons, Subias, & Travé-Massuyès, 2015). For the

* Corresponding author.
E-mail address: scpark@ajou.ac.kr (S.C. Park).

http://dx.doi.org/10.1016/j.conengprac.2016.12.001
Received 3 April 2016; Received in revised form 9 September 2016; Accepted 1 December 2016

second step, it simulates a software system in virtual ECU environment (Kasoju, Petersen, & Mäntylä, 2013), and verifies systematic errors among integrated modules (e.g. excessive memory usage, memory allocation crashes, unbalanced task schedules, and timing discordance) and the fault tolerance of an embedded system (Beeh, 2012; Short, Pont, & & Fang, 2008; Zeller & & Prehofer, 2013).

Testing methods for the integrated system of the two ECU components can be categorized into two approaches: functional testing and hardware-in-the-loop simulation (HILS). The functional testing approach sequentially transmits electrical signals into each input function in an ECU and checks the output signals of corresponding functions (Bhattacharya, Biswas, & Mukhopadhyay, 2012). The purpose of this approach is to validate chipset integrity using modules in an embedded system (Lidia, Guangyu, & Weizhe, 2012). The HILS approach runs an embedded system installed on a chipset in virtual vehicle environment constructed by electrical emulating for all sensors and actuators, and it verifies the integration of the two ECU components. Existing methods constructs virtual sensors by collected data from real-vehicle experiments (Palladino, Fiengo, & Lanzo, 2012) or by mathematical models (dSPACE, 2013; ETAS, 2009; Isermann, Schaffnit, & Sinsel, 1999), and emulates real actuators by replicas (Conatser, Wagner, Ganta, & Walker, 2004) or by dummy loads (Lee, Park, & Sunwoo, 2004).

1.2. Research objective

Compared to the development phase, verification processes in manufacturing focus only on inspecting chipset integrity and the serial number of an installed embedded system (ISO 26262-Part 7, 2011). However, ECUs in the manufacturing phase also need the verification process in the unified way of the two ECU components by three major causes: 1) embedded system revision, 2) incomplete installation of an embedded system, and 3) quality variation of chipsets. The first cause originates from continuous ECU revision that is to calibrate an embedded system (Vong & Wong, 2010). Numerous embedded system versions can be generated during the revision process, and it is impossible to validate an ECU like verification in the development phase against every change (Boot, Richert, & Rukgauer, 1999). Thus, the finally manufactured ECU may not operate properly on a chipset or may cause collisions with other functions in an embedded system. The second cause originates from the failure of writing an embedded system on a chipset. If a ROM writing machine did not recognize the failure, the defective ECU cannot be detected without integrated system testing. The third cause originates from the inconsistent quality of manufactured ECUs. Each manufactured ECU differs in quality from the designed chipset (Bettayeb, Bassetto, & Sahnoun, 2014), and non-defective chipsets that meet lower quality limit can be not support the operation of an embedded system. Although defect rate by those three causes could be very low in real manufacturing processes, it is not acceptable for an ECU that is a safety-critical system. Therefore, we need the integrated system testing in a manufacturing system for eliminating the possibility of undetected defective ECUs.

Among many reported methods for integrated system testing, functional testing methods are not capable to verify an ECU as the unified system, and the most HILS methods are only applicable for verification in the development phase (Fathy, Filipi, Hagena, & Stein, 2006). Because the developed HILS methods need much time and effort on experiment set-up and execution, result analysis, and the model-logic modification of crank-position-dependent sensors, they cannot meet the constraints of time, cost, and automation issues claimed in manufacturing environment (Huang et al., 2010). The main objective of this paper is to propose a framework for an HILS-based ECU verification system that is able to apply inspection in the manufacturing phase. The proposed framework builds a test-model that specifies the operations of every ECU function in a simulation, and configures virtual vehicle environment using contained information within a test model. This paper designs “virtual engine system (VES) model” for the test-model of the framework, and it is the core element in the framework to enable automated inspection with brief set-up time for various ECUs. A VES model defines reference vehicle states that decide in the operation states of ECU functions, and determines the operation state transition of each function by the reference vehicle state. In order to identify the operation state transition, we develop a reverse engineering approach that analyzes ECU operation data (including input signals from sensors, output signals to actuators, and communication messages) obtained from real vehicle experiments. By the reverse engineering approach, ECU simulation is possible without any modification. This paper describes the construction of a VES model and the structure of a VES model-based ECU inspection system.

The rest of this paper is organized as follows. Section 2 provides the technical approach of this research with the detail explanation of a VES model structure, and Section 3 describes the framework for the inspection system construction based on a VES model. The implementation results are presented with an example in Section 4. Finally, the conclusions are provided in Section 5.

2. Technical approach

The system requirements for the ECU HILS and the inspection for manufacturing are defined, and key-enablers on the respective requirement are identified as follows.

1. Key-enablers for the ECU HILS:

- Sensor operation modeling: Each sensor transmits sensing data to an ECU using electric signals, and the ECU recognizes the state of a vehicle by processing data from every sensor. Therefore, valid sensor signals of each sensor must be emulated to avoid that an ECU enters into a protection mode against fault input.
- Actuator operation modeling: Each actuator is controlled by electric signals from an ECU. When an ECU cannot transmit the signals to any actuator, the ECU enters into a protection mode against the failure of the actuators. Therefore, devices corresponding to each actuator-control function in an ECU are necessary to receive electric signals.
- Vehicle communication: An ECU communicates with other components by interchanging messages via a controller area network. Sending a message to an ECU is to command ECU operation, and receiving a message from an ECU is to figure out ECU states.

1. Key-enablers for the inspection for manufacturing:

- Predefined simulation scenario: Validated simulation scenarios that are executable during manufacturing are required to define in advance, to inspect every ECU using identical tests in cycle time.
- Simple inspection procedure: Inspection systems in manufacturing systems need simple procedures for test set-up and execution.
- Test result decision: Inspection systems in manufacturing processes have rules to determine whether each ECU is normal or defective based on the test results.

The proposed framework achieves key enablers for the ECU HILS by a VES model that contains the specific operations of all ECU functions, and for the inspection in manufacturing processes by an inspection system structure that emulates defined information in a VES model. The VES model consists of five categories: 1) meta data, 2) chipset, 3) engine, 4) communication, and 5) simulation scenario. The structure of a VES model as shown in Fig. 1, and information in each category is explained follows.

1) The meta data category contains product information to identify an