Simulation of Electrical Faults in the Lab and in the Vehicle

Koos Zwaanenburg, Paul Mohr, Jürgen Crepin

ETAS

ETAS Group
Automotive LifeCycle
Solutions

Content

1   Introduction

Embedded control systems in automobiles are exposed to a wide range of environmental conditions including vibration, shock, electromagnetic radiation, temperature, and humidity variations. For example, control units must be able to endure up to 30 g of vibrations. Ambient temperatures can range from -40 °C to +60 °C, and can even reach over 120 °C for some components.

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 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.

This white paper describes the requirements on a test system which simulates electrical faults of various types. In addition to theoretical examples it will provide use cases of the ETAS fault insertion tools ES4440 and ES4441.

Figure 1:
A common nuisance: Faulty electrical connections.

2 Diagnostic software tests

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

- **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.
- 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.
- **Diagnostic software** monitors application software and platform software. In case of a recognized fault, it triggers a certain fault management behavior, 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.

Numerous laws govern function and test of diagnostic software. In the US, OBD II (On Board Diagnostics) regulations are prevalent and in the European Union the EOBD (European OBD) prevails. Other countries have similar requirements which follow OBD II, such as KOBD (Korea) or JOBD (Japan). Some non EU-countries have adopted EOBD (Switzerland) or are planning to adopt it. The focus in this paper is on OBD II.

2.1 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, Section 1968.2[1] 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 The costs associated with diagnostic tests

A quick illustration of the expenses caused by diagnostic tests: A modern automotive engine controller of a medium-sized vehicle has around 200 input and output pins. Let's assume that more than three quarters of these 200 pins carry OBD-relevant signals. Let's also assume that the diagnostic software must be tested for correct responses to each of these pins being short-circuited to the battery voltage, to ground, and an open circuit. This results in 450 test cases that must be specified, executed and evaluated, just for the CCM-part of the CARB/EPA Certification Test

In many instances, even more test cases are required, because, for example, the time-dependent behavior of the diagnostic software must be considered, or other kinds of faults, such as contact corrosion or loose contacts come along. Automotive manufacturers and their suppliers often perform hundreds of ECU releases per year, resulting in millions of dollars being spent on diagnostic testing.

[1] http://www.arb.ca.gov/regact/obdii06/19682clean.pdf

2.3   Development of the diagnostic software

The development of diagnostic algorithms for engine ECUs can often be broken into three separate phases. The first phase is the development of the basic algorithm. During the second phase of development, the diagnostic algorithm developed in phase 1 is used multiple times with minor modifications across different engine families. For example, the algorithm may be used with a different number of cylinders or different fuel systems. In the third phase, the algorithm is applied to every variant within the engine family. For example, variants may have different horsepower or torque characteristics.

Testing of diagnostic software is often carried out during all three phases of the development cycle in order to detect software faults in time.

2.4   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) testing, dynamometer or test cell testing, and finally in production vehicles. This suggests a system configuration for electrical fault testing as shown in Figure 2.

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 (see Figure 3).
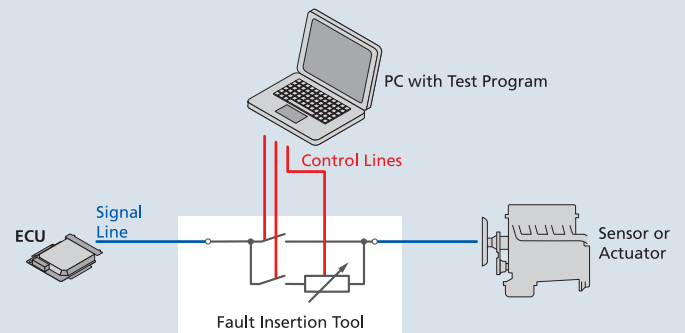

Figure 2:
Schematic configuration of fault insertion tool.

Figure 3:
Electric fault simulation in the vehicle.
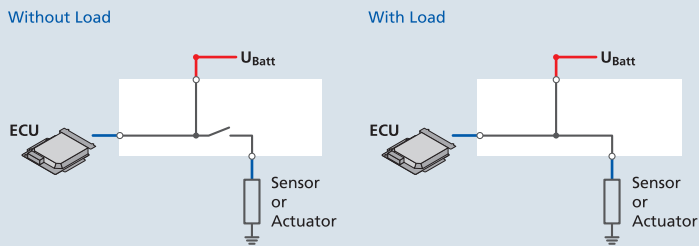
3 Types of electrical faults

Without Load

U_Batt

ECU

Sensor or Actuator

With Load

U_Batt

ECU

Sensor or Actuator

Figure 4:
Short circuits to the supply voltage with open or closed signal line.

ECU

Sensor   Actuator
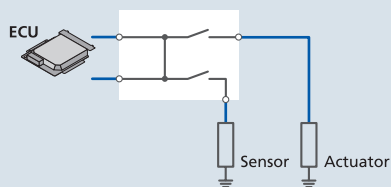
Figure 5:
Example of a short circuit between two signal lines.

U_Batt

Leakage
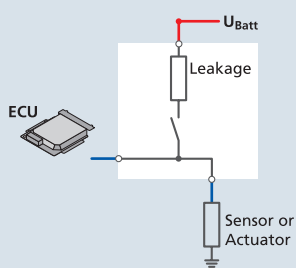
ECU

Sensor or Actuator

Figure 6:
Leakage current.

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 (see Figure 4).
- Short circuits between two lines with or without electric load. One or both lines can be interrupted (see Figure 5).
- Leakage currents between two lines. This is a variation of the previously described cases where a small electrical resistance causes the fault (see Figure 6).
- 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.

4 Requirements for electric fault simulation

5 Simulation of complex faults

The systematic test of the system behavior in case of an electrical fault places high demands on the capabilities of the test system:

- **Support of high voltages**
  High voltages such as in piezo injectors or other power devices must be switched safely.

- **Ability to automate**
  The potential to automate the tests allows for an economical realization of a large number of OBD tests.

- **Reproducibility**
  The tests have to be reproducible in order to detect software faults systematically and to confirm their elimination without any doubt.

- **Ease of use**
  The configuration of the test system, the test execution and the test analysis must be possible with low effort to allow efficient work.

- **High degree of reuse**
  Tests must be able to reuse across several projects to keep test creation cost effective.

- **High signal quality**
  The quality of the signal must be "good" to eliminate false positives. "Good" in this case means that the switches must be exact in time and the expected current and voltage has to be maintained precisely.

- **Fast switching times**
  For real-time systems and dynamic faults, switching times must be fast and deterministic.

5.1 Simulation of contact corrosion

Contact corrosion is a particularly problematic fault that occurs in the vehicle. It can be simulated with the connection of a configurable resistance cascade. It is important to switch the resistance cascade into the signal path correctly, to avoid the detection of a "wrong" error in the ECU. A "wrong" error could occur, for instance, when the switching of the resistance cascade affects the signal in the wrong way, such as in a short interruption at the switch.

5.2 Simulation of sensor drifts

Over time, the voltage level of the vehicle sensors can drift away. This can influence the control and diagnostic functions in the ECU and therefore, sensor drifts have to be considered during ECU development. To test for sensor drifts, specially calibrated sensors are used that simulate the aging effects on the limits of tolerance. They are installed into the test system instead of "good" sensors.

Another option to simulate sensor drift is the connection of a resistor against the positive or negative battery voltage (pull up/pull down resistor). The voltage drop at the resistor moves the analog signal in the positive or negative direction. With a resistor cascade, the deflection can be modified over a wide range. This procedure allows comprehensive and efficient testing for sensor drifts and eliminates the need for costly "bad" sensors.

5.3 Test of timing functions

ECU diagnostic functions take place on multiple levels. Depending on the type, the duration and the defined severity of the detected fault, a corresponding function in the ECU is executed.

For example, let us assume that the ECU detects an abnormal behavior of a sensor. If this failure is detected only for a short time, only an entry in the fault storage is made, but no activation of a fault management behavior takes place. Subsequently, if plausible values are registered again, the fault storage entry will be deleted after a defined time. This ensures that a short interrupt, e.g., caused by an EMC disturbance of a sensor, will not be an unnecessary nuisance for the driver.

To test the behavior of an ECU with multi-level failure reactions, it is necessary to have flexibility in selecting the duration of faults.

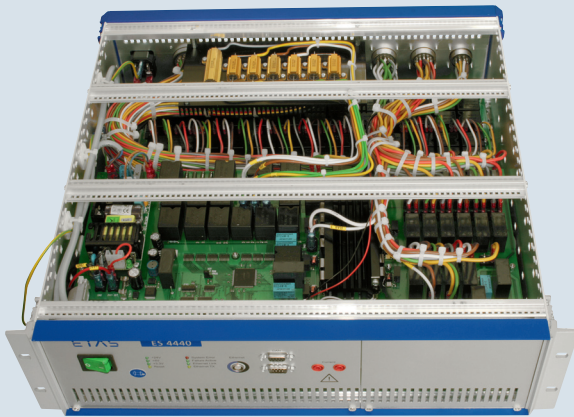# 6 Hardware requirements for a fault insertion tool

Figure 7:
The hardware of a failure insertion tool has to fulfill various requirements.

To assure comprehensive and economic testing of diagnostic software to all electrical faults and erroneous physical measurements described in the previous section, fault insertion tool hardware has to fulfill numerous requirements. Depending on the application, these requirements include:

- It needs to be portable so that it can be shared between different test systems and ECU development projects. Portability means, first, that the fault insertion tool comes with its own universal power supply so that it can work independent of any other equipment in the test bench. Second, it implies that it has on open control interface, such as Ethernet or CAN, connected to the user's workstation or laptop.
- Since the voltage and current ranges and impedance levels of sensor and actuators vary widely, the fault insertion system needs neutral electrical characteristics. This includes very low internal resistance and capacitance. It allows switching of low voltage sensor signals and communication networks such as CAN, LIN, FlexRay, etc.
- It needs the capability to connect to multiple battery voltages. Many modern electronic controllers use more than a single supply voltage and ground. For example, some heavy-duty trucks have 12 Volt and 24 Volt batteries in the vehicle. To test such a vehicle, one needs to be able to create short circuit conditions to both battery voltages and both grounds.
- The fault insertion tool needs to be able to insert faults in high-voltage signals such as piezo injectors which can reach up to roughly 400 V.
- It needs to be able to conduct high currents.
- To protect itself, the test bench, and the ECU under test, the fault insertion tool requires sufficient local intelligence to reset safe operating conditions in case short-circuit currents are too high, or in case heat dissipation crosses design limits. In addition to protective software, it also requires fuses.
- It should provide the simulation of cable breaks, shorts to power and ground, pin-to-pin contacts, and the use of programmable resistors in-line, between wires in the harness, and connected to battery voltages or ground.
- Oftentimes, embedded diagnostics must be activated after a fault condition has been present for a certain amount of time. To test that diagnostics start in a timely manner, accurate control of a fault's duration is necessary. Since the activation times of diagnostics can be very short, 10 to 100 ms, the fault insertion tool must be able to switch rapidly with a known switch time.
- In addition to short and deterministic switching times it also must be capable of producing intermittent electrical faults. For example, it needs to able to produce an intermittent contact, e.g. 2 Hz frequency and 50 % duty cycle.

- It should be able to make a ground wire in the harness float with respect to the system ground. This allows simulation of the effect of finite resistance in the power supply return lines, caused by dirt or corrosion.
- The fault insertion tool must allow easy access to the fuses.

6.1 Relay versus MOSFET – Two technologies for the electronic failure simulation

6.1.1 Fast and accurate circuit times with MOSFET technology

Oftentimes, the signal must be interrupted at a defined time (e.g., at a defined speed during the acceleration phase). Voltage controlled MOSFETs (Metal-Oxide-Semiconductor Field-Effect Transistor) offer these exact and contact bounce-free switch times. Since MOSFETs don't wear like relay switches, they can simulate loose contacts with many switch cycles without any problems (see Figure 8).

The disadvantage of MOSFET circuits lies in their inherent leakage current. In open state, the internal resistance is not infinite. The signal will not be cut absolutely. So despite the speed advantages of MOSFET technology, it cannot completely replace relay technology.

6.1.2 High voltage with relay technology

In the vehicle, high voltages up to 400 V can occur in the activation of the ignition coils or piezo injectors or at the electric power system of hybrid vehicles. Faults in the area of these high voltage signals can be simulated with relays. Relays are able to cut connections completely. However, the switch times are slower in comparison with MOSFETs and are subject to contact bounce.
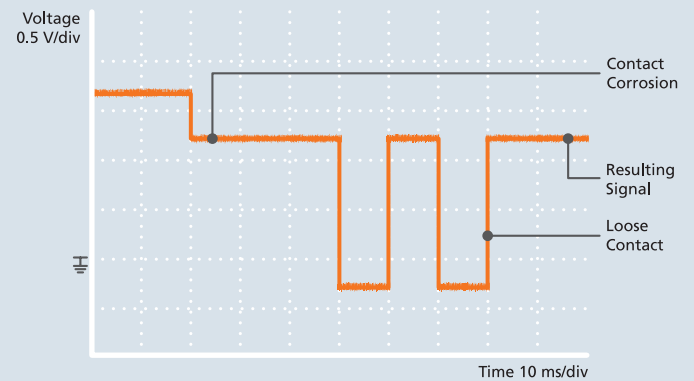

Figure 8:
Simulation of contact corrosion superimposed on a loose connection.

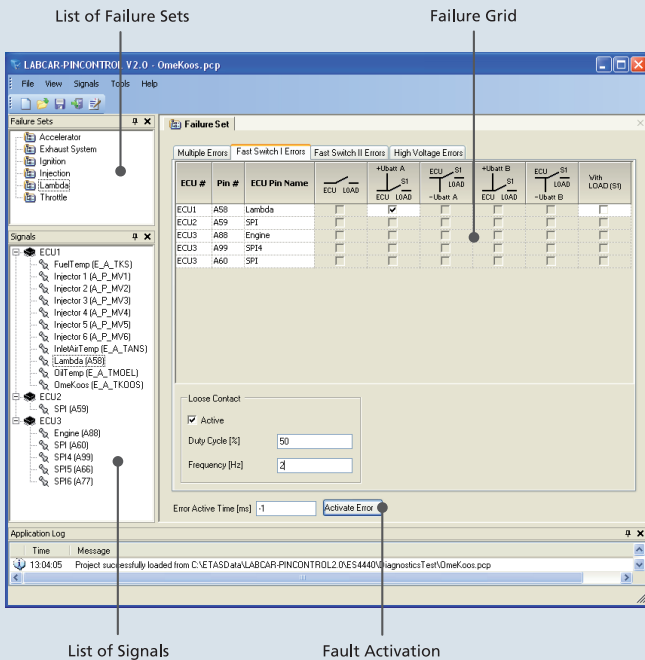# 7   Software requirements for a failure insertion tool

List of Failure Sets   Failure Grid

Figure 9:
GUI of the ETAS ES4440.

List of Signals   Fault Activation

To fully utilize the hardware possibilities described in the previous section and to ensure efficient, safe operation, the software of a fault insertion tool has to provide numerous features, depending on the application:

- Keeping track of 200 plus individual I/O pins of a modern ECU, while inserting faults can be cumbersome and is error-prone. At the same time most – if not all – pins have specific signals names associated with them, e.g. "Battery Plus (30)" or "Lambda Signal". Therefore, it is desirable to be able to refer to signals by their names rather than by a pin number. Stated differently, it is less error-prone to submit a request to "short the Lambda Signal to Battery Plus", than have the indirection with the pin number "short pin 79 to battery plus". Spreadsheets combined with smart control software can provide this capability.
- For convenient operation, the fault insertion tool needs an easy-to-use graphical user interface (GUI). This GUI must be able to perform the following functions:
  - Read in wire harness and signal name data.
  - Select the temporal aspects of the fault. For example, select either a static fault, a fault with a certain fixed duration, or an intermittent fault with defined values for frequency and duty cycle.
  - Activate and deactivate a selected fault.

Figure 9 shows a GUI of the ETAS ES4440 as an example. With this GUI, manual tests can be executed, configured and debugged in a convenient manner.

At bottom left in the GUI, it shows a named signal list. This list is entered into the GUI application from a spreadsheet or an xml file. Once the named signals are available in the signal list, the user can drag and drop individual signals to the failure grid. Once there, the user can select specific faults and activate them. In the example shown in Figure 9, the choice was made to connect the Lambda signal on pin A58 on ECU1 to the first battery voltage U_{batt} A. Note that the last column in the failure grid specifies that the load must stay on during the fault. In the activation window it shows that the fault will be an intermittent one with a frequency of 2 Hz and duty cycle 50 % to simulate a loose contact.

- The definition of "failure sets" simplifies work. Failure sets are configurable groups of ECU I/O pins that are related to a defined ECU function. For example, there could be 5 ECU pins that work together for the air-fuel ratio measurements performed by the oxygen measurement of the lambda sensor. While testing the lambda sensor diagnostics, there is no need to see all the other ECU I/O pins. Figure 9 shows that upon selection of the failure set named Lambda, the five relevant signals show up in the failure grid. The user can now concentrate on those signals only.
- Access controls protect the ECU under test from mistakes during experiments. Signals lines that are not in the signal list cannot be selected inadvertently and short-circuited. With this preconfiguration it is easy to prevent access to certain wires in the harness.
- Large numbers of diagnostic test cases require a high amount of automated test sequences. Therefore, the fault insertion tool requires an open and documented Application Programming Interface (API) that can be controlled from programs or scripts. This API must support all functions of the GUI (see Figure 11).
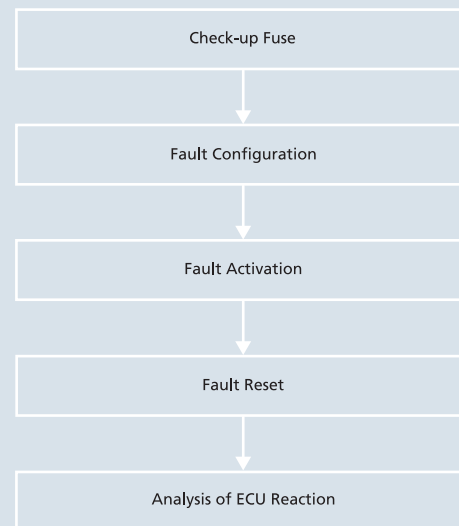- A self test with status report is also necessary.

Figure 10:
Schematic automated test workflow.

Information	Example
Check-up Fuse	`TestFuses();`
Fault **Configuration**	`OpenLoad_RealTime(ECU-Name,ECU-Pin,flag);`
Fault **Activation**	`ActivateRealTimeSwitch(mode,duration-time,duty-cycle,parameter);`
Fault Reset	`ResetAllErrors();`

Figure 11:
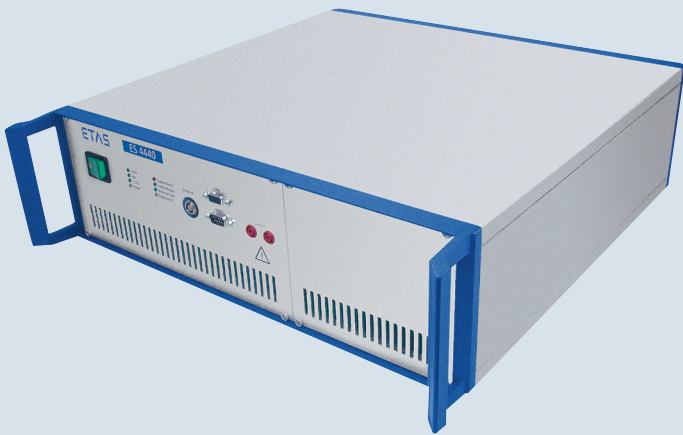Elements of an API protocol.

8   ETAS fault simulation modules

Figure 12:
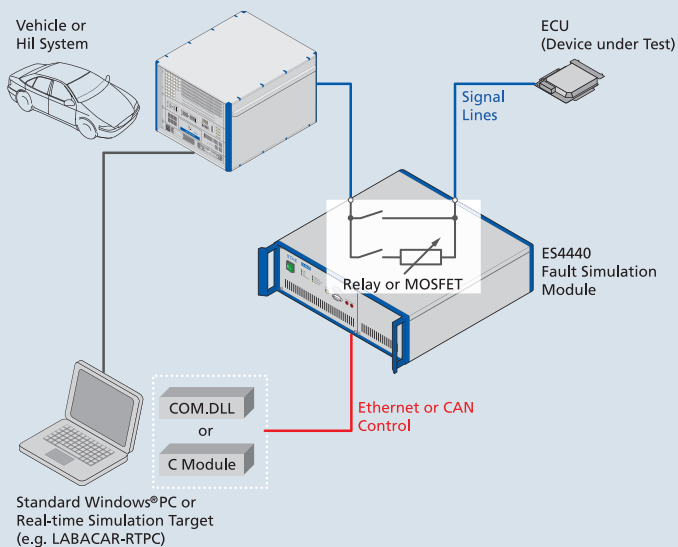ETAS' portable real-time Fault Simulation Module ES4440.

With ETAS' patented, cost-effective fault simulation modules ES4440 and ES4441 all above described types of electrical faults can be simulated in a perfectly reproducible way and – if needed – automated. The configuration flexibility allows a wide range of applications.

8.1   ES4440

ETAS' portable, compact ES4440 Fault Simulation Module (see Figure 12) provides all the necessary capabilities. It allows for testing of system behavior in case of electrical faults as well as in the Hardware-in-the-Loop (HiL) testing system, in the stationary vehicle, or with the vehicle on an engine or chassis dynamometer. Especially in case of safety critical driving conditions, automated HiL tests are an important part in the quality insurance so that components and systems can be tested and documented reproducible with great detail.

All kinds of failures can be inserted either static or intermittent. Therefore the ES4440 is the ideal tool for software release tests based on standardized specifications, e.g., the required tests for OBD II, CARB or EOBD. The control can be done via Ethernet or CAN interface (see Figure 13). Both interfaces are supported by the open API.

The ES4440 Fault Simulation Module is integrated in line into the signal line. The relay, the MOSFET or the resistor cascade affect the respective input or output signal according to the configuration. This is done on a standard Windows PC or directly in the simulation target of the HiL system. The configuration is stored on COM.DLL or C modules. The communication to the vehicle or the HiL system can be realized via CAN or Ethernet.

Figure 13:
Schematic circuit of the ES4440 Fault Simulation Module.

64 MOSFET channels for high current (30 V, 20 A) and 16 relay channels for high voltage (400 V, 10 A) are available in one unit (see Figure 14). The 14 programmable resistor cascades can be changed in from 2 Ω increments from 2 Ω to 16 kΩ. Several units can be connected to a compound. Thereby one ES4440 is configured as master and the other ones as slave.

The ES4440 allows the simulation of all relevant fault types ant their combinations: Cable breaks, short circuits to ground, battery voltage and between two lines, leakage currents, time limited or permanent loose contacts, galvanic corrosion, and sensor drifts. All this can be simulated in the vehicle or integrated in all common Hardware-in-the-Loop (HiL) testing systems. Configuration and control can be carried out manually or automated in the real-time system with the PINCONTROL software.

8.1.1 Combined applications

The compact ES4440 Fault Simulation Module from ETAS offers the possibility to use MOSFET and relays in complete separate circuits simultaneously. Hence it is possible to realize highly dynamic signals, high voltages as well as complete disconnection of signals in one unit, depending on the needs. The change between MOSFET and relays can be done with the easy to use, universal configuration. A change of the hardware or the cabling is not necessary. The circuit time is 5 ms for relays and 200 µs for MOSFET.

8.1.2 Simulation of contact corrosion

The exactly switch of the resistor cascade without interrupt at the user defined time is carried out in the ES4440 with a second signal path that is parallel to the standard signal path (input-output). This second path contains the resistor cascade and all other supported fault types of the ES4440. After the configuration and activation of the second signal path, the direct standard signal path is deactivated. So the second signal path enables a distortion free, exact switching of the fault (see Figure 15).

8.1.3 Simulation of sensor drifts

The ES4440 offers a simple way to simulate sensor drifts. A resistance is switched to the analogous sensor signal against the positive or negative battery voltage (pull-up/pull-down resistance).
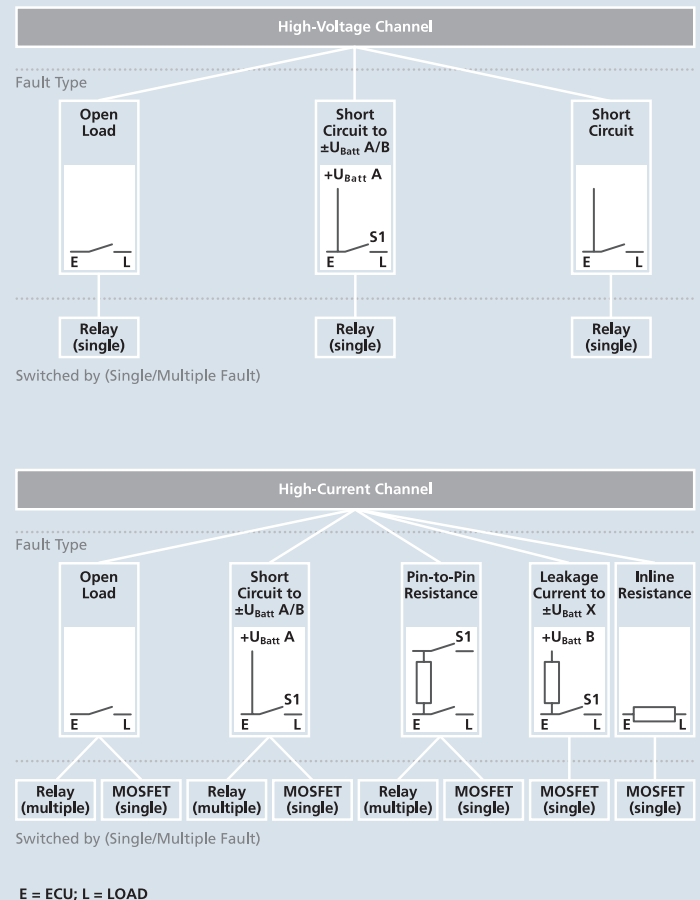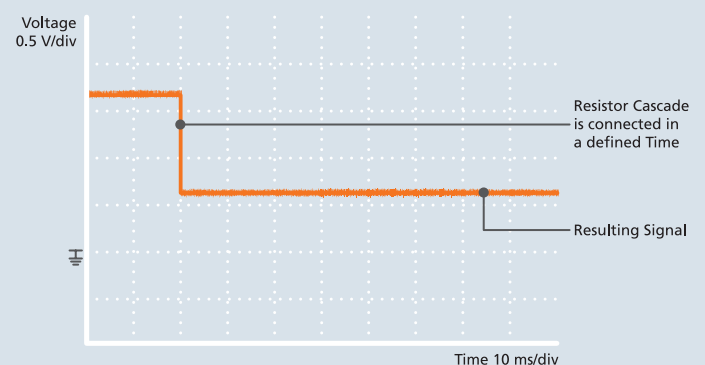

Figure 14:
Switch possibilities of the ES4440.

Figure 15:
High signal quality of the ES4440 during simulation of contact corrosion.

Figure 16:
ETAS' real-time Fault Simulation Module ES4441 with breakout box.

8.1.4 Test of timing functions

With the ES4440 it is possible to apply a time limit to any desired fault, starting with 1 ms. This way, the behavior of the control unit can be tested accurately, even with multi-level fault reactions. If for example, the standby function was released after 10 ms, faults with 8 ms and 12 ms duration can be looped in and the reaction of the control device can be recorded with the ES4440.

8.2   ES4441

ETAS also offers a larger failure simulation tool which is integrated into a break-out-box: the ES4441 (see Figure 16). In contains a breakout adapter in which the appropriate jumper can be tapped. This allows integration, for example, of a voltmeter or ammeter or an oscilloscope without additional modification of the cable harness. In this way, useful information of the signals can be easily obtained. The ES4441 also offers the possibility to change the ground by switching a programmable resistance into the ground line.

The ES4441 offers 255 channels for up to 30 V (202 channels for 1 A, 15 channels for 3 A, 31 channels for 7.5 A, 2 channels for 15 A, 4 channels for 20 A, 1 channel for 40 A). The programmable resistance cascade can be changed in 0.5 Ω increments between 0.5 Ω and 128 kΩ.

PINCONTROL , the corresponding software to the ES4441, offers the user interface for the configuration as well as for the manual and automatically operation with the communication API.

For more information, go to **www.etas.com/ES4440** or contact your local ETAS representative.

9 Tips and tricks

10 Explanation of used abbreviations

In many applications, it often makes sense to develop the control in two steps. In the first step the type of error is preconfigured, the resistance cascade is adjusted to the desired value if necessary, and the second signal trail is switched on (see "8.1.2 Simulation of contact corrosion"). This first step is not time-critical as it takes place independently. In the second step, the preconfigured fault will be switched on at a desired moment. In the case of a real-time simulation this can happen quickly and precisely. The two-stage control therefore allows exact time response with very good signal quality.

API   **Application Programming Interface**

Interface provided by a software system to enable other programs accessibility to the system.

CARB   **California Air Resources Board**

Government agency of the US Federal State of California in charge of air pollution control.

CCM   **Comprehensive Component Monitoring**

→ CARB test demand that states that an appropriate monitoring must exist with components relevant to waste gas in the motor vehicle.

ECU   **Electronic Control Unit**

EMC   **Electromagnetic Compatibility**

EMC tests check the robustness of an electric system against electromagnetic radiation (e.g., from broadcasting stations) and the radiation of the electronics.

EPA   **US Environmental Protection Agency**

GUI   **Graphical User Interface**

MOSFET   **Metal-Oxide-Semiconductor Field-Effect Transistor**

OBD   **On-Board Diagnostics**

From the → CARB introduced legislation for the diagnosis of components relevant to waste gas in motor vehicles. Similar demands exist in other countries. Occasionally, the term OBD is also applied for electronic diagnosis in general.

OEM   **Original Equipment Manufacturer**

Vehicle manufacturer

Did you know?

- The wiring harness of a passenger car consists of appr. 350 to 1500 different cables with a total length of about 765 to 3500 yards[1].

- The 70-250 connectors in a passenger car connect between 700 to 3000 contact pins[1.]

[1] **Source: Robert Bosch GmbH**