Research on OBD Performance of T-GDI Engine

Song Yan, Pengyuan Sun and Tonghao Song

Abstract In order to fulfill requirement of Euro V regulation and meet the high reliability and performance requirements, the On-board Diagnostics (OBD) software architecture of turbo-charged GDI engine is redesigned, the analysis and improvement of OBD reliability and performance based on software structure are described in this chapter, the main factors have been taken into consideration.

Keywords GDI · EURO V · OBD

1 Introduction

According to the inspection of EPA (Environmental Protection Agency), above 60 % air pollution from passenger car was caused by the fault of the emission control system. In recent years, many techniques have been applied to reduce the pollutant emission. On-board Diagnostics technique had been introduced to vehicle for this purpose in 1989 when the first OBD regulation released by CARB (California Air Resources Board), and the similar legislation appeared in European Union [1], which is called EOBD regulation. The Euro V stage, including EOBD for SI-engine on passenger vehicles, separated into two sub-stages, named as EU5a and EU5b, or as EU5 and EU5 plus, has been implemented since 2009 (Fig. 1).

F2012-D03-027

S. Yan (✉) · P. Sun · T. Song
FAW R&D Center, Changchun, China
e-mail: yansong@rdc.faw.com.cn

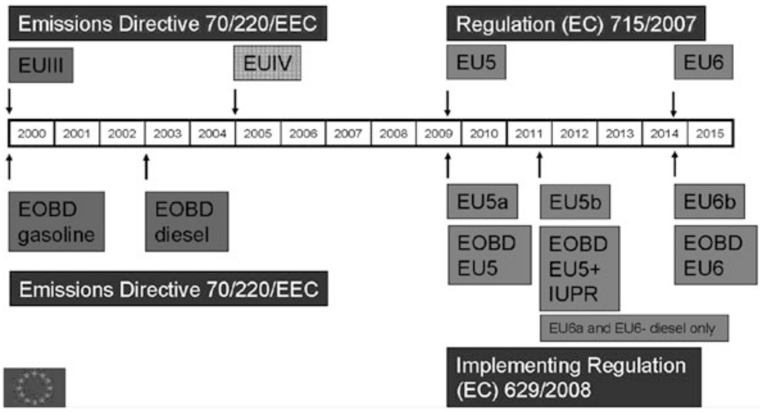


Fig. 1 Schedule of OBD implementing regulation

According to the clauses from EOBD regulation, all components or systems which might lead emissions exceed the standard limits should be monitored, and monitors should give some indications on finding any fault or deterioration of components or systems. In addition to the illumination of the MIL, the faults storage and a standardized environment datasets are required. All information should be sent to an external scan-tool by a standardized communication protocol and interface.

In recent years, the OBD software development has encountered challenges because it is experiencing a continuous increase of functionality extent and complexity. Today, in some high-end SI-engine based on Euro V regulation, the OBD related executable software codes in engine control unit is above 40 %, and the number of OBD calibration parameters exceed 15,000 because of the stricter legislation and costs saving demand. The cost reduction is mainly focused on minimized components cost. Take misfire detection strategy for example, it could be done by using cylinder pressure sensors to measure the cylinder pressure, which could be easier for the software implementation but higher cost because the cylinder pressure sensors is so expensive. It could also be done by using engine speed sensor to measure the fluctuation of engine speed, which could be more complicated for the software implementation but much cheaper because the engine speed sensor has already existed in system.

In order to fulfill the Euro V regulation and meet the high reliability and performance requirements, the OBD software of a new passenger car with turbocharged GDI engine is redesigned.

There are some researches related with OBD performance topics. But they are mainly focused on hardware design and test field. For example, in terms of hardware design, smart driver ICs are used by various loads like injector, fuel pump, O2 heater and lamps, and fed back diagnostic information to a micro chip via SPI [1], it can detect power-stage fault much reliably and decrease main chip load. In terms of test, to confirm the performance of the OBD system and to avoid any kind of misdetection in the OBD system, several field tests such as the fleet test and the environmental test have been described in the chapter [3, 4], and from the experimental results, it indicates that significant potential exists in the optimization of the real-world OBD system performance [5].

2 Reliability and Performance Analysis

The Euro V EOBD is inherited from Euro IV, most of components and systems monitors requirements are all similar except for some of special demands. For example, catalyst NOx conversion monitoring strategy is needed for Euro V EOBD.

The reliability mainly refers to regulation fulfillment and fault reports, and the performance is mainly referring to efficiency of functions execution.

Some factors, except for cost, which affect the reliability and performance are considered. In this chapter, the three factors are discussed, including optimization of OBD system management, condition interlock and efficiency of function execution.

2.1 Factor 1: Optimization of OBD System Management

In a short trip, some monitoring functions perhaps have no chance of execution, and with software continuous increase of complexity, it is possible that some functions would have no chance to run. It would not fulfill the regulation of Euro V plus. So a dynamic priority of functions and in-use performance ratio (IUPR) concept are introduced to solve the problem. The dynamic priority is conceived from operation system. Scheduler can determine the status of a function depending on the dynamic priority, and the status is comprised of sleep, ready, activated, and etc. When a function is activated, it means the function has got the priority to run. The IUPR is from EU5b stage regulation demand, it was legislated in Euro V plus, and used to monitor the diagnostic frequency of functionality by drive cycles. The dynamic priority is designed and is determined by the significance of the function and its history of IUPR records. As a monitoring function running frequency is so low in recent drive cycles that IUPR ratio value below a certain limit, the dynamic priority of the function should be increased. It can assure that all monitoring functions have chance to execute recently. All the work is implemented by function scheduler to arrange the monitoring function execution order (Fig. 2).

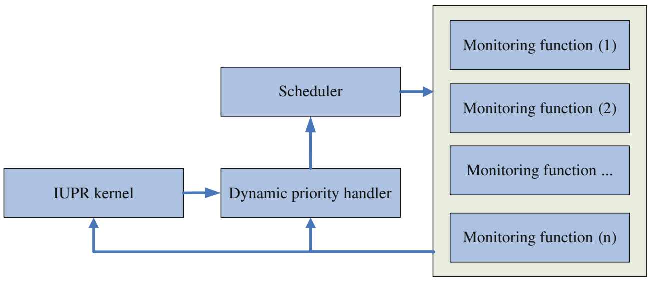
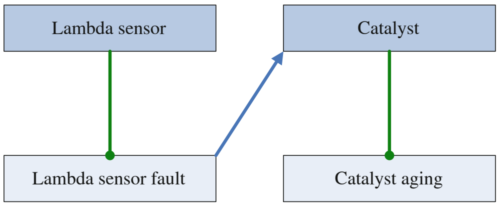


Fig. 2 Scheduling of monitoring function

Fig. 3 Example of derivative faults



2.2 *Factor 2: Condition Interlock*

To ensure the correct implementation of the functionality and to avoid incorrect results report, functions physical conditions are introduced, which are used to prevent monitoring functions from running with error. Too strict conditions would always lead monitoring functions to be locked, and too loose conditions may lead to incorrectly execution on the contrary. Both cases are not reliable. From the perspective of software development, the correct result is needed to be considered based on legal demand. It means running under correct conditions should be guaranteed from functionality consideration and should test the conditions under the NEDC cycle provided by European emission regulation.

Beside the physical conditions, there is another important condition called "derivative faults". Figure 3 shows an example of the "derivative faults". The three-way catalyst ageing monitoring strategies is based on dual lambda sensors (also called oxygen sensors), they are located in front and rear of catalyst respectively. If no consideration about "derivative faults" is taken, any defect of lambda sensors may lead to a false report of the catalyst, so both faults of lambda sensor and catalyst would be reported although catalyst has no fault actually. Therefore, the result should be regarded as incredible. To solve the "derivative faults" problems, the primary fault and secondary fault are defined to distinguish the root cause of faults. In this case, the lambda sensor fault is considered to be primary, while the catalyst fault is considered to be secondary. If the primary fault is reported, the function to report the secondary fault will be locked and no secondary fault will be reported.

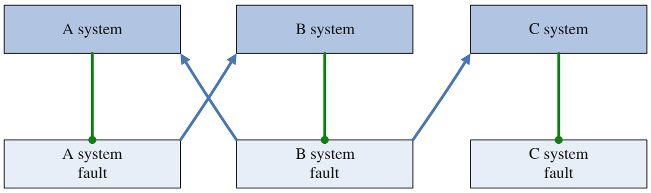


Fig. 4 Example of a more complexity of derivative faults

An even more complexity of “derivative faults” problem arises with cycles of fault primary-secondary relations (Fig. 4). “A system” fault will cause “B system” error, and “B system” error also can cause “A system” fault report. So both faults are primary faults and can not be distinguished for each other. In this example, the “C system” is secondary error to “B system”, and its status will distinguish the root cause fault: If “C system” has a fault, then “B system” is validated to be the root cause but not “A system”. Vice versa if “C system” has no fault, “A system” is the root cause. There is a simple case constituted by front lambda sensor, fuel trim system and rear lambda sensor respectively. The additional logic will make the diagnostic results more credible.

The existence of function lock is not only up to primary-secondary “derivative fault” relationship, but also up to the cross-influence of the functions. An example is introduced here in Fig. 5, the function which is pointed by a arrow will be effected by the other function, which is on the other end of the arrow. There are four monitoring functions around the lambda monitoring system. In case of rear lambda sensor monitoring function is not running, the other three functions would be affected. To prevent the influence, a test result is introduced. When a function is executed completely, its test-over flag will be reported. The flag is reported not only to other three functions which have influence on this function, but also to standardized external communication interface, like scan-tool. Only when the rear lambda sensor has been tested ok and test-over flag is reported, the other three monitoring functions will have the permission to be activated. Otherwise, they will be locked.

2.3 *Factor 3: Efficiency of Function Execution*

The efficiency here is not the code efficiency but the architecture of the program emphasized in this chapter. All functions come from regulation requirements. But when the functions are executed in CPU, the hardware load has to be taken into consideration. Most of cases, one functionality is able to be separated into several tasks to run in control unit (Fig. 6), and each task could be regarded as a sub- functionality, while the function integrity does not be affected. They could be running in deferent time slice on the the operating system. The basic principle of separation is according to sample rates of the output variables.

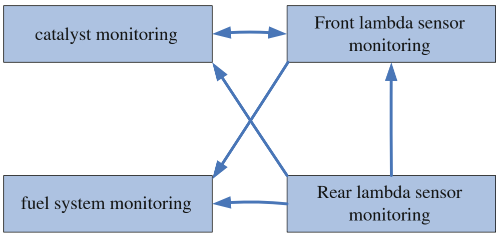


Fig. 5 An example of cross-influence of the functions

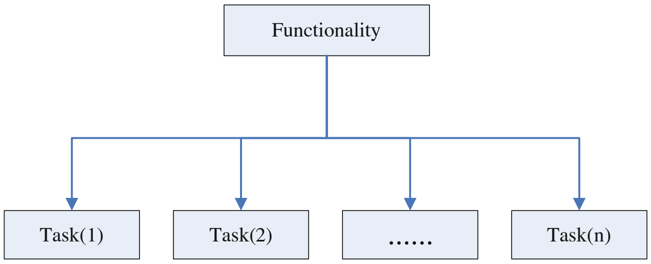


Fig. 6 Relationship between functionality and tasks

The task could be carried out in a certain time interval repeatedly, for instance, 10, 50 ms, and etc., or every ignition interrupt event, which is similar to OSEK operating system. In order to add the tasks into the operating system, it is necessary to analyze the relationship among tasks and to test CPU load under the most unfavorable case repeatedly. If a task could run in either 50 or 10 ms time interval, the 50 ms period is preferred for chip load consideration. If the task could run in either 10 ms time interval or ignition event, 10 ms period is preferred based on the same consideration as above.

Although the separation could decrease the CPU load, there is still a disadvantage that is the increasing of switch-over time between tasks as the number of tasks growing, and sometimes the total switch-over time may be considerable. To reduce the time increase, merging the tasks which have the same time interval is very useful.

Another issue around the functionality running efficiency is distinguishing between inline and offline running mode. The off-line are generally referring to non real-time running mode, like the functions activated by external tester request, while the in-line referring to real-time running mode. Take the tooth error learning function for example, the function is used to correct the crankshaft tooth wheel error, the error may have a significant impact on misfire diagnosis especially when the engine speed is too high. The tooth error learning function can running in either on-line or off-line mode while it has little significant impact on reliability. When the function is running in in-line mode, the tooth error learning function will run every time the engine is under overrun fuel cut-off condition. If the off-line mode is selected, the tooth error learning function will be activated only by the external tester request when the engine is under overrun fuel cut-off condition. Generally speaking, offline mode is priority to online mode from the viewpoint of efficiency, but it is more reliable if the online mode is selected.

3 Conclusion

The main goals of the research are improving the reliability and performance of OBD system to fulfill regulation of Euro V and meeting continuous increasing of software complexity. To improve the reliability and performance of software, three factors are taken care specially:

- Optimization of OBD system management
- Conditions interlock
- Efficiency of function execution.

Optimization of OBD system management has introduced the dynamic priority and in-use performance ratio (IUPR) concept; they are used by OBD monitoring function scheduler to improve OBD reliability and performance.

Conditions interlock is focused on "derivative faults" and the analysis of functions cross-influence.

Efficiency of function execution contains the analysis of task divisions and the differences between online function and offline function.

Acknowledgments The authors would like to thank Jilin University who supplies engine test bench to us for validating the functionalities.

References

1. Official Journal of the European Union (2007) Regulation (EC) no 715/2007 of the European parliament and of the council[S]
2. Xie H, Hu C, Nenghui Z, Hao M (2006) A Micro-controller based control unit for motorcycle engines to meet emission and OBD requirements[J]. SAE paper no. 2006-01-0402
3. Park S, Chung Y, Park J (1998) The OBD-II system in the hyundai accent[J]. SAE paper no. 982551
4. Unger A, Smith K (1993) The OBD-II system in the volvo 850 turbo[J]. SAE paper no. 932665
5. Tsinoglou DN, Samaras ZC (2009) Malfunctions in selected emissions-related components of Euro 4 passenger cars: emissions increase and OBD system response[J]. SAE paper no. 2009-01-0731