

Automotive ECUs Fault Diagnosis Modeling Based on the Fault Database

Yanqiang Li, Yang Li, Zhixue Wang, Ruke Zhuang and Jianxin Li

Abstract The automotive ECUs is becoming more and more complicated, and so is the fault diagnosis. In order to improve the maintenance quality and efficiency, the paper proposes a fault diagnosis approach based on fault database. By making full use of data stream, we firstly extract symptom vector by processing data steam and pre-processing rules, and then we use the symptom vector to match the fault pattern in fault database, we use the unmatched vector as the test case of C4.5 decision tree algorithm to create the link rules between fault symptom and fault reason, and finally store the rules into the fault database. An example of ETCs is showed to testify the fault diagnosis method. The test result confirm the reliability and validity of this diagnosis method.

Keywords Fault diagnosis modeling · Data pre-processing · Decision tree · Fault database

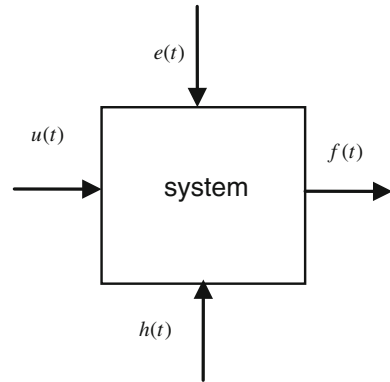
1 Introduction

With the rapid development of automotive Electronic Control Unit (ECUs), the fault diagnosis becomes more and more complicated. And the link between fault and fault symptom becomes less obvious. Diagnostic Trouble Code (DTC) and in-vehicle fast data snap-shop is very useful to diagnose an ECU, but the fault

F2012-D02-026

Y. Li (✉) · Y. Li · Z. Wang · R. Zhuang · J. Li
Shandong Provincial Key Laboratory of Automotive Electronic Technology,
Institute of Automation Shandong Academy of Sciences, Jinan, Shan dong, China
e-mail: lisim51@126.com

Fig. 1 To-be-diagnosed system



coverage of DTC is limited. For example, the DTCs of 2003 Accord only cover 37 % malfunctions. Additionally, sometime the faults indicated by DTCs cannot demonstrate the actual fault [1]. Although more and more diagnosis technology have been applied in this field, but those methods, such as [2–4], often boils down to the use of heuristics, associative case memories, or expert systems. These approaches are restricted with respect to the complexity of the diagnosed system and the faults to be detected [5].

This paper proposes a fault diagnosis approach based on fault database technologies in contrast to the above traditional approaches. Firstly, abnormal data stream is extracted from data stream and then a fault symptom vector is created by utilizing the data preprocessing approach, then we put the symptom vector into the fault database to match fault pattern, and we use the unmatched symptom vector as test case of decision tree to set up connection between fault symptoms and its failure reasons, and at last, we save to the fault database.

2 Description

Generally speaking, fault diagnosis modeling is a complex process. The to-be-diagnosed object can be abstracted as a system [6]. The system contains its hardware and software, and it has interaction with outside environment. The outside environment contains human factor, EMC factor, working environment factor etc. The system fault reason contains hardware fault, software fault or mechanical fault etc. The system can be abstracted as follows Fig. 1. The system has input information or input functions $u(t)$, such as driving command or data from sensors; the output information contains actuator command and fault information. In addition, the system has the other possible failure caused by environment factor $e(t)$ and human factor $h(t)$.

Because diagnosis object structure is complex under the condition of many interaction factors, it is difficult to implement the fault diagnosis. But through the

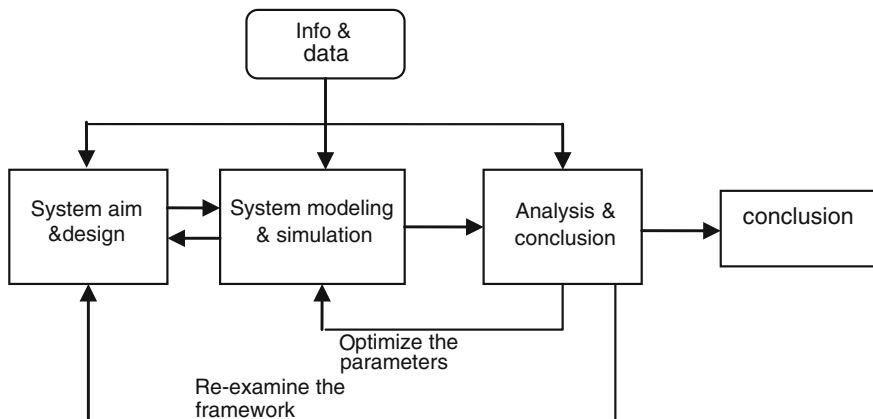


Fig. 2 Framework of Fault Diagnosis

establishment of a reasonable framework and idea of fault diagnosis, it is not complicated to solve this problem. The paper proposes fault diagnosis structure as shown in Fig. 2.

Firstly to clear the operating mechanism and fault symptom of whole system based on available theoretical knowledge and experience, and then to determine the system framework and boundaries and parameters of the variable for solving the problem.

Secondly to built the system modeling based on the first work. The input–output relationship should be described based on the mathematical and logic models. Based on those models, simulation should be implemented in order to adjust and optimize the parameters of those models.

Thirdly to optimize the previous parameters and boundary parameters based on the history data and simulation result several times.

Fourthly to re-examine the previous model framework and boundary definition based on the quantitative analysis several times in order to satisfy accurate requirements.

At last to store the modeling conclusion.

3 Diagnosis Modeling

3.1 Electrical Throttle Control-System

In this paper, we propose an example of Electrical Throttle Control-system (ETCs) which is the system that allow the ECM to precisely control the opening and closing of the throttle valve based on drivers input and is also interrelated with Traction Control ECU and Vehicle Stability Control ECU. The topology of the ETCs is as follows:

The ETCs-I is composed of an accelerator pedal assembly, a throttle body assembly, and an ECM. The ECM contains CPU, throttle motor control drive circuitry, a power supply, and inputs from other functions. In addition, the ECM electronic fuel injection and ignition functions provide fuel and spark in the correct amounts and at the correct time to keep the engine running. All three (i.e., air mass, fuel, and ignition) are needed in the correct proportion and sequence for the engine to run otherwise power output is diminished and/or the engine stalls [7].

3.2 *Diagnosis Modeling*

Fault diagnosis modeling is the core of the diagnosis. The paper proposes such the diagnosis process: picking up abnormal data flow from data stream and pre-process those data to extract fault symptom vectors based on the data pre-processing rules, putting vectors into the history fault database to match the same fault mode, those unmatched vectors are deemed as test case for decision tree, the decision tree creates diagnosis rules, and the rules are stored in the fault database. The working process is as follows [4]:

1. Firstly, we need to know the working principles of ETCs, and to analysis the possible fault reasons. The possible fault component concludes: Throttle position Sensor, Idle speed Sensor, Accelerator pedal sensor, Transmission gear Sensor, Throttle Motor, Cruise control System, Engine control System.
2. Secondly, we need to classify and check the data stream: the data stream is mainly divided into value information, control information and the other information. The premise of fault diagnosis is to judge whether the parameters are in a reasonable range, are stable and are change unreasonable.
3. Thirdly, we need to pre-process those abnormal data stream. The aim of data pre-processing is to extract useful information, and change data stream into symptom vector based on pre-processing rules, and the vector is the input of the fault database;
4. Finally, putting the symptom vectors into the history fault database which is the core component of fault diagnosis. The fault database is used to store fault symptom vector and its corresponding fault reason vector. If the input vector can match the fault mode, the fault reason can be found based on the fault reason vector; if it isn't matched, the fault symptom vector is used as test case of decision tree. Though decision tree algorithm, the fault reason rule is created. With the accumulation of test cases, the fault database will become more effective. Figure 3 shows the fault diagnosis process (Fig. 4).

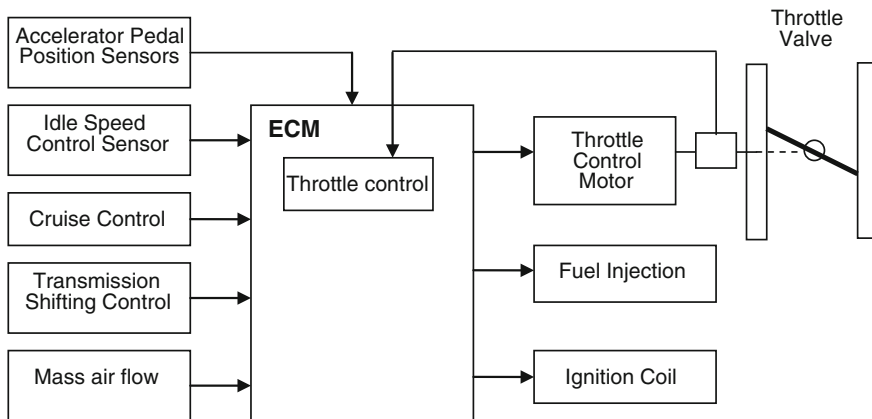


Fig. 3 Topology of ETCs-I

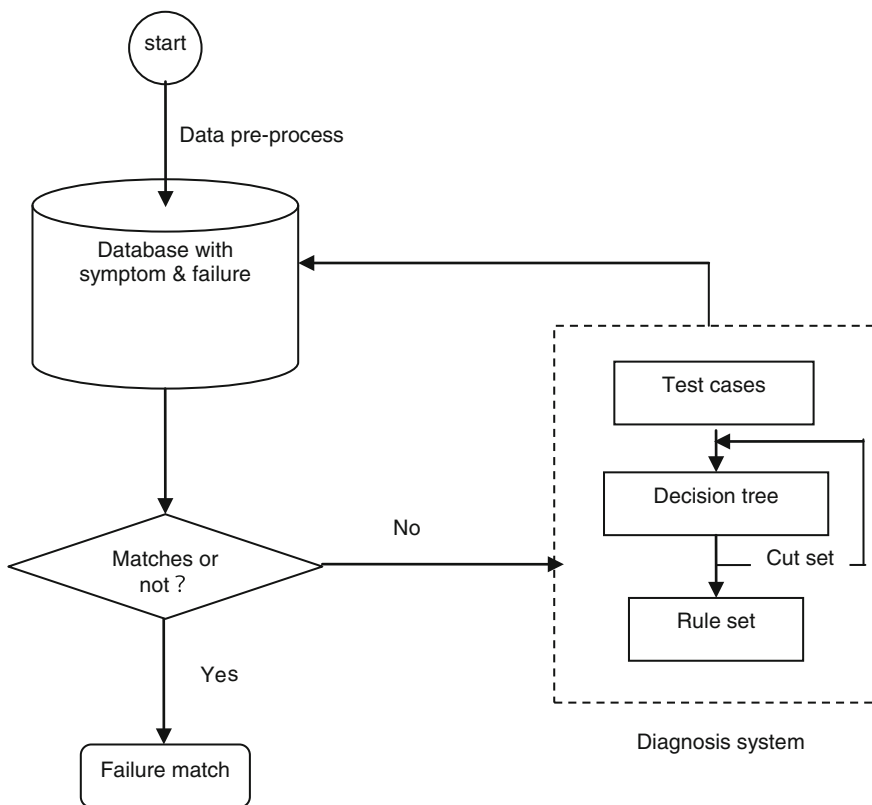


Fig. 4 Fault Diagnosis Process

3.2.1 Data Pre-processing

Based on the working principle of ETCs, we should analysis the possible failure and link between the fault symptom and fault reason [8]. According to the experience of domain expert knowledge, the fault symptom and fault reason of every possible fault component are encoded, Table 1 shows the fault symptom code of ETCs.

Fault database is used to store fault symptom vector and its corresponding fault reason vector, the fault database needs to refer to a lot of experiment and test result. The expert should set the initial value based on their experiment firstly, however, the low accuracy of database should be modified constantly, thus, with the increase of the test cases, the database becomes more and more accurate and credible. Table 2 shows the relationship between fault symptom and its related reason vector.

Fault occurs when abnormal data steam is generated. After extracting those abnormal data stream, the fault symptom vector can be created based on the fault component and its symptom, according to transacting rules of Table 2.

3.2.2 Decision Tree Algorithm Modeling

The most influential decision tree algorithm is ID3 and C4.5 proposed by Quinlan in 1986. The core algorithm of ID3 is to take all unused attributes and count their entropy concerning test samples, to choose attribute for which entropy is minimum and make node containing that attribute. The C4.5 is the improved ID3 algorithm, using information gain ratio as the attribute selection criteria, to make up for the inadequacy of ID3 algorithm [9, 10]. The working process is as follows:

1. Calculate the exception information of set S: Suppose S is the data set of s subset, suppose s_1, s_2, \dots, s_m are m subsets belonging to E, the information entropy is as follows:

$$I(s_1, \dots, s_m) = - \sum_{i=1}^m \frac{s_i}{s} \cdot \log_2 \frac{s_i}{s} \quad (1)$$

2. Calculate the entropy of the subset S after a split over the A attribute:

$$E(A) = - \sum_{i=1}^k (s_{ij} + \dots + s_{mj})/s \times I((s_{ij} + \dots + s_{mj})). \quad (2)$$

Table 1 Data Pre-processing

Electronics component	Identifiers	Failure symptom	Possible failure reason
Throttle Position Sensor (TPS)	f_1	Small openings $<3-5^\circ$ between normal sensor values and DTC for sensors not within the range.	Position sensor fail high, low, intermediate values. compromised power feed or ground to both sensors
	f_2	Small openings $<3^\circ$ between normal sensor values and DTC position does not match commanded	Incorrect learning of fully closed position, sensor voltage lower
Idle speed sensors	f_3	DTC for sensors not within operational zone	Temperature sensor fail high, low, intermediate values
	f_4	DTC for sensors not within operational zone, Engine stall	Incorrect Sensing of Engine Load, Gear, Power Steering, A/C, Electric Load, Engine Speed
	f_5	Small openings $<3-5^\circ$ between normal sensor values and DTC limit	Incorrect learning of throttle angle for Idle speed
Transmission sensors	f_6	DTC when selected Gear doesn't match sensed gear ratio	Transmission shifting, torque converter lockup throttle modulation
Pedal sensors	f_7	DTC for high, low and outside lane. None, if Pedal sensors fail within operational lane DTC	Position sensor fail high, low, intermediate values
	f_8	Engineered Fault in operational lane Valid pedal signal escapes detection, not DTC. Electrical Failures should leave trace	Incorrect learning value, Dual failures to specific voltages that result in voltages within operational lane
	f_9	Dual failures that result in voltage within operational lane	Engineered Fault in operational lane Valid pedal signal escapes detection, no DTC.
Throttle motor	f_{10}	DTC high current, transitory small opening until DTC activates fail-safe mode <0.5 s	Motor shorted to power or ground, h-bridge fail, lath-up or transistor short
Cruise control	f_{11}	Short of power, ground, open disable function	Failure of Steering Column Switches or ECM input circuit
	f_{12}	DTC when both Brake Switch poles do not agree. Cannot switch into Gear	Failed Brake Switch
	f_{13}	DTC for Speed Sensor, small throttle openings to maintain 0.06 g acceleration to set speed	Vehicle Speed indicate lower speed

(continued)

Table 1 (continued)

Electronics component	Identifiers	Failure symptom	Possible failure reason
Software of ECU	f_{14}	Theoretical Fault Escapes Detection	Software unilaterally opens throttle with Accelerator released., Idle Fuel Cut not active, Watchdog serviced, no EDAC error

Table 2 Fault symptom vector

Num	Fault symptom vector					Fault reason vector (U)
	f_1	f_2	f_3	...	f_{14}	
1	1	0	0	...	0	U_1
2	0	1	0	...	0	U_2
3	0	0	1	...	0	U_3
...
n	0	0	0	...	1	U_n

3. Calculate the gain of the attribute A is:

$$Gain(A) = I(s_1, s_2, \dots, s_m) - E(A) \tag{3}$$

4. Calculate the gain of the subset S after a split over the A attribute:

$$Gain\ Ratio(A, S) = Gain(S, A) / SplitInfo(S, A) \tag{4}$$

$$SplitInfo(A, S) = - \sum_{i=1}^c (|S_i|/|S| \times \log_2 |S_i|/|S|) \tag{5}$$

SplitInfo (S,A) expressing the breadth and uniformity of split set S according to attribute A.

The highest attribute of the information gain ratio as the test attributes of set S is used to create a node. To create branch according to A property of all value or all intervals, so to divide the sample.

5. Cut Sets- the formal method are pre-pruning and post-pruning, and post-pruning allows over fitting data, and then built the tree pruning; The pre-pruning method is difficult to estimate when to stop tree growth accurately, so the post-pruning method is more practical in the real problem. The paper use the post-pruning method to avoid the tree over growing and to avoid data over fitted, and using the test case itself to judging whether pruning or not.

$$\Pr \left[\frac{f - q}{\sqrt{q(1 - q)/N}} > z \right] = c \quad (6)$$

N = quantity of test cases, $f = E/N$ error rate (E is the N instances in the number of classification errors), q = real error rate, c = degree of confidence (C4.5 default value is 2.5), z is the corresponding standard deviation of the degree of confidence c .

Suppose there 14 fault according to the current test samples. The fault reason is as follows: U_1 = sensors shorted to the ground or power, U_2 = Sensor supply voltage is too low resulting in the position value error, U_3 = temperature value not within the valid range; U_4 = Error sensing data leading to data collection error; U_5 = incorrect learning of throttle opening, U_6 = Torque conversion error; U_7 = position sensor's voltage lower or higher, U_8 = Position sensor output error learning, U_9 = Fault coverage for non-DTC torque, U_{10} = Electrical shorted to power or ground, H-bridge failure, U_{11} = steering wheel key failure or ECM input circuit failure, U_{12} = Brake switch failure, U_{13} = Speed error, U_{14} = watchdog action or login errors.

According to the formula, we need to calculate exception information of all test cases; Then, calculating the information gain ratio of each attribute. And constructing every brunch of decision tree according to the formula above. After traversing the decision tree we get the following classification rules:

IF throttle opening increases AND DTC occurs for throttle sensor not within the valid range THEN throttle sensor shorted to Power or Ground;

IF throttle opening increases AND DTC occurs for pedal sensor not within the valid range THEN position sensor's voltage lower or higher;

IF throttle opening increases AND cruise control system shorted to power or ground THEN steering wheel key failure or ECM input circuit failure;

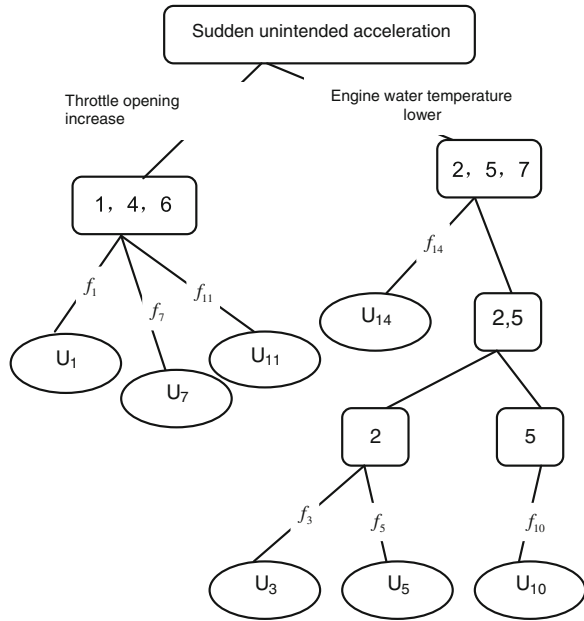
IF engine temperature is too low AND ECM fault escape detection THEN watchdog action or login errors;

IF engine temperature is too low AND pedal sensor opening small openings $<3-5^\circ$ between normal sensor values THEN temperature value not within the valid range;

IF engine temperature is too low AND pedal sensor has no DTC and throttle small opening <3 to 5° THEN incorrect learning of throttle opening;

IF engine temperature is too low AND throttle motor small-angle instantaneous detecting large current DTC THEN Electrical shorted to power or ground, H-bridge failure (Fig. 5).

Fig. 5 Fault Decision Tree



3.3 Experiment Result

Fault phenomena of one type electronic throttle system is that throttle opening increases unexpected without any operation. After monitoring signals by using CANoe, we find the abnormal data stream comes from pedal sensor. Based on the data-preprocessing rules, we get the fault symptom vector $f = [0, 0, 0, 0, 0, 0, 1, \dots, 0]$. After put the vector into the fault database, the fault pattern is matched and the fault reason is the output of position sensor is too high or too low. The fault symptom disappeared after changing the position sensor.

4 Conclusion

This paper proposes a fault diagnosis modeling structure in the first place, then it proposes a fault diagnosis algorithm based on fault database. The main process is to extract the abnormal data stream and change the data stream into fault symptom vector based on data pre-processing rule, and then put the vectors into database to match fault pattern, and we use the unmatched symptom vector as test case of decision tree to set up connection between fault symptoms and its failure reasons. At last, we store the rules in the fault database. The paper proposes ETCs as an example to test the fault diagnosis model, and the test result confirm the reliability and validity of the modeling method.

Acknowledgments This work has been supported by Natural Science Foundation of Shandong Province, China (ZR2011FQ034).

References

1. Ya D (2009) Data analysis in automotive fault diagnosis. Machinery Industry Press, Beijing, pp 2–4
2. Yi J (2011) Research on fault diagnosis expert system of automotive engine based on ontology. *Electrical Control Eng (ICECE)*, pp 5409–5412
3. Choi K (2006) Data reduction techniques for intelligent fault diagnosis in automotive systems. *Autotestcon*, pp 66–72
4. Namburu SM (2006) Application of signal analysis and data-driven approaches to fault detection and diagnosis in automotive engines. *Systems, Man and Cybernetics*, pp 3665–3670
5. Stein B (2003) Model compilation and diagnosability of technical systems. In: Hanza MH (ed) *Proceeding of the 3rd IASTED international conference on artificial intelligence and application (AIA 03)*, BenalmAqdena, Spain, pp 191–197, ACTA Press, Sept 2003
6. Xuesen Q (2005) A new discipline of science-The study of open complex giant system and its methodology. *Urban Stud* 12(5):1–8
7. Technical support to the national highway traffic safety administration on the reported toyota motor corporation unintended acceleration investigation. 2011
8. Li Y, Li Y (2010) Fault diagnosis of automobile ECUs with data mining technologies. *Adv Sci Eng* 2010:156–161
9. Quinlan JR (1993) *C4.5: Programs for machine learning*. Morgan Kaufmann Publishers, Burlington
10. Hui O, lebin L (2010) Research of paper metadata extraction algorithm based on C4.5. *Comput Eng Des* 2010(16):3708–3711