Measurement 54 (2014) 222–233

Contents lists available at ScienceDirect

Measurement

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

# Functional testing of measurement-based control systems: An application to automotive

I. Arsie [a], G. Betta [b,*], D. Capriglione [b], A. Pietrosanto [a], P. Sommella [a]

[a] Department of Industrial Engineering, University of Salerno,Via G. Paolo II 132, 84084 Fisciano, SA, Italy
[b] Department of Electrical and Information Engineering "Maurizio Scarano", University of Cassino and Southern Lazio, Via G. Di Biasio 43, 03043 Cassino, FR, Italy

ARTICLE INFO

*Article history:*
Available online 1 April 2014

*Keywords:*
Software testing
Functional testing
Automotive
Measurement
Control
Engine

ABSTRACT

The number of measurement-based embedded systems increases continuously in automotive, avionic, defense, and communication industries. In order to ensure the system's compliance with requirements, adequate solutions for functional testing are needed. The paper describes an original approach for the functional validation of measurement-based complex control systems. The main focus is on the automatic generation of realistic continuous data sets (test stimuli) within the operational ranges of the measurement system under test, by applying enhanced statistical techniques, which are able to reproduce both timing and data dependencies. A thorough case study using an air fuel ratio control system from the automotive domain is performed to assess the approach. Results indicate that the proposed method is suitable for automated functional testing of embedded control systems within a Software-in-the-Loop test environment.



## 1. Introduction

Embedded systems are increasingly used in numerous application areas, such as automotive [1], aerospace [2], [3], defense [4] and communication industries [5], where safety is crucial. Thus, greater attention is devoted to testing complex systems during the operational phase in order to detect possible faults and anticipate their negative effects. At present, testing can already take up to more than 50% of the total development cost and effort for a given project [6]. Usually, a significant amount of the tester's time has to be spent on manual test design, test data selection, and test evaluation because most of the tool support available for testing complex embedded systems does not automate these tasks. Nevertheless, no guarantee is given that all errors will be found. Consequently, new methodologies are required to help overcoming this situation, thus increasing both efficiency and effectiveness of the testing process.

This paper aims at improving the functional testing for those complex embedded systems, which are based on measurement information and are employed in process/plant monitoring and control.

Generally speaking, functional testing is considered a sub-category of black-box testing and does not take the structure of the test object into account. Instead, a set of input combinations (the so called test cases, against which the system behavior is examined) are derived from the functional requirements specification and executed by the system under test (SUT). When it comes to large and complex systems, analytical and classical test generation methods often fail because of the combinatorial explosion. To limit the search for test data to critical regions where an erroneous behavior of the system is assumed, statistical testing may be adopted, through several methods derived from experimental design, Taguchi's robust method, orthogonal arrays or probability theory [7–9].

* Corresponding author. Tel.: +39 07762993673.
*E-mail addresses:* iarsie@unisa.it (I. Arsie), betta@unicas.it (G. Betta), capriglione@unicas.it (D. Capriglione), apietrosanto@unisa.it (A. Pietrosanto), psommella@unisa.it (P. Sommella).

http://dx.doi.org/10.1016/j.measurement.2014.03.017
0263-2241/

Additional complications with respect to covering the input domain arise when dealing with measurement software and/or embedded systems. In fact, traditional techniques are mostly suitable to boolean or a few-level discrete variables systems and/or disregarding the correlations existing among inputs. On the contrary, measurement software is typically concerned with processing continuous signal inputs, that are often, (i) varying in wide ranges and, (ii) correlated one each other in a complex way.

In previous papers [10,11], the authors introduced a statistical approach for the functional testing of complex instrument software. In more details, the testing methodology was applied to the diagnostic software of a commercial car engine with the aim of locating faults on sensors included in the charge detection system. The analyzed software operates by processing data coming from several sensors and generating the diagnostic results on the basis of steady-state operating conditions. Therefore, the proposed testing methodology was designed to generate random test cases, each one accounting for the interdependence (in terms of amplitude) among the software inputs, without regard for their temporal order within the sequence to be submitted to the SUT.

Pushed by the good performance provided by the proposed approach, the authors have tackled the problem of extending the method to the validation of a wider class of measurement and control software. In particular, they will deal with software systems for which the test cases have to be designed to take also into account the time interdependence among the software inputs. Starting from the preliminary notes highlighted in [12,13] about the signal acquisition campaign and the correlation matrix estimation, this work relates to the novelties introduced into the validation method and mainly focused on the representation of the input parameters and test case generation.

As an example, the test case considered in the present paper refers to automotive engines whose design and control must comply with stringent regulations on pollutants and, more recently, fuel consumption. Such regulations are defined with respect to standard driving cycles, varying upon the geographical area, which can be significantly different from the real driving conditions experienced by the driver in the day-life use (e.g. depending on the driving style, traffic and road conditions, etc.).

The paper is organized as follows: Section 2 provides the background for the functional testing of automotive embedded systems. Section 3 briefly recalls the main steps included in the statistical approach for software V&V (Verification and Validation), whereas the modified methodology for the functional testing of continuously embedded system is proposed in Section 4. The proposal for generating the test cases is illustrated in Section 5 by taking as application example the validation of a software module for the Engine Management System (EMS) and Section 6 gives conclusion based on the experience gained and the work planned in future.

## 2. Recent trends in functional testing of control systems

Automotive system design, implementation, and verification represent an ongoing industrial research due to complexities posed by such systems: a modern vehicle comprises between 30 and 80 different electronic control units (ECUs) devoted to different areas of application (telematics, infotainment, power control, comfort electronics). Thus, the integration of software and hardware components requires good testing, debugging, and process simulation tools to ensure verification of requirements and system and software design.

An extensively adopted approach is the iterative V-Model development process, which also represents a flow for the functional verification of an automotive embedded system. According to the model, early investigation of a correct behavior is carried out at each design phase of the software-intensive system in order to achieve better and lower-cost product, as well as faster development and testing. More in details, through suitable methodologies, techniques, and tools, the verification process tests and enhance certain features of the system that are relevant at each abstraction level.

The Model in the Loop (*MiL*) simulation is intended for sub-systems such as control algorithms, which are modelled in mathematical notations. The test execution verifies the model before proceeding towards the development process.

In the Model to Software in the Loop simulation (*MiL-to-SiL*), the control algorithm is replaced with the written/generated C-code for simulation and verification in the same environment of the previous phase. Then, the coding process continues toward the simulated/real target. The Software in the Loop (*SiL*) environment provides a verification approach for target code correctness by porting the code; it also provides editors/generators for designing/developing the target description resources. Finally, the Hardware in the Loop (*HiL*) verification implies integrating the on-target implementation (ECU) with the other car components (i.e., sensors, actuators, and plants), whether they are real or virtual, and modelled environments/surroundings.

Although the above framework proves to be very useful in the automotive industry, which is characterized by strongly distributed development processes (among ECU supplier and OEM car manufacturers), a comprehensive and systematic quality assurance of automotive embedded systems is still a large challenge. High manual portions of testing activity (i.e. a low level of automation) are performed through a multiplicity of often proprietary test methodologies, systems and platforms. The latter rely on diverse description and specification techniques, which are only weakly formalized and normally not designed for the exchange between different test systems.

To overpass the above mentioned limitations, investigation efforts have been addressed to the introduction and development of approaches for automatic test case generation. For example, [14] describes an enhanced Classification Tree Method for Embedded Systems (CTM/ES) aiming to the functional verification plan for combined HW/SW automotive systems. The approach supports functional stimulus patterns as well as acceptance and test quality criteria which relate to requirements and enable requirements coverage definitions. In [15] the Model Checking is proposed to automate the generation of test