AN OVERVIEW OF THE ERROR SIMULATION/INJECTION FOR SOFTWARE TESTING

Bhaskar R, Dr K Shanmukha Sundar

Student, Professor & HOD
Dept. of Electrical and Electronics Engg.
Dayananda Sagar College of Engineering, Bangalore, India

Abstract: Error simulator box (ESB) is a simulation box for performing Fault diagnosis. The ESB is connected between the ECU and LABCAR (load). The software's developed are to be tested for three different types of faults are OC, SCB, SCG. The fault injection is controlled via the serial communication between the computer application, a separate user interface is created for different methods of fault testing i,e. event based, trigger based and time based.

Hence electronic based fault diagnosis system ESB is needed so that the faults at the ECU terminals can be performed at desired intervals i.e, at critical software program which cannot be reached by manually creating faults. The Error Simulator Box terminals has to be interlinked with ECU terminal.

Index Terms **- Error Simulation, software testing, Types of faults, short to battery, fault injection using switches.**

I. INTRODUCTION

Automotive embedded control systems in automobiles are exposed to a wide range of environmental conditions including vibration, shock, and electromagnetic radiation, temperature, and humidity variations. Developing reliable and fault tolerant software is difficult and requires discipline both in specifying system functionality and in implementing systems correctly. Approaches for developing highly reliable software include the use of formal methods [9], and rigorous testing methods [2, 7].

While modern electronic control units (ECUs) are extremely robust and provide a high level of safety against any of these external threats, connectors, cable harnesses, and sensors and actuators have become the most vulnerable components in relation to external disturbances. These vulnerabilities triggered the development of embedded diagnostics that monitor and detect electrical faults in a system and activate mitigation procedures when needed.

Manufacturers of Electronic Control Units (ECUs) spend a lot of time and money minimizing the potential impact of any of these threats on their products. As a result, modern ECUs are extremely robust and provide a high level of safety against any of these external threats. Meanwhile, connectors, cable harnesses, and sensors and actuators have become the most vulnerable components in relation to external Introduction disturbances.

For example, ECU connectors are vulnerable to vibration and contact corrosion and a wire harness can be damaged through impact with another object. These vulnerabilities triggered the development of embedded diagnostics that monitor and detect electrical faults in a system and activate mitigation procedures when needed. For example, an anti-lock braking system (ABS) controller continuously checks to see if all wheel speed sensor signals are present and plausible.

When a failure is recognized, the controller will execute diagnostic code that alerts the driver of the problem, and enter a safe system state, possibly followed by a transition into a degraded operating mode. The amount of diagnostic code in a modern ECU is significant. It is often up to half of the total application software. This trend extends beyond the automotive industry to industry sectors such medical engineering, energy management, and telecommunications, all dealing with different legal requirements, not explored further in this paper.

An Engine Control Unit (ECU), also commonly called an engine control module (ECM), is a type of electronic control unit that controls a series of actuators on an internal combustion engine to ensure optimal engine performance. It does this by reading values from a multitude of sensors within the engine bay, interpreting the data using multidimensional performance maps (called lookup tables), and adjusting the engine actuators.

Before ECU's, air fuel mixture, ignition timing and idle speed were mechanically set and dynamically controlled by mechanical and pneumatic means. If the ECU has control over the fuel lines, then it is referred to as Electronic Engine Management system (EEMS). The fuel injection system has the major role to control the engine's fuel supply.

The whole mechanism of the EEMS is controlled by a stack of sensors and actuators. Figure 1 depicts the faulty electrical connection due to corrosion and leads to electrical shorting.

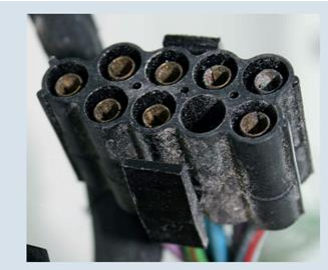

Figure 1: Faulty electrical connections

An ignition system generates a spark or heats an electrode to a high temperature to ignite a fuel-air mixture in the spark ignition internal combustion engines. It includes the battery, ignition coil, distributor, spark plugs and associated switches and wiring. Ignition coil charges battery voltage to a high voltage (30000 volts and greater). ECU receives the feedback signals from the primary side of the ignition coil and it controls the pulse timing.
Fuel injection is the introduction of fuel in an internal combustion engine, most commonly automotive engines, by means of an injector. An electronic fuel injection (EFI) system consists of fuel injectors, fuel pump, fuel pressure regulator, ECU, variable sensors and wiring.

The engine control unit is central to EFI system. The ECU interrupts data from input sensors to calculate the amount of fuel to inject. When signaled by the ECU the fuel injector opens and sprays the pressurized fuel into the engine. The duration that the injector is open (called the pulse width) is proportional to the amount of fuel delivered

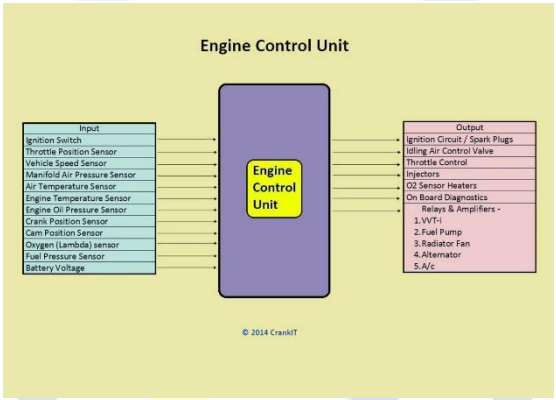

Figure 2: block diagram of ECU

II. OPERATION PRINCIPLES

2.1. DIAGNOSTIC SOFTWARE TESTS

The software within a control unit essentially consists of three parts:

2.1.1 Application software realizes functions noticeable to the user. For example, the idle speed control of an engine, the brake intervention of the ABS, the temperature regulation of the air conditioner, or the automatic station search on the radio.

2.1.2 Low-level platform software enables the implementation of the application software on the control unit. It contains, for example, software services and input-output drivers as well as sensor and actuator controls.

2.1.3 Diagnostic software monitors application software and platform software. In case of a recognized fault, it triggers a certain fault management behaviour, such as a warning to the driver.

Obviously, testing diagnostic software‘s capability to detect electric faults is important. While faults like cable breaks and short circuits can be identified relatively easy, leakage currents, loose connections and contact corrosion are often more difficult to represent.

Because of the many electrical lines and contacts in the vehicle, electric fault tests are very time-consuming and, therefore, very expensive. The various test steps are very similar (e.g., open circuits, short circuits). As it is usually necessary to repeat the tests many times in different variants of a control unit, these tests often become the bottleneck in the development of diagnostic software.
Powertrain controller diagnostics (OBD II) Testing diagnostic software is not only important, it is often mandatory. Since 2005, both the US Environmental Protection Agency (EPA) and the California Air Resources Board (CARB) require that carmakers provide proof that the on-board diagnostic system of every powertrain ECU has undergone a full verification test. This verification needs to be performed on a production vehicle with production ECUs and production software.

CARB testing requirements in Title 13, California Code Regulations, and Section 1968.21 specify in further detail that such tests must include Comprehensive Component Monitoring (CCM). CCM in the context of EPA and CARB regulations requires that the diagnostic software in an engine ECU monitors and detects any electrical fault in a component that can cause a measurable increase in emissions during any reasonable driving condition.
This means that sensor inputs to the ECU must detect electrical faults in the wire harness or rationality faults such as unrealistic signal values.

In addition, all ECU outputs must be monitored for functional faults. For example, the diagnostic software must be able to detect loose contacts or short circuits in the wire harness that connects to the injectors and the throttle valve and execute protective measures if needed.

2.2 BASIC ARCHITECTURE AND OPERATION.

The Error simulator box is placed between the ECU and the load/lab car. The Error simulator has 12 sensor/actuator and 4 sensor pins from the ECU, the outputs of the same are provided to the load/lab car. There is a separate power supply of 12v given to the module. The inputs to the Error simulator box can be used for ignition/injection or any other actuators/additional sensor inputs.

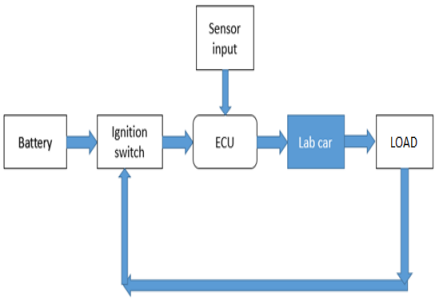

Figure 3: connection diagram of ESB

The simulator is a rectangular box with slots on the side for connecting the sensor/actuator terminals of the ECU.
The Error simulator box has the ECU on the low power side and the load box/ the lab car on the high power side.
The ESB is designed with three modes of controls, they are:

- Event based error simulation
- Time based error simulation
- Trigger based error simulation

In figure 3 the block diagram of the overall working process is depicted. During the normal operation the power flows from the battery to the ignition switch i.e, the power supply for the ECU is provided and thereafter the ECU senses the signal from the sensors and takes corrective actions and hence controlling the electrical loads such as Fuel injection, ignition etc.

Process workflow

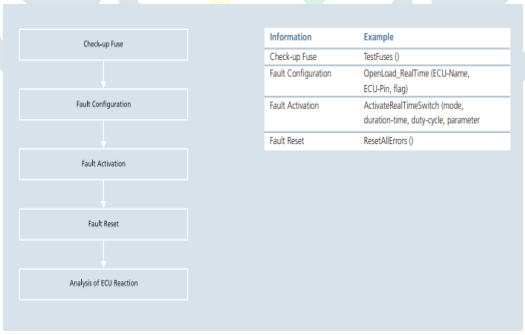

Figure 4: schematic automated test workflow and elements of API protocol

Individual control switches either toggle switch/push buttons with on/off indication lights are provided for all the modes of operation.

In order to execute the event based simulation two switches with separate controls and indication are provided, based on the truth table of the toggle switch/position the de-multiplexer selects the respective faults to be created.
Figure 4 shows the process workflow in an algorithm where the ECU takes the corrective actions and hence injection of the fault and checks the action that is being taken by the ECU.

For time and trigger based error simulation a programmable controller connected to a semiconductor switch which consumes low power, so that the required fault can be created by programming the microcontroller.

The simulation box is provide with on/off ignition switch along with led indications for ignition switch. The size of the simulator is compact and portable. The facility for the simulator box in the actual loads is also provided with short circuit protection.

There are mainly three types of faults those are induced into the ECU they are

- Short circuit to battery
- Short circuit to ground
- Open circuit condition

During the short to battery the ECU signal is directly made to come in contact to the battery by switching the semiconductor device and hence creating a fault at the desired point on the signal of the ECU. Similarly short to ground the ECU signal to sensor/actuator is made to come in contact to the ground by switching the corresponding semiconductor device and hence creating a fault at the desired point on the signal of the ECU.During the open circuit condition the circuit connection is broken down and hence disconnecting the ECU and the lab car/ load of the ECU. During occurrence of the fault the ECU raises a token and stops providing the signal to the ECU based on the time duration of the occurrence of the fault.

II. SYSTEM CONFIGURATION

When developing test cases, the goal is to be able to reuse as many as possible. This also applies to diagnostic function tests. The tests take place during all important development stages: First as the diagnostic algorithms are designed, next after they have been implemented in code, and finally as they are integrated with all other ECU software during Hardware-in-the-Loop (HiL) [7] testing, dynamometer or test cell testing, and finally in production vehicles. This suggests a system configuration for electrical fault testing.

In the figure 5 the schematic configuration of the fault insertion tool is shown where a test program controls the switches to inject a fault. The fault insertion tool is inserted in the cable harness between the ECU and the sensors or actuators. It is able to open and close each individual connection, and to insert a finite resistive value in each one. The fault insertion tool is controlled from a program on a PC. The test program on the PC performs two functions:
First, it commands the fault insertion system to simulate a certain fault, e.g., a broken wire. Next, it measures the response from the ECU. This can be validated by a stored fault code and, if necessary, a defined fault management behavior (e.g., enter “limp home” mode) is executed. The software then compares the ECU response with the expected response and logs a „pass “or „fail “entry for the appropriate test case. These tests can take place in the lab, in the HiL test system, or in a vehicle on a test bench.

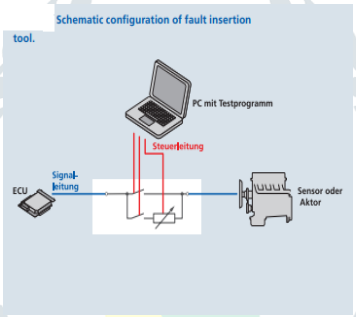

Figure 5: schematic configuration of fault insertion

TYPES OF FAULTS
In order to simulate electrical faults, the following scenarios need to be generated: A cable break, resulting in an open electrical circuit.

- Leakage current and resistance to simulate a voltage drop or an incorrect physical signal.
- Short circuits to the supply voltage or to ground with or without the electrical load still connected. There are two possibilities here: The signal line is either open or closed.
- Short circuits between two lines with or without electric load. One or both lines can be interrupted.
- Leakage currents between two lines. This is a variation of the previously described cases where a small electrical resistance causes the fault
- Contact corrosion by resistance in the line. In addition to the basic fault scenarios described above, temporal aspects need to be considered as well.

The following time dependent use cases exist:

- Multiple simultaneous faults which occur, for instance, when multiple wires in a wiring harness become damaged at the same time.
- Temporary faults which are present only for a defined amount of time. For example, consider a case of loose contacts which only disconnects during certain vehicle movements.
- Intermittent faults which occur, for example, when system vibrations open and close a loose electrical connection periodically. Figure 6 shows that the sensor with load and without load. Without load the ECU is directly connected to the battery and hence resulting in shorting the ECU and injection of the fault. With load only the sensor signal is provided to the sensor or actuator and hence only the power needed for the operation of the actuator is being tapped from the battery.

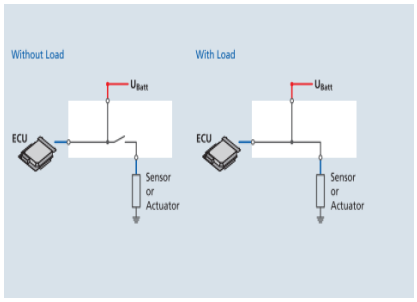

Figure 6: short circuit to the supply voltage with open or closed signal line

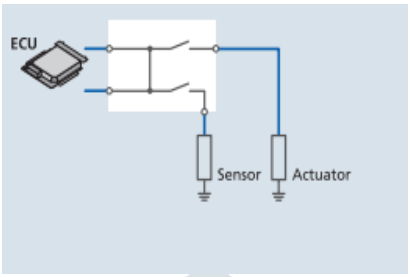

Figure 7: example of short circuit between two lines

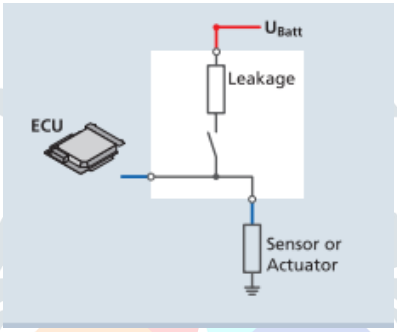

Figure 8: examples for short to battery (SCB)

Figure 7 shows the example of the short circuit between the two lines of either the sensor or the actuator. Due to the vibrations and wear and tear and usage of the automobile there will be certain faulty conditions occurring due to which the ECU terminals are shorted and hence results in the damage of the ECU.
In figure 8 there is some amount of leakage current flowing from the battery to the sensor/actuators. Due to which the terminal voltage applied at the sensor input is reduced and would result in poor operation of the sensor.

III. CONCLUSION

This paper has been used to develop the assertion violation mechanism for inserting faults. The method mutates state by violating specified function pre- and post-conditions. The above discussed is the method of fault injection for creating specific faults, which plays a vital role in the fault diagnosis in automotive applications. In this work the microcontroller does the job of injecting the faults onto the ECU bus terminals. Using this fault injection and simulation techniques are bought at the code level.

REFERENCES

[1] A. R. Khatri, A. Hayek, and J. Börcsök, "Validation of selecting SP-values for fault models under proposed RASP- FIT tool," in 2017 First International Conference on Latest trends in Electrical Engineering and Computing Technologies (INTELLECT), (Karachi, Pakistan), pp. 1–7, IEEE, Nov 2017.

[2] Z. Navabi, Digital System Test and Testable Design. Boston, MA: Springer US, 2011.

[3] M. Alderighi, F. Casini, S. D'Angelo, M. Mancini, D. M. Codinachs, S. Pastore, C. Poivey, G. R. Sechi, and G. S. R. Weigand, "Experimental validation of fault injection analyses by the FLIPPER tool," in 2009 European Conference on Ra- diation and Its Effects on Components and Systems, (Bruges), pp. 544–548, IEEE, Sep 2009.

[4] J. Barton, E. Czeck, Z. Segall, and D. Siewiorek. Fault in- jection experiments using FIAT. (Fault Injection-based Au- tomated Testing). IEEE Trans. Computers, 39(4):575–583, April 1990.

[5] T. A. Budd. Mutation analysis: Ideas, examples, problems and prospects. In B. Chandrasekaran and S. Radicchi, ed- itors, Computer Program Testing, pages 129–134. North- Holland, 1981.

[6] L. J. White. Basic mathematical definitions and results in testing. InB. Chandrasekaran and S.Radicchi, editors, Com- puter Program Testing, pages 13–24. North-Holland, 1981.

[7] H. Yin and J. Bieman. Improving software testability with assertion insertion. In Proc. Int. Test Conf., Oct. 1994.