Published 03/28/2017
Copyright © 2017 SAE International
doi:10.4271/2017-01-0599
saepcmech.saejournals.org

Dynamic Misfire Threshold Determination Based On Zone-Level and Buffer-Level Adaptations for Internal Combustion Engines

Yichao Guo
Ford Motor Company

ABSTRACT

Misfire is generally defined as be no or partial combustion during the power stroke of internal combustion engine. Because a misfired engine will dramatically increase the exhaust emission and potentially cause permanent damage to the catalytic converters, California Air Resources Board (CARB), as well as most of other countries' on-board diagnostic regulations mandates the detection of misfire. Currently almost all the OEMs utilize crankshaft position sensors as the main input to their misfire detection algorithm. The detailed detection approaches vary among different manufacturers. For example, some chooses the crankshaft angular velocity calculated from the raw output of the crankshaft positon sensor as the measurement to distinguish misfires from normal firing events, while others use crankshaft angular acceleration or the associated torque index derived from the crankshaft position sensor readings as the measurement of misfire detection. Regardless which measurement is chosen, an optimized threshold setting that clearly separates misfiring and normal firing cylinder events is desired. Traditionally, the threshold setting is determined during pre-launch calibration process with test data obtained from development vehicles. The threshold for each speed-load zone is typically a fixed value by looking up of a 3D calibration table. Such threshold setting approach, however, faces more and more challenges from the application of new engine technologies, because the crankshaft position sensor outputs could gradually change over the vehicle's life cycle, and could even change swiftly under stable engine speed and load conditions due to the application of certain new technologies. A fixed misfire threshold setting will yield false detections and result in regulation compliance concerns as well as unsatisfactory of the customers. To address this issue, this article propose a unique approach to dynamically determine the misfire threshold with zone-level adaptation to adjust thresholds for vehicle life-cycle changes and buffer-level adaption to address the needs to quickly adjust misfire thresholds in the same speed-load zone.

CITATION: Guo, Y., "Dynamic Misfire Threshold Determination Based On Zone-Level and Buffer-Level Adaptations for Internal Combustion Engines," *SAE Int. J. Passeng. Cars - Mech. Syst.* 10(2):2017, doi:10.4271/2017-01-0599.

INTRODUCTION

According to California Air Resources Board (CARB), "Engine Misfire" refers to lack of combustion in the cylinder due to absence of spark, poor fuel metering, poor combustion, or any other cause, and the combustion events in non-active cylinders due to default fuel shut-off or cylinder deactivation strategies are excluded [1]. Because the exhaust gas temperature will be dramatically increase by the unburned hydrocarbon from engine misfires, engine hardware, especially hardware in the exhaust system, will be potentially damaged. Therefore, the emission regulations worldwide mandate the detection of engine misfire and redeem it as a key on-board diagnostic (OBD) function [2].

To meet the stringent detection requirements on misfire, different approaches are developed. Some uses the instantaneous engine exhaust gas pressure [3]. Some other researches proposed to utilize the ionization signal in the combustion chamber [4, 5]. Some even developed the methods to use the model-based or sensor-based combustion chamber pressure to detect engine misfire [6]. Currently the most common way that is used by almost all the OEMs is to utilize the crankshaft timing wheel and the associated crankshaft position sensor to obtain the angular velocity or acceleration of the engine crankshaft due to the low-hardware-cost and high-hardware-reliability advantages [7, 8, 9]. With the expectation of fluctuations in crankshaft angular velocity or acceleration caused by misfired cylinders, misfiring cylinder events are separated from normal firing cylinder events by applying different filters and various signal processing methods to the raw crankshaft position sensor outputs.

Although different OEMs take different approaches to separate misfiring cylinder events from the normal firing ones, all of these methods calculate an index (i.e., the misfire detection index) based on the crankshaft angular speed or acceleration, and require an optimized threshold setting for the calculated index that clearly identifies all the misfiring cylinder events with the detection rate that meets the regulation requirements. Two major approaches are adopted by OEMs to locate such a threshold for misfire detection: the after-processing approaches and the real-time processing methods [7, 11, 12, 13, 14, 15, 16].

The post processing approaches refer to the methods that rely on the data collection on development vehicles to calibrate and preset misfire thresholds for prescribed engine operating speed-load zones [7, 14, 15]. The misfire thresholds from the post processing method are fixed values for a specific engine operating speed-load zone such that the threshold values are lack of the ability to address the changes in crankshaft angular speed/acceleration that is NOT caused by misfire (for example, the vehicle-life cycle changes, the noisy signal from the sensors, the deterioration of sensors, and the application of new control algorithms / hardware).

The real-time processing methods are the ones that calculate or determine the misfire thresholds in real time, and have the capability to adjust the thresholds based on the past crankshaft position sensor outputs to achieve a better separation between the misfiring cylinder events from the normal firing ones [11]. Comparing with the post processing approaches, the real-time processing methods have the advantages of saving the engineering cost of post-processing calibrations, and adding the capability of addressing some of the changes in crankshaft angular speed/acceleration that is NOT caused by misfire. Therefore, the real-time processing methods are preferred by the industry. The studies and applications of real-time processing misfire threshold determination methods, however, fall far behind the development of misfire detection methods. The concept of real-time dynamic threshold for misfire detection was first proposed by Zhijian Wu and Anson Lee in 1999 [11], but its application is limited to the real-time processing methods with normalized misfire detection indices. It cannot handle the misfire detection index that varies from one speed-load zone to another, has challenges to normalize the detection index across speed-load zones and to determine a threshold when some of normal firing detection indices are mixed with misfiring detection indices, and has difficulties to balance long-term vehicle life-cycle changes and short-term index noises due to hardware or control algorithms. Masanobu Ohsaki and Hisanori Ozaki later proposed a mixed method utilizing the average correlation between the detection index and the calculated threshold. Their method first calculate a threshold based on said engine operating conditions, then calculate an average correlation between the diagnosis data and the threshold, and finally determine whether the misfire judgment is valid [13]. Although the usage of average correlation helps the reduction of false detection, their method cannot resolve the false detection issues caused by sensor deterioration, vehicle life-cycle change, and the sudden noise-level changes in the detection index caused by controls or hardware.

It is, therefore, desirable to have a dynamic threshold based on realtime crankshaft data processing that is not only able to gradually adjust the threshold settings with the vehicle lift cycle, but also capable to quickly adjust the threshold settings when there are sudden changes in the detection indices. The method proposed in this study meets the above needs by utilizing the zone-level adaptations to adjust for the long-term slow changes (such as the vehicle life-cycle changes), and the buffer-level adaptations to capture all short-term variations. In addition, the proposed method is independent of the normalization of the detection index because the division of speed-load zones is based on the characteristics of the associated detection index (such as the mean value) and the specific crossing zone algorithm design ensures the smooth transition of thresholds in different zones.

This paper first introduces the fundamental principle to determine the misfire threshold. Then it investigates some popular misfire threshold determination methods, and points out their shortcomings. The dynamic misfire threshold determination method that is based on zone-level and buffer-level adaptations is finally introduced in detail with simulation results to demonstrate the advantages of this method. Summaries and conclusion are at the end.

FUNDAMENTAL PRINCIPLE TO DETERMINE MISFIRE THRESHOLD POPULAR AND DESIGN FOR SIX SIGMA

Regardless which method is used for misfire threshold determination, the fundamental principle of threshold determination should be followed. That is, the threshold should clearly separate normal firing and misfiring events and minimize the false detection ratio. Regulations, such as CARB [1], have very strict requirements on the detection ratio for misfire.

Figure 1 demonstrated an example of the distribution of misfire detection index for a four- cylinder engine. The normal firing detection indices are distributed in a band shape, and so are the misfiring detection indices. This naturally allows the application of the principles of design for six sigma (DFSS) [17] to minimize the false detection ratio.

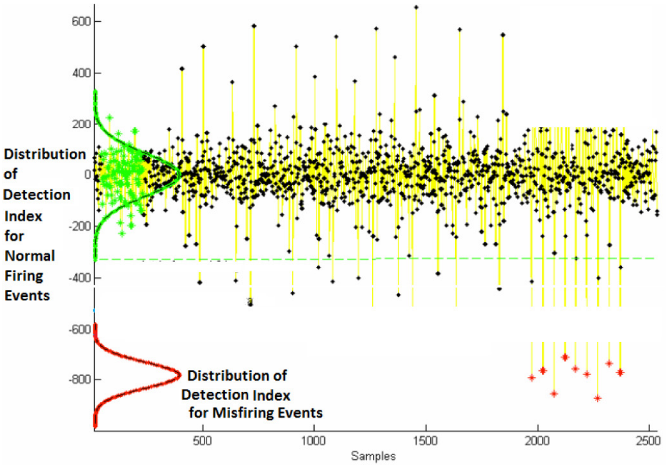

Figure 1. An example of misfire threshold determination for after-processing methods. Note: the vertical axis represents the values for detection indices, and the horizontal axis represents the probability of the detection indices to be close to a certain value.

In most of the engine operating conditions, the detection index follows the above distribution shape. This allows an easy application of DFSS principles to the setting of thresholds. If the threshold is set anywhere inside the shaded area in Figure 2, the detection ratio for misfire will be at least 99.99966%, and such threshold setting will satisfy any regulation requirements in the world. Most OEMs follow the DSFF principles to determine the misfire thresholds.

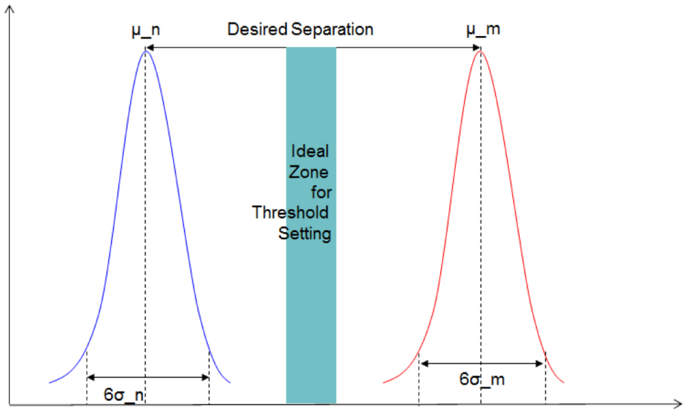

Figure 2. Desire zone for misfire threshold settings where μ_n is the mean of normal firing events, 6σ_n indicates the six sigma zone for normal firing events, μ _m is the mean of misfiring events, and 6 σ _m indicates the six sigma zone for misfiring events. Note: the horizontal axis represents the values for detection indices, and the vertical axis represents the probability of the detection indices to be close to a certain value (or the number of detection index samples per unit detection index zone).

MISFIRE THRESHOLD DETERMINATION METHODS AND THEIR SHORTCOMINGS

As mentioned in the introduction section, all the misfire threshold determination methods fall into two categories: post processing and real-time processing.

Post Processing Methods

All post processing methods are similar. They rely on the data collected from the development vehicles, post-processing of the data yields the zone division, the worse cases of normal firing and misfiring events, the associated distributions, and the best areas to set thresholds. For example, if Figure 1 is the test data collected from a development vehicle in the zone of engine speed [1000 RPM , 1500 RPM] and load [10%, 15%]. For that specific zone, the threshold could be set inside the blue band as show in Figure 3.

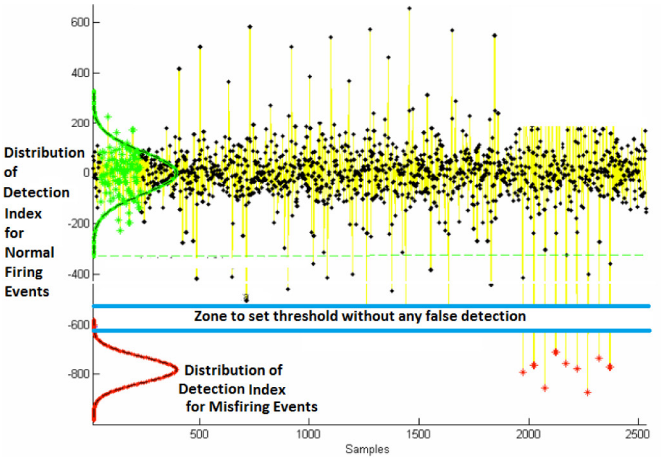

Figure 3. An example for misfire threshold determination for post-processing methods. Note: the vertical axis represents the values for detection indices, and the horizontal axis represents the probability of the detection indices to be close to a certain value

The post processing methods work well for vehicles has little changes in detection indices over its life cycle, as long as the worse cases are considered during the post-processing calibration process. Once the threshold is set, it is generally fixed (there could be some fine-turning on the threshold for specific operating conditions by multiplying a coefficient to the threshold or adding/subtracting a factor). Thus, if the distribution changes in that zone (for example, due to crankshaft sensor deterioration), false detection will happen. This is the biggest shortcoming for all post-processing methods.

For example, Figure 4 illustrates one such case of false detections. The distributions of normal firing and misfire detection indices (in solid blue and red lines) are shifted to the dashed blue and red lines. The original ideal zone for setting the misfire threshold is no longer safe, because there will be high probability for normal firing cylinder events to be falsely detected as misfiring cylinder events if the threshold remain the same in the shaded zone in blue color.

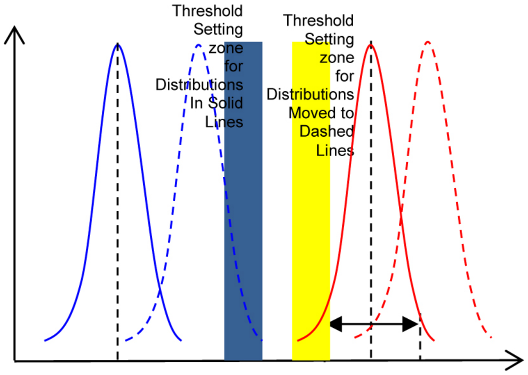

Figure 4. An example of false detections due to shifted distributions of detection indices where the blue curves represent the normal firing distribution and the red curves represent the misfiring distribution (solid curve- original distribution, dashed curve- shifted distribution). Note: the horizontal axis represents the values for detection indices, and the vertical axis represents the probability of the detection indices to be close to a certain value

Another disadvantage for post processing methods is that the test data used to develop the misfire threshold is typically collected from development vehicles. The sample size is usually small. Even if the worst cases are considered by the calibration engineers, false detection could happen due to the non- representative test data from the small sample of test vehicles. To improve the detection ratio and the robustness of threshold settings, the calibration engineers usually are required to perform several cycles of data collection – data analysis – threshold adjustments. This involves huge engineering efforts and costs.

Real-Time Processing Methods

Real-time processing methods refer to the methods that could actively adjust misfire threshold based on the real-time data. They usually have the capability to lean the past detection indices, and use the learned information to adjust the threshold for the future misfire detections. To actively adjust the threshold setting to the best zone that separates normal firing and misfiring events is the biggest advantage for real-time processing methods. For example, in Figure 4, if a real-time processing method learns that the distributions of normal firing and misfire detection indices are shifted, it will adjust the threshold setting to the shaded zone in yellow color such that false detentions could be prevented.

The current popular real-time processing methods [11, 13], however, have serval obvious shortcomings that limit their applications. Due to the limitation of RAM/ROM and ECU throughput at high engine speed, the number of past cylinder events to be learned is very restrictive and the information extracted from these cylinder events is limited to mean or standard deviation of the detection indices [11]. The limitation on the number of cylinder events to be learned, however, restricts the learning is only limited to the immediate past events. Therefore, there will no tracking on the long-term changes.

In addition, the mean and/or standard deviation of the detection indices could be misled by the small sample size, such that it is difficult to find the zone that clearly separates the normal firming and misfiring events.

Finally, real-time processing methods usually only applies to the misfire detection methods with normalized detection indices. The main reason is because the threshold setting zone is relative stable for methods with normalized detection indices. Normalized detection indices refer to the detection indices that are normalized to a fixed range across all the engine speed – load zones by scaling and shifting. For example, the detection indices for normal firing events in Figure 5 is normalized to 0, and the detection indices for misfiring events is normalized to 1. The zone to set threshold is always between 0.5 and 0.75. When the engine operation changes from one speed-load zone to another, the threshold setting zone will not change, and this ensures the smooth transition of misfire threshold for zone-to-zone transitions. It is, however, very difficult to normalize misfire detection indices across all the operation zones.

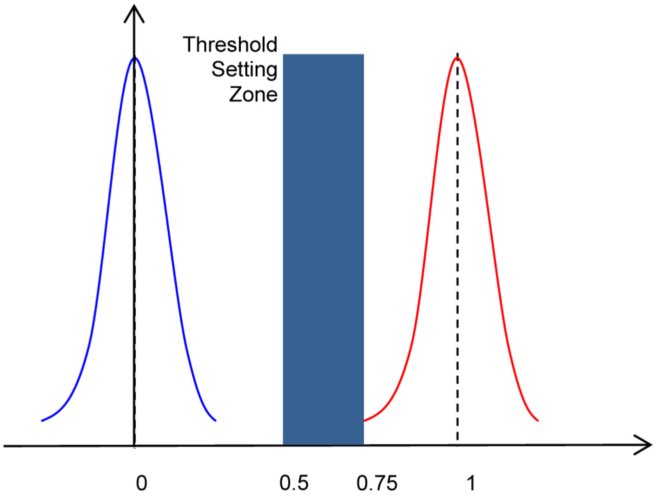

Figure 5. An example of normalized detection index distribution where the blue solid curve represents the normal firing distribution and the red curve represents the misfiring distribution. Note: the horizontal axis represents the values for detection indices, and the vertical axis represents the probability of the detection indices to be close to a certain value.

For detection indices that is not normalized, however, the zone to set threshold could shift dramatically when the engine operation crossing speed-load zones. If the threshold cannot be quickly adjusted to the correct zones, false detection / missed detection will happen. For example, the detection index threshold zone is [5, 15] for zone A ([0.15, 0.25] load and [2000, 3000] RPM engine speed zone). The detection index, however, jump to [80, 120] for zone B ([0.4, 0.5] load zone and [3000, 4000] RPM engine speed zone). For such a big jump in the threshold setting from zone A to zone B, if the threshold could not be immediately shifted to the right threshold setting zone (for example, the delay caused by filters or buffers), false detection will happen due to the application of improper threshold. This is one of the biggest changes to apply real-time processing methods to productions.

In addition, real-time processing methods face challenges in proper balancing of the swift changes in detection indices in the same speed-load zone and the long-term slow drifts of threshold zone changes.

ZONE-LEVEL AND BUFFER-LEVEL ADAPTATION BASED MISFIRE THRESHOLD DETERMINATION METHOD

Provided the analysis on the post processing and real-time processing misfire threshold detection methods, it is desired to have a method that dynamically adjusts the threshold, captures all short-term variations, tracks long-term changes, does not require normalized detection index, and could output valid threshold with small sample size. The proposed method in this study incorporates all the above desired functions. The high-level flowchart of the proposed zonelevel and buffer-level adaptation based misfire threshold determination method is as shown in Figure 6.

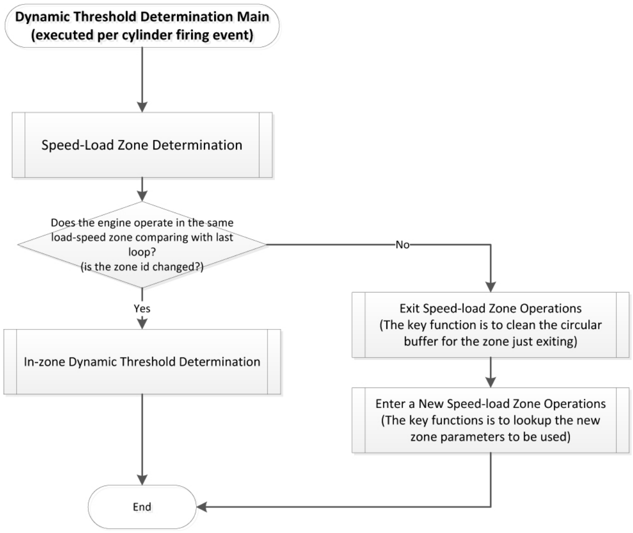
Dynamic Threshold Determination Main
(executed per cylinder firing event)
Speed-Load Zone Determination
Does the engine operate in the same load-speed zone comparing with last loop?
(is the zone id changed?)
No
Yes
In-zone Dynamic Threshold Determination
Exit Speed-load Zone Operations
(The key function is to clean the circular buffer for the zone just exiting)
Enter a New Speed-load Zone Operations
(The key functions is to lookup the new zone parameters to be used)
End

Figure 6. High-level flowchart of zone-level and buffer-level adaptation based misfire threshold determination method

Speed-Load Zone Determination

The first process box "Speed-load Zone Determination" is used to take care of the methods with detection indices that is not normalized. The engine speed and load are divided into different zones in such a way that the value of detection indices in each zone is relative stable. The breakpoints for engine speed and load are calibratable so that flexibilities are provided to different powertrain systems. Table 1 illustrates one such division of speed-load zones. The break points for engine speed and load in Table 1 is just for demonstration purpose, and the break points for actual productions should be adjusted per the specific engine configuration and the associated speed-load map. Based on the current engine speed and load, the speed-load zone for misfire threshold is then determined.

Table 1. This is an example of an engine speed - load division table and table title. Each zone is assigned a zone number for the convenience of zone lookup.

Load\RPM	<=500	[500, 1500)	[1500, 2500)	[2500, 3500)	[3500, 4500)	[4500, 5500)	[5500, 6500)	>=6500
<=0.15	1	2	3	4	5	6	7	8
[0.15, 0.25)	9	10	11	12	13	14	15	16
[0.25, 0.35)	17	18	19	20	21	22	23	24
[0.35, 0.45)	25	26	27	28	29	30	31	32
[0.45, 0.55)	33	34	35	36	37	38	39	40
[0.55, 0.65)	41	42	43	44	45	46	47	48
[0.65, 0.75)	49	50	51	52	53	54	55	56
[0.75, 0.85)	57	58	59	60	61	62	63	64
>=0.85	65	66	67	68	69	70	71	72

In-Zone Dynamic Threshold Determination

When the engine operates in the same speed-load zone, the "In-zone Dynamic Threshold Determination" process box is executed. The details of this process is shown in Figure 7. As long as the engine operates in the same speed-load zone, two circular buffers begin to store the detection indices for normal firing and misfiring cylinder events respectively. Data is written into the circular buffers one memory location after another. When a circular buffer is matured (indicated by MF_NF_Buf_Mature =1 or MF_MF_Buf_Mature =1), the newest value to be stored in circular buffer replaces the oldest value stored in the circular buffer. The buffer-level parameters, including mean (Mean_NF, Mean_MF), standard deviation (Std_NF, Std_MF), minimum (Min_NF, Min_MF), and maximum (Max_NF, Max_MF) of the buffered normal firing and misfiring data respectively, are then calculated per each cylinder event (or engine firing event). These buffer-level parameters are designed to quickly capture the dynamic changes of detection indices due to control or hardware reasons. That is to say, when the detection indices change in the same speed-load zone, the buffer-level parameters are designed to capture the trend of the changes. Meanwhile, the noises from individual cylinder events are filtered by the buffer. The size of the buffer is recommended to be the number of cylinder events for two complete engine revolutions, because all the cylinder events could be buffered once for such design of buffer size (this paper does not perform further study on the impact of buffer-size on the final threshold determination).

One thing that is worth to pay attention is that the circular buffer for normal firing events matures quickly (MF_NF_Buf_Mature =1) for normal engine operation, while the circular buffer for misfire events is usually empty (MF_MF_Buf_Mature =1) unless the engine is misfiring. That is to say, the buffer-level parameters for misfire events (including mean, standard deviation, maximum, and minimum) can NOT be calculated due to the immaturity of the buffer for misfire events. For such case, the zone-level parameters should be used for the final threshold determination. Please refer to Figure 9 for more details.

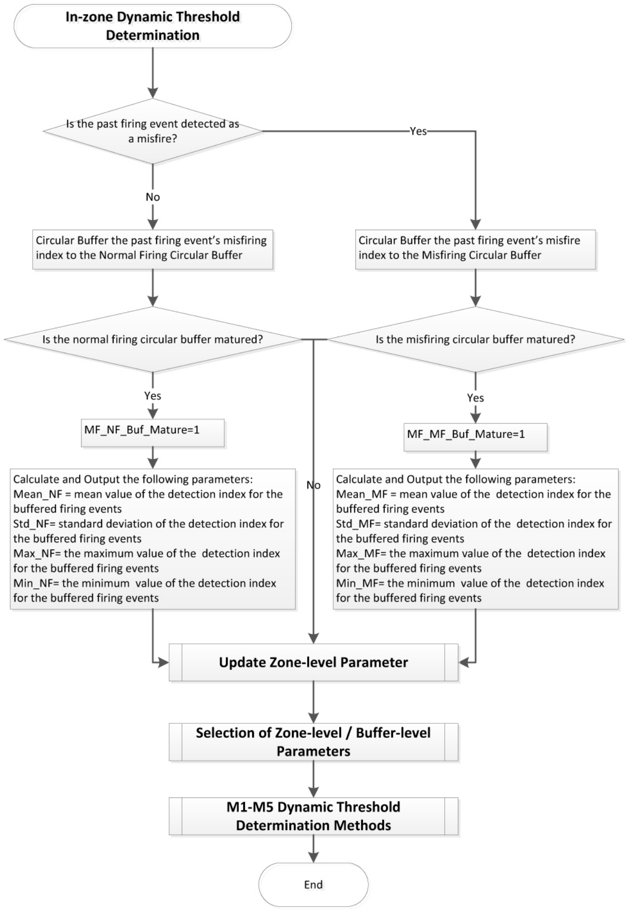
In-zone Dynamic Threshold Determination
Is the past firing event detected as a misfire?
Yes
No
Circular Buffer the past firing event's misfiring index to the Normal Firing Circular Buffer
Circular Buffer the past firing event's misfire index to the Misfiring Circular Buffer
Is the normal firing circular buffer matured?
Is the misfiring circular buffer matured?
Yes
Yes
No
MF_NF_Buf_Mature=1
MF_MF_Buf_Mature=1
Calculate and Output the following parameters:
Mean_NF = mean value of the detection index for the buffered firing events
Std_NF= standard deviation of the detection index for the buffered firing events
Max_NF= the maximum value of the detection index for the buffered firing events
Min_NF= the minimum value of the detection index for the buffered firing events
Calculate and Output the following parameters:
Mean_MF = mean value of the detection index for the buffered firing events
Std_MF= standard deviation of the detection index for the buffered firing events
Max_MF= the maximum value of the detection index for the buffered firing events
Min_MF= the minimum value of the detection index for the buffered firing events
Update Zone-level Parameter
Selection of Zone-level / Buffer-level Parameters
M1-M5 Dynamic Threshold Determination Methods
End

Figure 7. The detailed function for in-zone threshold determination

After each cylinder event, zone-level parameters, including mean (Zone_Mean_NF, Zone_Mean_MF), standard deviation (Zone_Std_NF, Zone_Std_MF), minimum (Zone_Min_NF, Zone_Min_MF), and maximum (Zone_Max_NF, ZoneMax_MF) of the associated zone, are slowly filter and updated and stored in the non-volatile memory of ECU as shown in Figure 8. The updating rate is controlled by the filtering coefficient C1 – C8. Note the initial values for the zone-level parameters are the average statistical parameters from the development vehicles, and they are stored in the non-volatile memory.

Because zone-level parameters are the default parameters to determine fail thresholds when the buffer-level parameters are not available, it is import to follow DFSS rules to set the initial values for zone-level parameters. Since the zone-level parameters are slowly updated, they are able to capture the long-term life-cycle changes in detection indices of that vehicle, and the associated fail threshold, thus, is adapted for those changes.

Noting that zone-level parameters are designed to capture the long-term life-cycle changes, the selection of the associated filtering coefficient C1 – C8 should comprehend it and prevent any short-term noise factors.

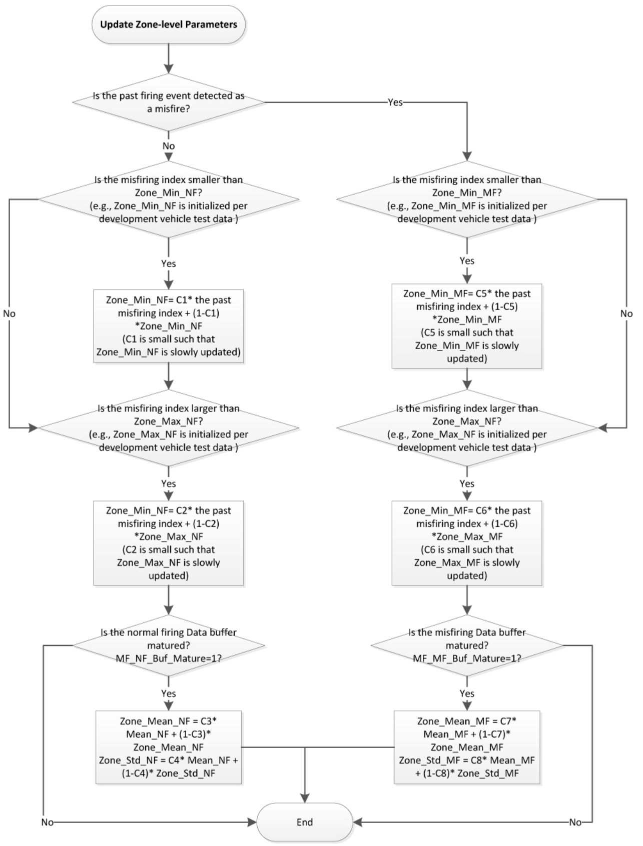
Update Zone-level Parameters
Is the past firing event detected as a misfire?
Yes
No
Is the misfiring index smaller than Zone_Min_NF? (e.g., Zone_Min_NF is initialized per development vehicle test data)
Is the misfiring index smaller than Zone_Min_MF? (e.g., Zone_Min_MF is initialized per development vehicle test data)
No
Yes
Yes
No
Zone_Min_NF= C1* the past misfiring index + (1-C1) *Zone_Min_NF (C1 is small such that Zone_Min_NF is slowly updated)
Zone_Min_MF= C5* the past misfiring index + (1-C5) *Zone_Min_MF (C5 is small such that Zone_Min_MF is slowly updated)
Is the misfiring index larger than Zone_Max_NF? (e.g., Zone_Max_NF is initialized per development vehicle test data)
Is the misfiring index larger than Zone_Max_NF? (e.g., Zone_Max_NF is initialized per development vehicle test data)
Yes
Yes
Zone_Min_NF= C2* the past misfiring index + (1-C2) *Zone_Max_NF (C2 is small such that Zone_Max_NF is slowly updated)
Zone_Min_MF= C6* the past misfiring index + (1-C6) *Zone_Max_MF (C6 is small such that Zone_Max_MF is slowly updated)
Is the normal firing Data buffer matured? MF_NF_Buf_Mature=1?
Is the misfiring Data buffer matured? MF_MF_Buf_Mature=1?
Yes
Yes
Zone_Mean_NF = C3* Mean_NF + (1-C3)* Zone_Mean_NF Zone_Std_NF = C4* Mean_NF + (1-C4)* Zone_Std_NF
Zone_Mean_MF = C7* Mean_MF + (1-C7)* Zone_Mean_MF Zone_Std_MF = C8* Mean_MF + (1-C8)* Zone_Std_MF
No
End
No

Figure 8. The zone-level parameter updating function

Before the final determination of threshold, algorithm is needed to select which parameters should be used (Figure 9). As previously-discussed in the beginning of this section, there could be cases that there are no valid buffer-level parameters. To address these cases, the algorithm in Figure 9 provides a solution: to use only zone-level parameters for the final threshold determination before the maturity of buffer-level parameters.

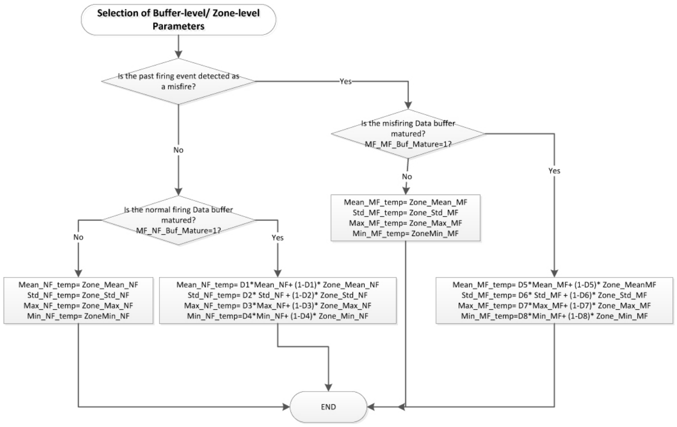
Selection of Buffer-level/ Zone-level Parameters
Is the past firing event detected as a misfire?
Yes
No
Is the misfiring Data buffer matured? MF_MF_Buf_Mature=1?
No
Yes
Mean_MF_temp= Zone_Mean_MF
Std_MF_temp= Zone_Std_MF
Max_MF_temp= Zone_Max_MF
Min_MF_temp= ZoneMin_MF
Is the normal firing Data buffer matured? MF_NF_Buf_Mature=1?
No
Yes
Mean_NF_temp= Zone_Mean_NF
Std_NF_temp= Zone_Std_NF
Max_NF_temp= Zone_Max_NF
Min_NF_temp= ZoneMin_NF
Mean_NF_temp= D1*Mean_NF+ (1-D1)* Zone_Mean_NF
Std_NF_temp= D2* Std_NF + (1-D2)* Zone_Std_NF
Max_NF_temp= D3*Max_NF+ (1-D3)* Zone_Max_NF
Min_NF_temp=D4*Min_NF+ (1-D4)* Zone_Min_NF
Mean_MF_temp= D5*Mean_MF+ (1-D5)* Zone_MeanMF
Std_MF_temp= D6* Std_MF + (1-D6)* Zone_Std_MF
Max_MF_temp= D7*Max_MF+ (1-D7)* Zone_Max_MF
Min_MF_temp=D8*Min_MF+ (1-D8)* Zone_Min_MF
END

Figure 9. Function to select buffer-level or zone-level parameters for final misfire threshold output

Once the parameters to be used for final threshold calculation is determined, this study proposes a dynamic output of the misfire detection threshold from one of the following five calculation methods as show in Figure 10.

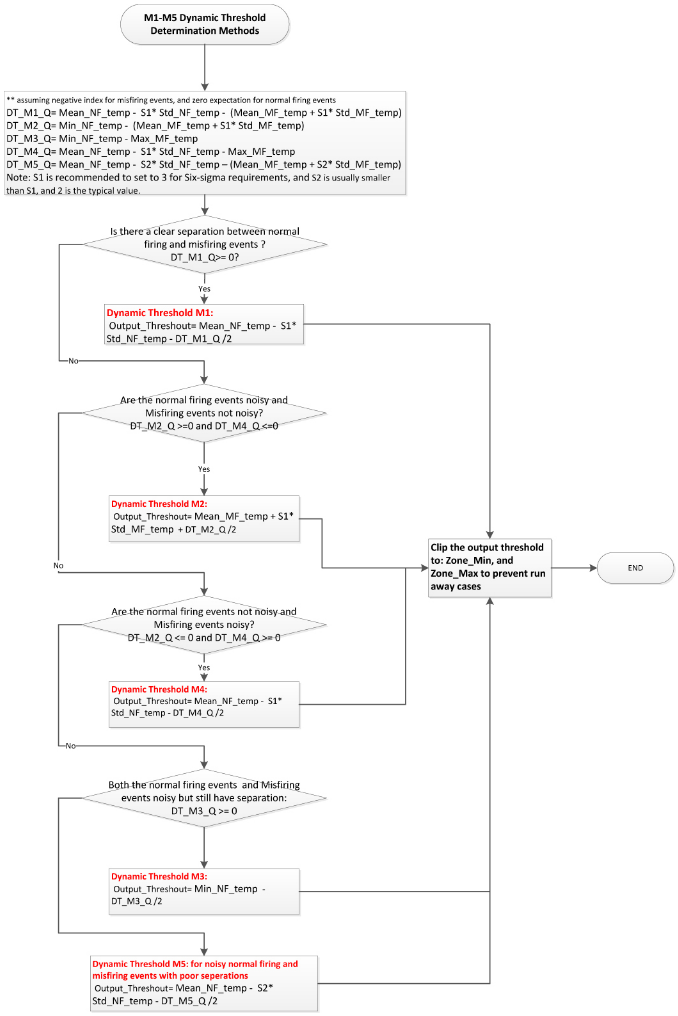
M1-M5 Dynamic Threshold Determination Methods
** assuming negative index for misfiring events, and zero expectation for normal firing events
DT_M1_Q= Mean_NF_temp - S1* Std_NF_temp - (Mean_MF_temp + S1* Std_MF_temp)
DT_M2_Q= Min_NF_temp - (Mean_MF_temp + S1* Std_MF_temp)
DT_M3_Q= Min_NF_temp - Max_MF_temp
DT_M4_Q= Mean_NF_temp - S1* Std_NF_temp - Max_MF_temp
DT_M5_Q= Mean_NF_temp - S2* Std_NF_temp – (Mean_MF_temp + S2* Std_MF_temp)
Note: S1 is recommended to set to 3 for Six-sigma requirements, and S2 is usually smaller than S1, and 2 is the typical value.
Is there a clear separation between normal firing and misfiring events ? DT_M1_Q>= 0?
Yes
Dynamic Threshold M1: Output_Threshout= Mean_NF_temp - S1* Std_NF_temp - DT_M1_Q /2
No
Are the normal firing events noisy and Misfiring events not noisy? DT_M2_Q >=0 and DT_M4_Q <=0
Yes
Dynamic Threshold M2: Output_Threshout= Mean_MF_temp + S1* Std_MF_temp + DT_M2_Q /2
Clip the output threshold to: Zone_Min, and Zone_Max to prevent run away cases
END
No
Are the normal firing events not noisy and Misfiring events noisy? DT_M2_Q <= 0 and DT_M4_Q >= 0
Yes
Dynamic Threshold M4: Output_Threshout= Mean_NF_temp - S1* Std_NF_temp - DT_M4_Q /2
No
Both the normal firing events and Misfiring events noisy but still have separation: DT_M3_Q >= 0
Dynamic Threshold M3: Output_Threshout= Min_NF_temp - DT_M3_Q /2
Dynamic Threshold M5: for noisy normal firing and misfiring events with poor seperations Output_Threshout= Mean_NF_temp - S2* Std_NF_temp - DT_M5_Q /2

Figure 10. M1-M5 Dynamic Threshold Determination Function

M1 is the method to set the threshold in the middle of the zone that is beyond 6 sigma of normal firing and misfiring detection indices. This is used for the case of clear separation between normal firing and misfiring events as show in Figure 11. Note it is assumed that negative indices are for misfiring events, and zero expectation is for all normal firing events. An example for the threshold setting is

$$\text{Threshold \_M1} = (\mu\_n - 3*\sigma\_n)/2 + (\mu\_m + 3*\sigma\_m)/2 \tag{1}$$

where μ_n, μ_m are the mean detection indices for normal firing and misfiring events that comes from the “Selection of Buffer-level/ Zonelevel Parameters” function, and σ_n, σ_m are the associated standard deviation.

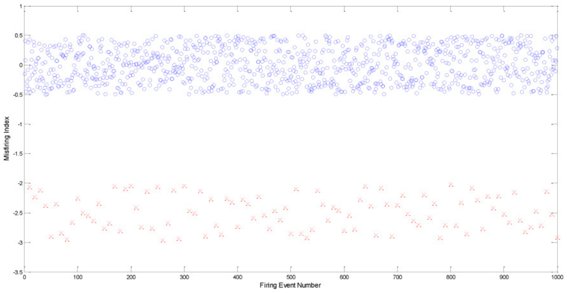

Figure 11. A typical plot for M1 application, where the blue circles indicate normal firing events, and the red crosses indicate misfiring events.

M2 is the method used for the case of noisy normal firing events with NOT noisy misfiring events, such as Figure 12. The key is to set the threshold to be less than the minimum of normal firing events and to be beyond 6 sigma zone of misfiring events. A typical setting for M2 threshold is

$$\text{Threshold \_M2} = \text{min\_n}/2 + (\mu\_m + 3*\sigma\_m)/2 \tag{2}$$

where min_n is the minimum normal firing detection index that comes from the “Selection of Buffer-level/Zone-level Parameters” function.

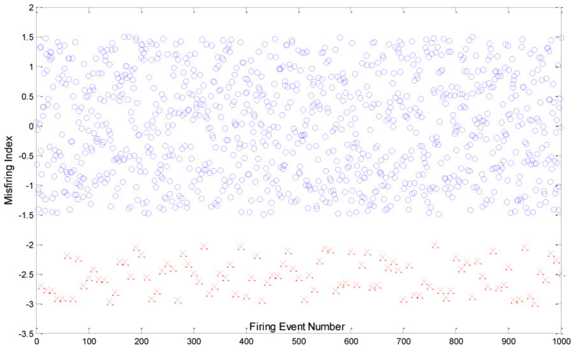

Figure 12. A typical plot for M2 application, where the blue circles indicate normal firing events, and the red crosses indicate misfiring events.

M3 is used for the case of noisy misfiring events with noisy normal firing events (while still separable) as show in Figure 13. The idea is to set the threshold to be between the minimum of normal firing events and the maximum of misfiring events. A typical formula for M3 threshold is

$$\text{Threshold \_M3} = \text{min\_n}/2 + \text{max\_m}/2 \tag{3}$$

where max_m is the minimum misfiring detection index that comes from the “Selection of Buffer-level/Zone-level Parameters” function.

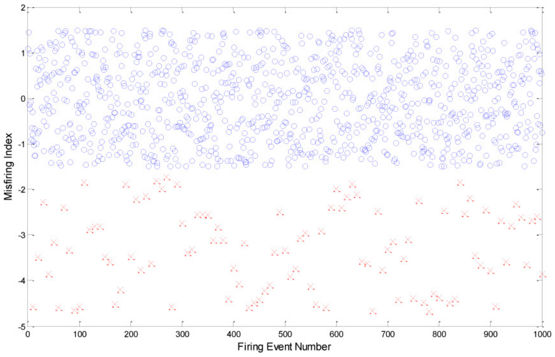

Figure 13. A typical plot for M3 application, where the blue circles indicate normal firing events, and the red crosses indicate misfiring events.

M4 is used for the case of noisy misfiring events with NOT noisy normal firing events as shown in Figure 14. The threshold is chosen to be larger than the maximum of misfiring events and to be beyond the 5-sigma zone of normal firing events. The M4 threshold could use the following equation to determine:

$$\text{Threshold \_M4} = (\mu\_n - 3*\sigma\_n)/2 + \text{max\_m}/2 \tag{4}$$

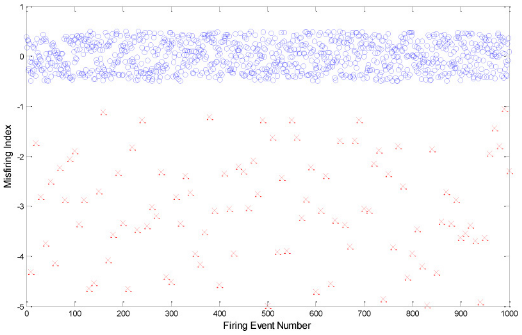

Figure 14. A typical plot for M4 application, where the blue circles indicate normal firing events, and the red crosses indicate misfiring events.

Finally, if the 6-sigma zones for normal firing and misfiring events overlap, and the minimum of normal firing events is smaller than the maximum of misfiring events (Figure 15), there is no clear separation between normal firing and misfiring event. For such case, the current detection methods associated with the misfiring index calculation essentially failed. To ensure continuous misfiring monitor and minimize the chances of false detection, one has to yield on the confidence level for threshold setting. For example, it may be possible to set a threshold to be beyond 4-sigma zone for normal firing and misfiring events. The following equation (5) shows a suggested threshold setting with M5method:

$$\text{Threshold \_M5} = (\mu\_n - 2*\sigma\_n)/2 + (\mu\_m + 2*\sigma\_m)/2 \tag{5}$$

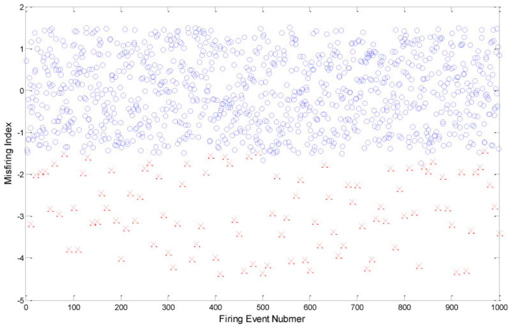

Figure 15. A typical plot for M5 application, where the blue circles indicate normal firing events, and the red crosses indicate misfiring events.

Finally, the final threshold to be used for misfire detection has to be clipped to a certain boundary for that specific zone in order to prevent run-away cases.

Operations to Exit a Speed-Load Zone

This main purpose for the operations when exiting a speed-load zone is to ensure that the old zone's data will not impact the threshold determination in a new zone. To purge the circular buffers (delete all the detection indices that are stored in the circular buffers) is the main operation here.

Operations to Enter a New Speed-Load Zone

This main function here is to ensure the smooth transition when entering a new speed-load zone. To load the zone parameters for the new zone is one of the main functions. These zone parameters are used to determine the dynamic threshold before the maturity of the circular buffers which are purged when leaving the old zone.

Summarizing the above, an innovative real-time processing method is introduced in detail. This method utilizes buffer-level parameters to trace all short-term variations, and make use of the zone-level adaptations to track long-term changes. It does not require normalized detection index, and could output valid threshold with small sample size.

Comparing with the traditional misfire threshold determination method, this proposed method is more complicated in terms of algorithm design, and potentially consumes more ECU resources in terms of RAM/ROM. Its advantages, such as short-term and long-term adaptations, however, outshines these shortcomings, and make misfire detection robust again potential noise factors caused by new control strategies and new engine components.

EXAMPLE

To confirm the effective of the threshold setting method proposed in this paper, simulations are performed based on the test data collected from vehicles. Figure 16 shows how the actual misfire detection indices (that is not normalized) look like in a specific speed-load zone, when misfires are generated in a fixed cylinder. Due to restrictions of data/information protection, the details of engine type/vehicle model cannot be provided in this paper.

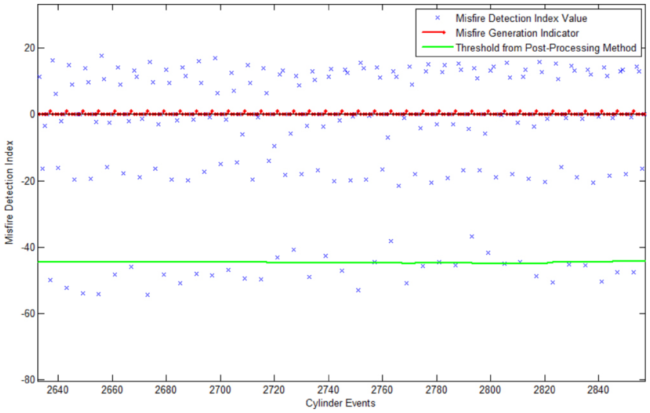

Figure 16. An example of a failed threshold setting from a post-processing method

Each blue cross in Figure 16 represents one cylinder event. The solid green line is the fixed threshold comes from a post-processing method. As you can see, this threshold fails to separate misfire events from normal firing events, because there are lots of misfires that are NOT detected (all generated misfire events are indicated by the red dots with non-zero values).

With the proposed in this paper, the dynamic threshold easily separates normal firing and misfiring events.

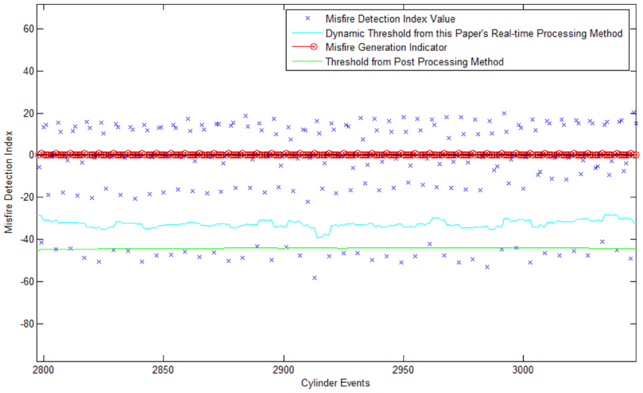

Figure 17. An example of how the dynamic threshold proposed in this paper effectively separates normal firing and misfiring events

Even when the detection indices for some normal firing events are really tossed up with the actual misfiring events (Figure 18).

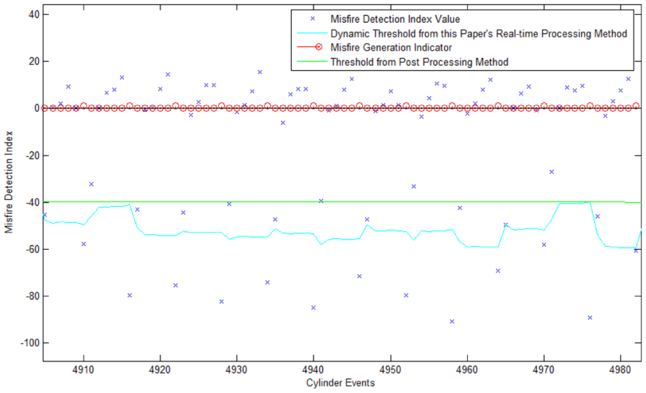

Figure 18. An example of the effectiveness of dynamic threshold to handle noisy normal firing events when the post-processing methods completely fails

During engine speed-load zone changes (Figure 19), the dynamic threshold closely follows the trends of how detection indices vary and always set the threshold in the best zone for separation, while the threshold from the post-processing method is lack of such adjustments. For example, starting firing event 3250, the threshold from the post-processing method (the green curve in Figure 19) is pushed up due to the zone changes, because the threshold is calibrated to be smaller in the new zone. The dynamic threshold (the cyan curve in Figure 19), however, is pushed down due to the noisy level of normal firing events in the new zone increases.

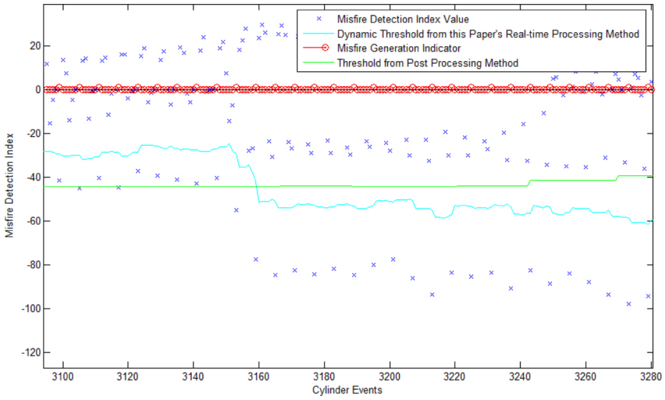

Figure 19. An example of the effectiveness of dynamic threshold to handle zone changes

In short, all the simulation results and actual test data results show that the dynamic threshold is very effective to handle complicated situations for misfire detection and it can always find the best threshold setting to separate normal firing and misfiring events.

SUMMARY

This study points out the disadvantages of post-processing methods for misfire threshold determination, and the limitations of the existing real-time processing methods. By utilizing buffer-level parameters to trace all short-term variations, and make use of the zone-level adaptations to track long-term changes, this study proposes a unique approach to dynamically determine the misfire threshold. This method fits the needs of accurate misfire detection under more and more complicated engine operating conditions that could make the detection index vary in a larger than expected range. The proposed method can help improve the detection ratio of misfire for internal combustion engines, and make the detection method unimpressionable to engine controls and hardware changes.

REFERENCES

1. California Air Resource Board 1968.2, www.arb.ca.gov
2. Worldwide emissions standards – passenger cars and light duty vehicles, 2015, Delphi Automotive, http://delphi.com/docs/default-source/catalogs/delphi-worldwide-emissions-standards-pc-ldv-15-16.pdf
3. Willimowski, M. and Isermann, R., “A Time Domain Based Diagnostic System for Misfire Detection in Spark-Ignition Engines by Exhaust-Gas Pressure Analysis,” SAE Technical Paper 2000-01-0366, 2000, doi:10.4271/2000-01-0366.
4. Ohashi, Y., Koiwa, M., Okamura, K., and Ueda, A., “The Application of Ionic Current Detection System for the Combustion Condition Control,” SAE Technical Paper 1999-01-0550, 1999, doi:10.4271/1999-01-0550.
5. Hellring, M., Munther, T., Rögnvaldsson, T., Wickström, N. , “Robust AFR Estimation Using the Ion Current and Neural Networks,” SAE Technical Paper 1999-01-1161, 1999, doi:10.4271/1999-01-1161.
6. Sellnau, M., Matekunas, F., Battiston, P., Chang, C. , “Cylinder-Pressure-Based Engine Control Using Pressure-Ratio-Management and Low-Cost Non-Intrusive Cylinder Pressure Sensors,” SAE Technical Paper 2000-01-0932, 2000, doi:10.4271/2000-01-0932.
7. Merkisz, J., Bogus, P., Grzeszczyk, R., Overview of engine misfire detection methods used in on board diagnostics, Journal of Kones, Combustion Engines, Vol 8, No 1–2, 2001
8. Klenk, M., Moser, W., Mueller, W., and Wimmer, W., “Misfire Detection by Evaluating Crankshaft Speed - A Means to Comply with OBDII,” SAE Technical Paper 930399, 1993, doi:10.4271/930399.
9. Mukherjee, U., Kundu, A., System and method to detect errors and predict potential failures, 2008, US 7409594B2
10. Volponi, A.J., Hybrid model based fault detection and isolation system, 2004, US 7415328B2
11. Wu, Z., Lee, A., Real-time misfire detection for automobile engines with medium data rate crankshaft sampling, US 5862507A
12. Wu, Z., Lee, A., Misfire Detection Using a Dynamic Neural Network with Output Feedback, US 5862507A
13. Ohsaki, M., Ozaki, H., Misfire detecting apparatus for internal combustion engine and method thereof, US 7117727 B2
14. Assaf, T.K., Naik, S.M., Mathews, D. S., Engine misfire and rough road detection systems and methods, US 7950273B2
15. Bidner, D. K., Zimlich, G. A., Robichaux, J. D., Onboard diagnostic misfire detection monitor for internal combustion engines, US 6415656 B1
16. Lewis, D., System and Method for Detecting Engine Misfires, US 7540185B2

CONTACT INFORMATION

Dr. Yichao Guo
Principle Engineer
Powertrain OBD – Future Development
Nissan Technical Center North America, Inc.
yichao.guo@nissan-usa.com
Phone: +1 248.488.4140

Dr. Guo is a former Powertrain Controls & OBD Subsystem Engineer in Ford Motor Company and the study in this paper was fully conducted and completed while he worked in Ford.

ACKNOWLEDGMENTS

The present study is fully conducted and completed while the author worked in Ford Motor Company. I would thank my former colleagues, Chris Weber, Robert Baskins, and Robert Jentz for their generous help and support for this study.

DEFINITIONS/ABBREVIATIONS

CARB - California Air Resources Board

OBD - On-board Diagnostics

DFSS - Design for Six Sigma

Positions and opinions advanced in this paper are those of the author(s) and not necessarily those of SAE International. The author is solely responsible for the content of the paper.