

Vehicle Level Approach for Optimization of On-Board Diagnostic Strategies for Fault Management

Dibyendu Palai
Tata Motors Ltd.

ABSTRACT

As the vehicle functions are getting distributed over multiple ECUs in order to realize various complex control functions, the need for sophisticated on-board diagnostic strategies are increasing in automotive domain, leading to a significant amount of hardware and software implementations for fault management inside various ECUs in the vehicle.

This paper proposes optimized vehicle level approach for fault management strategies, wherein a centralized intelligent Gateway Module is proposed in the vehicle network architecture, which will be responsible for fault management of the complete vehicle in a chronological sequence. This Gateway Module will thereby have the possibility to group a cluster of faults raised by different ECUs and correlate them meaningfully to guide the operator towards root cause of the fault. This paper also proposes how such an intelligent gateway module will capture, classify and correlate the fault information from the distributed network architecture, leading to efficient filtering of primary faults of the vehicle. The primary fault records of the vehicle will then be managed within the error memory of the Gateway Module throughout the complete life cycle in a chronological sequence.

The proposed on board diagnostic strategies in this paper are not only aimed to provide off-board guidance to technicians, enabling enhanced guided diagnostics in the field service, but also to provide on-board safety oriented perspectives during vehicle's operation. This will allow the ECUs to go to selective degradation mode, without completely shutting down their critical control functionalities, which might lead to safety issues and customer dissatisfaction during vehicle's operation.

CITATION: Palai, D., "Vehicle Level Approach for Optimization of On-Board Diagnostic Strategies for Fault Management," *SAE Int. J. Passeng. Cars – Electron. Electr. Syst.* 6(1):2013, doi:10.4271/2013-01-0957.

INTRODUCTION

A rapid evolution is observed in on-board vehicle diagnostics over last couple of decades, not only to satisfy stricter regulatory norms in different geographical markets, but also to incorporate various safety requirements in next generation vehicles and to improve customer satisfaction by reducing mean time to repair a vehicle in field service [3].

This paper is in the field of on-board diagnostics where different Electronic Control Units (ECUs) monitor, manage and store various types of fault records in order to help service technician to retrieve fault records stored within those ECUs and identify root cause of the fault quickly. The proposed method is applicable for all such distributed network architectures, wherein ECUs are connected to vehicle communication networks such as HS CAN, LS CAN, LIN, etc.

The need for failure diagnostics in automobiles originates from two perspectives: a maintenance-oriented perspective and a safety-oriented perspective [2, 4]. The maintenance-

oriented perspective deals with improving service diagnostics to aid service technicians to find the root cause of the fault quickly and reduce warranty costs, when the vehicle is in an off-board mode. The safety-oriented perspective is focused with supporting the overall requirements of safety-critical systems, typically when the vehicle is in an on-board mode. The scope of failure diagnostics is much beyond various fault detection techniques, because a significant amount of extended information is provided by failure diagnostics methods on the underlying cause of a system failure, thanks to the greater wealth of information available nowadays on a vehicle's communication network [1]. Typically fault detection techniques are mainly limited to determine whether some fault has occurred or not, on the other hand failure diagnostics is focused to capture what kind of fault has occurred, what components are responsible, and what has caused the fault. This extended information is very much useful for service technicians during field service and also at run-time of the vehicle to allow various systems and sub-

systems of the vehicle for more robust failure mitigation mechanisms.

One of the existing methodologies for failure diagnostics deploys model based diagnostics approach [6], in which the control system to be diagnosed is modeled based on the physics of the system, and this model of the system is executed at run-time along with the actual system. State observers are used to estimate state variables that cannot be measured directly. At run time, the actual outputs of the control system are compared with the outputs predicted by the model. Residual values are calculated as the differences between the actual and predicted outputs. Threshold values of the various residuals or of combinations of residuals are correlated at design time with various anticipated faults from the fault model. During run time, whenever such residual values are encountered that exceed predefined thresholds, the corresponding fault is considered to have occurred. The limitations of this approach arise from the additional computational expenditure resulting additional CPU requirements, which are required for on-line modeling and from the sensitivity of the diagnosis system in modeling unavoidable real life faults. Such model based diagnostic approaches are typically able to detect sensor faults, actuator faults, and faults in the mechanical, electromechanical, or hydraulic plant, which is being controlled. However, these model based diagnostic strategies normally assume that the computing hardware on which the control algorithms are getting executed, the serial communication links with which ECUs communicate, the control software, memory devices, etc do not fail and these are assumed to be always operating in their fault-free modes. These assumptions are not correct in practical scenarios.

Existing methodologies for fault management are also mainly based on ECU specific approach and does not focus on vehicle level approach. The problems associated with existing practice are that for any given fault, multiple Diagnostic Trouble Codes (DTCs) are registered by multiple ECUs which result in complexity and confusion for service technician. This in turn leads to increased mean time to repair a given fault in the field service. Yet another limitation with existing methodologies is that various fault detection mechanisms in vehicle networks mainly focus on different network management strategies, which mainly deals with co-ordination of wake-up and sleep for different ECUs, monitoring presence (or absence) of partner ECUs through message monitoring approach, and monitoring different physical short-circuit and open-circuit faults of network bus systems. It does not focus on correctness of the data received on the network and monitor the root cause of such faults over various vehicle maneuvering conditions.

In this paper a vehicle level holistic approach for fault management and suitable vehicle network architecture to support such fault management approach are presented.

DETAILED DESCRIPTION OF THE METHOD

Terminologies

Following terminologies are defined to explain the proposed method.

Test: An on-board diagnostic routine that determines the status of a component or system. A test may run continually, or only once during an operation cycle. A test returns a result based on whether a fault is present, and whether it has reached maturity. Each test is associated with a single DTC. Test can also be termed as “monitor” or “diagnostic function”.

Complete (test): An indication that the test was able to determine whether a fault was present (or not present) in the current operation cycle.

Prefailed (test): An indication that the test is currently maturing a fault condition. The prefailed state may be used at vehicle EOL testing as an early indicator of malfunction.

Failed (test): An indication that the test ran to completion and it is determined that a fault was present and fully matured.

Passed (test): An indication that the test ran to completion and it is determined that a fault was not present.

Pending: An indication that the test has reported a “failed” result during the current and / or the last completed operation cycle. Once the test has reported a “passed” result for a complete operation cycle, the pending status is reset.

Monitoring: It is the process of running tests to monitor the status of systems and components.

Monitoring cycle: The Period during which a test runs, i.e. the test sequence is executed, is termed as Monitoring cycle. A monitoring cycle may be executed several times during an operation cycle.

Operation cycle: An operation cycle defines the start and end conditions for tests to run.

For non-emissions related ECUs, the operation cycle starts when the operation of the ECU starts (i.e. after completion of power-up initialisation) and stops when the ECU has initiated its shut-down procedure. If the ECU is always powered up and never initiates shut-down procedure, then the operation cycle will begin when the ECU goes from Accessory mode into IGN ON mode (and either stays in this mode or goes immediately to engine running mode). The cycle will end, and another cycle will begin, the next time this transition occurs. The cycle will also end if the ECU is completely powered down (all power inputs removed). Another cycle will begin when power is restored. When a new operation cycle begins following complete power down and restoration of power, this cycle will continue in the event of a transition to Active mode as above. This cycle will end at the next such transition, or if the ECU is again completely powered down.

For emissions-related ECUs, the operation cycle (driving cycle) starts when the operation of the ECU starts (i.e. after

completion of power-up initialisation) and stops when the next such ECU start-up event occurs. The cycle must include an engine shut-down event and subsequent start-up event. The cycle will ignore engine shut-down and restart events resulting from engine control strategies (e.g. engine shutoff at idle.) Engine start-up following a stall or other such event (not caused by an engine control strategy) will be considered valid events for this cycle.

Driving cycle: A type of operation cycle which is used for emissions-related ECUs and defined by legislation; uses an engine-running or engine-off time period to define the cycle.

Power-up cycle: The period during which the ECU is continuously connected to its principal power supply.

Long-term memory: Non-volatile memory, e.g. EEPROM, or battery-backed RAM capable of maintaining stored data for a minimum of 60 days while disconnected from its power supply.

Short-term memory: Volatile memory (e.g. RAM) used to hold transient data (e.g. fault detection counter).

DTC status byte: Additional DTC information that identifies the ongoing status of the DTC; this is related to the test cycles used to determine the fault condition.

DTC snapshot data record: A record of data (DIDs/PIDs) associated with a DTC event. This is also known as freeze-frame data.

DTC extended data record: A data record that stores specific information assigned to a fault.

Fault record: Fault record will consist of DTC (base code and DTC Failure Type Byte), DTC status byte, DTC extended data record, DTC snapshot data record. The term “fault record” applies to the DTC data whether held in short-term memory or long-term memory.

Aging: Unlearning/deleting of a no longer failed event/DTC from event memory after a defined number of operation cycles.

Primary fault: A fault that occurs independently of any other faults.

Secondary fault: A fault that is triggered by the occurrence of another fault.

Pending DTC: A pending DTC relates to a fault that has been detected and matured in the current or previous operating cycle but has not yet been confirmed.

Confirmed DTC: A confirmed DTC is a DTC that provides an indication that the test has reported a “failed” result on enough consecutive operation cycles, such that it should be accepted as a real fault. The DTC remains confirmed until aging has been completed.

Current DTC: DTCs those are either prefailed or pending or confirmed is together defined as current.

Historic DTC: A historic DTC relates to a fault that was detected in a previous operation cycle and resulted in a current DTC, but has not been detected in subsequent operation cycles and has completed aging.

Permanent DTC: A DTC that is latched so that it remains in memory until cleared by the on-board system. This type of DTC cannot be cleared by the tester, and is mandated by

OBD for emissions-related faults where the MIL must be illuminated.

Objective

The main objective of this paper is to pin point the root cause of the faults in order to allow systems to go to selective degradation mode during vehicle running conditions in onboard mode of operation, and also to help service technician to reduce mean time to repair a fault, when the vehicle is diagnosed in service station in off-board mode of operation.

Assumptions

It is assumed that various ECUs are connected over serial communication networks (such as CAN, LIN, etc) to share their data with other ECUs in order to realize various distributed vehicle functionalities.

Proposed Method

Vehicle level optimized fault management strategy and supporting network architecture are proposed in this paper.

Fault classification

In distributed vehicle network architecture, different ECUs share thousands of signals with other ECUs through vehicle's communication networks in order to realize distributed control functionalities. Faults in such distributed network architecture could be classified as follows:

Internal ECU faults typically encompass ECU memory faults, open / short circuit / stuck contact faults in ECU internal relay / MOSFET circuitry, faults due to ECU over temperature, current overload, etc.

External related system faults are the faults which are located in all circuits connected to the ECU and which usually include switch stuck faults and unexpected or implausible value for any switch input to ECU; short to battery voltage, open circuit, short to ground, unexpected or implausible value or rate of change for any analog input to ECU; short to battery voltage, open circuit, short to ground, unchanging position feedback signal for any output with high side or low side driver, low supply voltage for the power supply connected to the ECU, etc. Some external related system faults may be detected by the ECU after generating a stimulus to the vehicle (or by waiting until the ECU operating strategy generates a stimulus) and then looking for a correlated response in the vehicle. For example, in an air conditioning system if the compressor is activated, the heater air temperature should decrease after a little time. This method does not detect the fault directly, but implies it from available data.

An indirect fault is a fault that originates outside the system controlled by the relevant ECU. These faults may affect the relevant system by providing it with incorrect signals or data. Hence these faults typically include various network related faults, such as network short circuit / open circuit faults, bus-off faults, intermittent loss of

communication, faults due to invalid / implausible serial data received, etc. Although the network components (transmitter, receiver etc.) may be physically located within the ECU or system, however due to the cascading nature of these faults, they need to be treated differently and considered as external to the system.

It is very important to note that these indirect faults may affect more than one system. There are occasions when faults will be detected by more than one system. Examples of such situations could be, when components are providing an input to one system that passes data to another system, or when primary internal or external faults, which lead to further faults through cascading effect, or when network or gateway faults affect one or more systems connected in same or different sub-networks, etc. Oftentimes it is observed that, these indirect faults may lead to detections at a distance or to multiple detections of the same basic fault by multiple ECUs. Therefore this paper focuses on indirect faults and proposes methodology to avoid cascading effects of indirect faults, resulting uncorrelated spurious DTCs.

Localization and discrimination of faults

A methodology is proposed in this paper for localization and discrimination of primary faults in the distributed network architecture. The aim of the proposed methodology is to identify and filter out the primary faults of the vehicle from the population of all faults, which are getting monitored and matured by different ECUs in the vehicle.

Network architecture

To realize the proposed vehicle level fault management strategy, centralized intelligent gateway module based vehicle network architecture is proposed, wherein the centralized Gateway Module needs to be connected to various other sub-networks in the vehicle, such as HS CAN, LS CAN, LIN, FlexRay sub-networks, etc.

A typical passenger car with various ECUs from power-train, chassis, body & security, comfort and infotainment domains is considered to explain the proposed network architecture and the proposed vehicle level fault management strategy in detail. It is also assumed that few of those ECUs such as IBS, IA, RLS, VSCM and WCM support LIN interface for network communication and other ECUs support HS CAN interface for network communication.

Refer [Figure 1](#), which depicts the network architecture as per the proposed method, where BCM is acting as the centralized Gateway Module in this architecture, which is not only acting as CAN communication gateway, but also acting as diagnostic gateway and LIN Master for two clusters of LIN nodes. In order to achieve cost benefits, it is recommended not to use any dedicated Gateway Module; instead it is recommended that BCM will play the role of centralized Gateway Module. There is also a significant paradigm shift in this proposed network architecture, compared to existing network architectures in existing vehicles. Apart from the centralized Gateway Module there is

no other ECU in this topology, which is required to support gateway functionality either in terms of CAN to CAN communication gateway or CAN to LIN communication gateway or diagnostic gateway. This leads to the fact that there exists no other Gateway Module in between the centralized Gateway Module (i.e. BCM) and any other ECU in this architecture. As this centralized Gateway Module is responsible for fault management of the complete vehicle, it is very important that the fault status information from various ECUs in the sub-networks reach the Gateway Module as quickly as possible, thereby enabling the Gateway Module to decide on the primary faults quickly. Therefore it will be ensured that there is no other delay introduced by any other Gateway Module between any ECU and the centralized Gateway Module. Hence, the network architecture as shown in [Figure 1](#) is proposed, such that there is no other intermediate Gateway Module between any ECU in the architecture and the centralized Gateway Module.

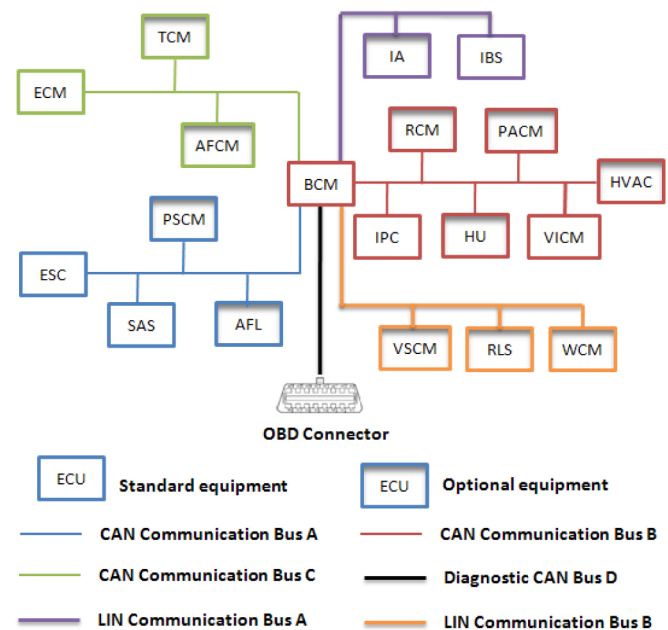


Figure 1. Network Architecture with centralized intelligent gateway

Diagnostic communication

In the next step, the on-board diagnostic communication between different ECUs in the network architecture and centralized Gateway Module (BCM as per [Figure 1](#)) needs to be defined. Typically diagnostic communication is required only when an external diagnostic tester tool (i.e. client) is connected to the vehicle and requesting for responses from any ECU (i.e. server) in the vehicle, which is typically off-board mode of operation and event driven communication. However, in this paper on-board and periodic diagnostic communication is proposed between different ECUs in the network and the centralized gateway module during vehicle's operation. Each ECU in each sub-network needs to

communicate the information of various faults along with the status of each fault which are in the process of maturation or de-maturation by the respective ECUs as outcome of running various tests / monitors locally, through diagnostic message to the centralized Gateway Module.

For each fault monitored by each ECU locally, one Fault Detection Counter (FDC) will be supported by the ECU locally. The FDC is a single-byte (signed) counter, that is used to determine when the fault is matured. The value of DTC Status Byte associated with each DTC, as specified in UDS on CAN Protocol specification (ISO 14229-1), is determined based on the value of FDC.

Faults may be given calibrated values in relation to their importance. Each occurrence of testResult being failed, will increment the FDC by the calibration value, and each occurrence of testResult being passed will decrement the FDC by the calibration value.

testFailed bit (bit “0” in the DTC Status Byte defined in ISO 14229-1) will be set to 1 when the FDC reaches $7F^h$ and set to 0 when the FDC reaches 80^h . If the FDC is between these two values, testFailed will retain its setting during the operation cycle. testFailed bit will be set to 0 between operation cycles. The fault does not lead to action until the FDC reaches $7F^h$ and testFailed bit is set to 1. If testResult is failed while the FDC is negative (i.e. between 80^h and FF^h), the counter will be reset to 00^h before being incremented by the calibration value. Note that the FDC does not “wrap round”. If an increment would otherwise take the counter beyond $7F^h$ (+127) or 80^h (–128), the counter will stop at the limiting value.

The structure of the diagnostic message is proposed. For each fault monitored by each ECU locally, the respective ECU will transmit one 2 bit status signal to the Gateway Module. [Table 1](#) specifies the definition of the diagnostic status signal for each fault monitored by the ECU locally. As one CAN Frame supports 8 Bytes of payload, it will be possible to accommodate status information of 32 different local faults monitored by each ECU in one diagnostic message. If the ECU monitors more faults, then number of such diagnostic messages transmitted by the ECU need to be increased. As each ECU monitors its own local faults, the increment or decrement of FDC as outcome of each Test run, will be managed by each ECU locally. Therefore, it is sufficient to convey the minimum fault status information, as specified in [Table 1](#), to the Gateway Module, thereby optimizing the total number of diagnostic messages, which needs to be transmitted by each ECU in the network. [Table 1](#) also specifies the condition to transmit a particular signal value of Diagnostic Status Signal for a fault with reference to DTC Status Byte definition as specified in UDS on CAN Protocol specification (ISO 14229-1). If any other diagnostic protocol is implemented in the vehicle, then similar conditions to transmit a particular fault status need to be derived.

Table 1. Definition of Diagnostic Status Signal

Bit value (in Binary)	Signal value of Diagnostic Status Signal	Condition of DTC Status Byte as specified in UDS on CAN Protocol
00	Fault is currently not present	testFailed (bit 0 in DTC Status Byte) = 0 AND testNotCompletedThisOperationCycle (bit 6 in DTC Status Byte) = 0
01	Fault maturation in progress	testFailed (bit 0 in DTC Status Byte) = 0 AND $-128 < FDC < +127$
10	Fault de-maturation in progress	testFailed (bit 0 in DTC Status Byte) = 1 AND $-128 < FDC < +127$
11	Fault is	testFailed (bit 0 in DTC Status Byte) =
Bit value (in Binary)	Signal value of Diagnostic Status Signal	Condition of DTC Status Byte as specified in UDS on CAN Protocol
	matured and present	1

The priority of such periodic diagnostic messages will be lower than the regular functional messages in the network, but higher than event driven diagnostic messages, which are triggered by external diagnostic tester tool. Hence, as usually CAN IDs from 700 Hex to 7FF Hex are used for diagnostic communication between external diagnostic tester and ECUs in the vehicle, it is recommended to use CAN IDs from 600 Hex to 6FF Hex for the proposed periodic diagnostic communication between each ECU and the Gateway Module. The cycle time of these periodic diagnostic messages will be 50 to 100 ms.

Identification and filtering of primary faults

The objective of this on-board, periodic diagnostic communication is to make all the fault information of the vehicle available to a centralized ECU, who has the global view of the complete vehicle's health status and does the classification of the faults based on spatial and temporal information associated with received fault information. Gateway Module processes the received fault information from various ECUs to localize the primary fault and store the appropriate fault record corresponding to the localized primary fault only in its long-term memory. The spatial information is derived from the fact that the Gateway Module knows which particular ECU is broadcasting this particular diagnostic message from which particular sub-network. Temporal information is not explicitly included in the diagnostic message; however, the periodic transmission of diagnostic messages by respective ECUs implicitly encodes