UKACC International Conference on Control 2012
Cardiff, UK, 3-5 September 2012

A DNA Sequencing Based Approach to Fault Diagnosis in Automotive Electronic Networks

Roozbeh Bonyadi, R. Peter Jones
School of Engineering
The University of Warwick
Coventry, United Kingdom

James Taylor, Mark Amor-Segan
WMG
The University of Warwick
WMG is a member of the High Value Manufacturing Catapult

***Abstract*—The increasing number of Electronic Control Units within the network of a vehicle is increasing the level of complexity of these networks. Thus, fault diagnosis of these sophisticated systems becomes more complex. The aim of this paper is to provide a technique to enable CAN-based fault detection in a premium vehicle network or any other network which uses the Controller Area Network (CAN) protocol for communication. The fault detection technique described here is based on a sequential behaviour of the CAN network of a vehicle and by using signal processing methods commonly used in DNA sequencing analysis, fault detection was achieved and data were classified in clusters of normal scenarios and fault scenario.**

Keywords-Fault Diagnosis;Electronic Control Unit; Controller Area Network Protocol; DNA Sequencing; Density Power Spectrum; Cross Correlation; Classification; Clustering

I. INTRODUCTION

Over the past two decades, numerous mechanical and hydraulic systems have been gradually replaced by electronics due to the development of embedded systems in automotive networks of vehicles [1]. These systems precisely assist driver to take control of the vehicle through the steering, engine control, suspension, braking, stability and traction functions [2]. Each function enabled by these electronic systems has an embedded electronic control unit (ECU) [3], or is distributed among a group of ECUs. The automotive networks of vehicle became more complex as the number of ECUs increased. ECUs need to exchange information among each other. Consequently, for managing data and granting read and write access to ECUs, protocols such as Controller Area Networks (CAN), Local Interconnect Network (LIN), Media Oriented Systems Transport Network (MOST) and more recently the future technology, FlexRay, were defined. One of the most widely used protocols in automotive systems is CAN protocol defined by a Robert Bosch GmbH in early 1990's [4].

In premium vehicles there are over 50 ECUs for both infotainment and control systems which are connected to each other. This increases the level of complexity of the network. Thus, the need of fault diagnosis within these networks becomes more important. To the best of our knowledge system wide diagnostic is not yet available. Fig.1 shows a typical premium vehicle network system architecture that supports both infotainment systems and control systems.

Here the fault diagnosis of the CAN network of a premium vehicle is studied. This vehicle has two different bandwidths: Low speed CAN which takes control of body functions and high speed CAN which takes control of the critical functions of the vehicle which needs real time control. The fault diagnosis within modern vehicle network systems so that faults within the complex network of the vehicle can be identified as belonging to a certain category is investigated here. The introduced method can also be applied in new areas, adopting the real-time fault detection and on-board fault diagnosis, utilising the data rich environment of the CAN network. To verify the system's behaviour fault injection technique is used. This is inserting of an artificial fault into the system and monitoring the response of the whole system [5].

The main purpose was to find a specification in different tests to find a way to distinguish fault scenarios from normal scenarios. After investigating different aspects of the CAN network, based on DNA sequencing approach, a method for distinguishing fault and normal scenarios is presented. Classification and data mining were used to classify scenarios.

It is essential to understand the level of complexity of CAN networks. In the next section different applications of this method is provided and data types and data gathering is discussed in section 3. In section 4, a DNA sequencing based method is proposed as a method to distinguish fault scenarios from normal scenarios. Section 5 provides the results of data analysis. The conclusion of the paper is provided in section 6.

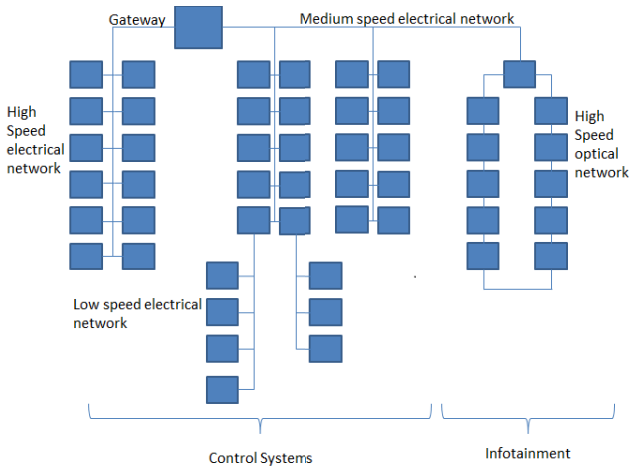

Fig. 1 A typical premium vehicle network system architecture

II. Application

The CAN protocol enables robust serial communication and was introduced by German automotive system supplier Robert Bosch. It was chosen for embedded systems networked applications in various markets such as medical equipment, test equipment, industrial automation and mobile machines. More recently, CAN networks have been popular in automation and control applications too [4, 8]. The techniques used in this paper were developed on the CAN network of a vehicle, but they can also be used in any other applications apart from vehicle industry in which CAN protocols are used.

III. Data Collection

For the data collection, specific experiments were carried out on a representative vehicle network. The aim of these experiments was to collect data relating to illegal vehicle wake-up. The vehicle network is monitored and saved on computer, utilising a CANcase device to interface to the high speed and low speed networks via USB.

A. Message Types

Two types of messages are published through the CAN network of car, periodic messages and non-periodic messages. Periodic messages are sent regularly through the network for long periods of time. Non-periodic messages are a classification of messages which have an external actuator and are the main focus of this paper. In this network, there are 14 message IDs in the high speed CAN and 47 message IDs in the low speed CAN which are involved in the wake-up process.

B. Scenarios and Tests

The network of the vehicle enters sleep mode when there are no interactions between ECUs to reduce the battery consumption. The sleep mode is a very low-current standby mode with bus wake-up capability [6].

When a non-periodic message is sent through the network, ECUs start to communicate by sending information about their task. After completion of the task, ECUs start to notify that they are ready to enter sleep mode, and upon receiving a confirmation message of "go to sleep mode" ECUs gradually start entering sleep mode [7]. The period of time which network goes from sleep mode to wake-up mode and then again back to sleep mode is called a test. The external action causing data to flow, which classifies the type of the test, is called a scenario.

If the non-periodic message which produced a test is normal function defined by manufacturer, the scenario is considered as a normal scenario. A fault scenario is caused by artificially injecting a non-periodic message into the network.

C. Data Visualisation

Each test is stored as a text file into the computer. As each message passing through the CAN is logged into the computer, it is stored in a new line. Fig.2 shows a small part of data collected from one normal scenario. The first column is time which shows the time that messages were logged in. The second column shows in which CAN the message has been transmitted. If it is 1 it shows that message belongs to the High speed CAN and if it is 2 the message belongs to the Low speed CAN. The third row is the identification number or ID of message in hexadecimal code. Each message has individual number which shows specific data. The rest of 8 columns show data within the message in hexadecimal code. Each message has individual ID which shows specific data. In the analysis stage, each ID is converted into decimal numbers. A normal scenarios occurs when a valid wake-up message is sent, such as pressing the lock button on the car key, and a fault scenario is produced by sending a message not normally involved in wake-up, such as message 1C8, into the data. As all normal scenarios belong to the body functions of the vehicle, analysis will be focused on the low speed CAN.

IV. Diagnostic Approaches

A. DNA Sequencing

The digital genomic information are characterised in a form of sequence of finite numbers of entries coming after each other [9]. There are some similarities between the collected data from the real vehicle's CAN network and DNA strands. As CAN messages enter the network through competitive arbitration, the sequence of messages becomes a feature of the network, similar to the sequence of bases in a DNA strand. In this section signal processing tools are applied on these sequences. Using classification methods described in section D fault diagnosis in the CAN network of the vehicle is achieved.

B. Binary Indicators

47 different messages passing through the low speed CAN network were identified as significant. Thus 47 numbers are multiplied in its corresponding binary indicator sequences and then all of them are summed in order to build the sequence. Each sequence in the binary indicators (u_{IDi}) is valued "1" if ID_i exists in that sequence or a value of "0" if ID_i is not present in that sequence:

$$x[n]=ID_1.u_{ID_1}[n]+ID_2.u_{ID_2}[n]+\ldots+ID_{47}.u_{ID_{47}}[n]$$
$$n=0, 1, 2, 3, \ldots, N-1 \quad (1)$$

```
6.098786 2  1C8          Rx   d 8 80 00 02 08 18 02 20 11
6.099729 2  248          Rx   d 8 00 00 09 56 73 39 18 00
6.100736 2  4D8          Rx   d 8 00 00 00 00 00 00 00 00
6.104767 2  68           Rx   d 8 FC 00 00 3E 00 00 42 04
6.105726 2  1A8          Rx   d 8 00 11 3C C5 8E 7D 00 00
6.106669 2  368          Rx   d 8 DD C8 D7 26 00 00 00 00
6.107628 2  398          Rx   d 8 2A 27 10 11 00 00 00 00
6.108611 2  88           Rx   d 8 FF 04 00 1F 40 00 FF FF
6.109570 2  A8           Rx   d 8 FE F0 19 29 80 3F 00 00
6.110521 2  268          Rx   d 8 EB 08 00 00 04 02 09 01
6.111503 2  2A8          Rx   d 8 C0 00 00 00 01 60 00 00
6.112526 2  208          Rx   d 8 00 02 00 00 00 00 00 00
```

Fig. 2 An example of data collected from vehicle in text file format opened in Windows Notepad.

Frequency domain analysis using DFT is performed, using a sequence of length of N to provide the frequency content, X[k], at a frequency of k.

$$X[k]=ID_1.U_{ID_1}[k]+ID_2.U_{ID_2}[k]+\ldots+ID_{47}.U_{ID_{47}}[k]$$
$$k=0, 1, 2, 3, \ldots, N-1 \qquad (2)$$

In which $U_{ID_1}[k]$, $U_{ID_2}[k]$ … $U_{ID_{47}}[k]$ are the DFT of each of the binary indicator sequences x[n], producing a 47 dimensional representation of a frequency spectrum of a sequence of the messages passing through the CAN. The total spectral content, S[k], of a sequence of messages in the CAN at frequency of k is:

$$S[k]=\left|U_{ID_1}[k]\right|^2+\left|U_{ID_2}[k]\right|^2+\ldots+\left|U_{ID_{47}}[k]\right|^2 \qquad (3)$$

Three normal scenarios and two fault scenarios were established to perform fault diagnosis. Fig. 3 illustrates the Density Power Spectrum of the sequence of a test from one of the normal scenarios. The peaks of the power spectrum alter between scenarios, although all scenarios occur in a similar frequency range. In the case of DNA sequences, there are only four binary indicators and as a result there is only one peak that occurs at the frequency of N/3 where N is the length of sequence of x[n]. In the message sequences of the CAN network, there are 47 binary indicators. Hence, there is more than just one peak in the sequence of IDs.

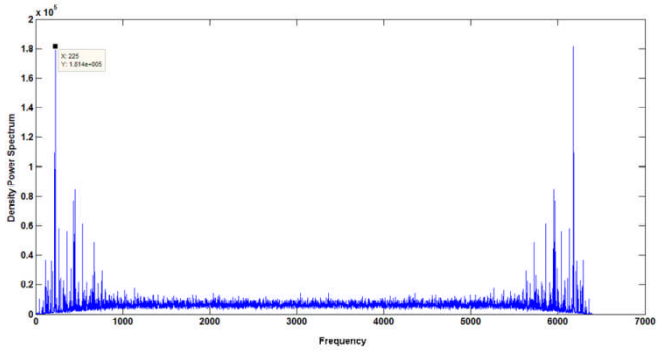

Fig. 3 Density power spectrum of a test from scenario of pressing open boot button on car key.

C. Clustering Data

Clustering is assigning a set of data into subsets or groups so that data in the same clusters have similarity in some cases [10]. The frequency peaks in the DPS cluster the tests from these five scenarios into 3 different groups: normal scenario 1, normal scenario 2 and faulty scenarios.

D. Classifier and Classification Method

Classification is a data mining process which aims to accurately assign target classes to data sets. The classifier which is used here is a hybrid system which effectively combines hand-built classifiers and empirical learning methods together [11]. This has the advantage of being able to utilise the characteristics of entire power spectrum in the classification.

One of simplest classification problems is binary classification between two states. This means that target group has only two possible states and classifier predicts state of the data set on a basis of whether they have the property or not. This type of classification is used in this paper.

$$for\ i,j = 1{:}5$$
$$if\ |AXCTS(1,j) - AMVXC(i,j)| \leq STD(i,j)$$
$$\rightarrow scenario\ group\ of\ i$$
$$if \begin{cases} 1 \leq i \leq 2 & \rightarrow normal\ scenario\ group\ 1 \\ i = 3 & \rightarrow normal\ scenario\ group\ 2 \\ i > 3 & \rightarrow fault\ scenario \end{cases}$$

Fig. 4 Algorithm of comparing a test set with training sets for classification.

1) Training Set

Classification methods require training to be effective, which means a training data set is needed. This training set used to establish the relationship between predictors and targets. Different features can be defined in order to extract useful information from training set, and this information can be used to determine classes of future test sets.

Since the length of sequences in different scenarios was different, the test with the longest length, test l, was considered as a basis length of the sequences (length *l*). The length of test l according to the 100 randomly picked tests is 24592 sequences. For the rest of data sets in the test sets, if the length of the sequence of the messages in a test is less than *l,* then sequences of zeros are added to all of the 47 binary indicator sequences of $u_{ID_1}[k]$, $u_{ID_2}[k]$, …, $u_{ID_{47}}[k]$; and if the length of the sequence is greater than *l*, the first *l* sequences were considered, and the rest discarded. This is done because the values produced by this method are dependent on the length of the data sets. By using a standard base length, this effect will be normalised.

Next, the DFT of all the binary indicators are calculated and the density power spectrum of the whole sequence is achieved using equation 3. The density power spectrum of a discrete signal of the messages passing sequentially through the CAN network of the vehicle is a discrete signal. For classifying different scenarios, these discrete signals need to be compared to each other and form the basis of the classification scheme.

2) Comparison Tool

The method chosen to compare these discrete signals is the cross correlation. It is a common method for estimating the correlation rate between two series. For a discrete signal, cross correlation is:

$$R_{xy}(d)=\begin{cases}\sum_{n=0}^{l-d-1} x[n+d]y^*[n] & d\geq 0 \\ R_{xy}^*(d) & d<0\end{cases} \qquad (4)$$

In which, *l* is the base length of the data sets and d is the signal delay. With the delay of d in x[n], the degree of correlation between x[n] and y[n] is calculated by dot-product of these two signals. The signals are considered to be periodic, so the cross correlation has twice of the length of its original signals. This operation was performed in MATLAB using the cross correlation command, which by default, computes the

raw correlation between two signals without normalising it. The lengths of all data sets are converted to *l* so the normalised cross correlation which is the cross correlation divided by the length of the signals is achieved in advance (biased normalisation).

$$R_{xy,\,biased}(d)=\frac{1}{l}R_{xy}(d) \quad (5)$$

20 tests are chosen from each of the five scenarios, giving a total of 100 tests. The density power spectrum of all the 100 tests is calculated. Hence, there are 100 discrete time signals available. The cross correlation of each of the two of these power spectrums are calculated. The result of these calculations are stored in a massive cell with a dimension of 100×100 and in each of these cells, the cross correlation results with the length of 2×*l* is stored.

The peak of these cross correlations shows the maximum correlation of the two tests. The cross correlation of the density power spectrum of each test set and each test in the training sets are calculated and stored in a massive cell. For the calculated cell of the cross correlation with the dimension of 100×100 maximum values of the cross correlation is stored in a matrix with the same dimensionality to create the training sets.

Next, the matrix is divided into 5 matrices with dimension of 20×20 which are the maximum values of cross correlation between the scenario 1 and all the 5 scenarios (including auto-correlation with scenario 1). This is the procedure of making the training set for the first scenario. This should be completed for the second scenario, the third scenario and so on.

The maximum value of the cross correlation, shows rate of correlation between two scenarios. Average of maximum values of the cross correlation (AMVXC) of the 20 tests of each matrix are calculated and saved for each scenarios (Table 1). Also the Standard Deviation (STD) of these maximum values of the cross correlation of these 20 tests are calculated and stored for each scenario (Table 2).

The average value and the standard deviation of each matrix create the training sets for the classification. Fig. 5 shows the steps and the procedure of creating the training sets for the classification of first scenario. So the result of all these calculation will be a 5×5 matrix of average of the maximum values of the cross correlation between the scenarios and another 5×5 matrix is the standard deviation of the maximum values of the cross correlation between the scenarios.

3) Classifying a Test Set

For each given test set the cross correlation between the density power spectrum of that test set and the density power spectrum of the 100 tests used to build the training set is calculated. This results in 100 cross correlations.

The maximum value of the cross correlation between the test set and the 20 tests of the scenario 1 will be stored in a separate matrix; the maximum value of the cross correlation between the test set and the 20 tests of the second scenario will also be stored in another matrix and so on. Eventually, there will be 5 matrices with dimension of 1×20. The average of these values in each matrix will be calculated and stored in a matrix called Average of XCorrelation of Test Set or AXCTS (i ,j) (Table 3).

The AXCTS is then subtracted from each of the scenario's AMVXC (Table 4). Comparing each row with its counterpart in Table 2, it can be seen that this example test belongs to scenario 2 (normal scenario group 1) as the values in this row are less than those in Table 2. This algorithm is illustrated in Fig.4.

V. RESULTS

Six test sets from each of the 5 scenarios (total of 30 tests) are randomly picked and considered as the test sets. The classification procedure is carried out on each test set. The confusion matrix in Table 5 shows 100% accuracy for assigning scenarios to the 3 classification groups. Fig.6 shows these 30 test sets in AXCTS matrix classified in the three clusters. In this plot each row of the AXCTS matrix is shown with 5 markers representing AXCTS value for its corresponding scenario. As it can be seen for all tests, the clusters are distinguished.

TABLE I. THE AVERAGE OF THE MAXIMUM VALUES OF THE CROSS CORRELATION FOR THE TRAINING SET

AMVXC	Scenario1	Scenario2	Scenario3	Scenario4	Scenario5
Scenario 1	68.5296E+11	64.8896E+11	90.7273E+11	32.8471E+11	34.0116E+11
Scenario 2	64.8896E+11	61.6390E+11	85.9672E+11	31.5055E+11	32.6112E+11
Scenario 3	90.7273E+11	85.9672E+11	623.5030E+11	45.9586E+11	47.3855E+11
Scenario 4	32.8471E+11	31.5055E+11	45.9586E+11	18.4082E+11	18.9465E+11
Scenario 5	34.0116E+11	47.3855E+11	18.9465E+11	19.5366E+11	24.0919E+11

TABLE II. THE STD OF THE MAXIMUM VALUES OF THE CROSS CORRELATION FOR THE TRAINING SET

STD	Scenario1	Scenario2	Scenario3	Scenario4	Scenario5
Scenario 1	2.8279E+11	3.5061E+11	5.5539E+11	2.6415E+11	2.4432E+11
Scenario 2	3.5061E+11	3.9585E+11	6.5875E+11	2.6612E+11	2.5055E+11
Scenario 3	5.5539E+11	6.5875E+11	32.5400E+11	3.1451E+11	3.1038E+11
Scenario 4	2.6415E+11	2.6612E+11	3.1451E+11	1.6417E+11	1.5966E+11
Scenario 5	2.4432E+11	2.5055E+11	3.1038E+11	1.5966E+11	1.5493E+11

TABLE III. THE AVERAGE VALUE OF THE MAXIMUM VALUES OF THE CROSS CORRELATION BETWEEN GIVEN TEST SET AND THE TRAINING SETS

AXCTS	Scenario 1	Scenario 2	Scenario 3	Scenario 4	Scenario 5
Test Set	65.4680E+11	62.0810E+11	85.616E+11	31.673E+11	32.792E+11

TABLE IV. DIFFERENCE BETWEEN THE AMVXC OF THE TRAINING SETS AND AXCTS OF THE TEST SET

|AXCTS – AMVXC|	Scenario 1	Scenario 2	Scenario 3	Scenario 4	Scenario 5
Scenario 1	3.06E+11	2.81E+11	5.11E+11	1.17E+11	1.22E+11
Scenario 2	0.58E+11	0.44E+11	0.35E+11	0.17E+11	0.18E+11
Scenario 3	25.30E+11	23.90E+11	538.00E+11	14.30E+11	14.60E+11
Scenario 4	32.60E+11	30.60E+11	39.70E+11	13.30E+11	13.80E+11
Scenario 5	99.50E+11	109.00E+11	105.00E+11	51.20E+11	56.90E+11

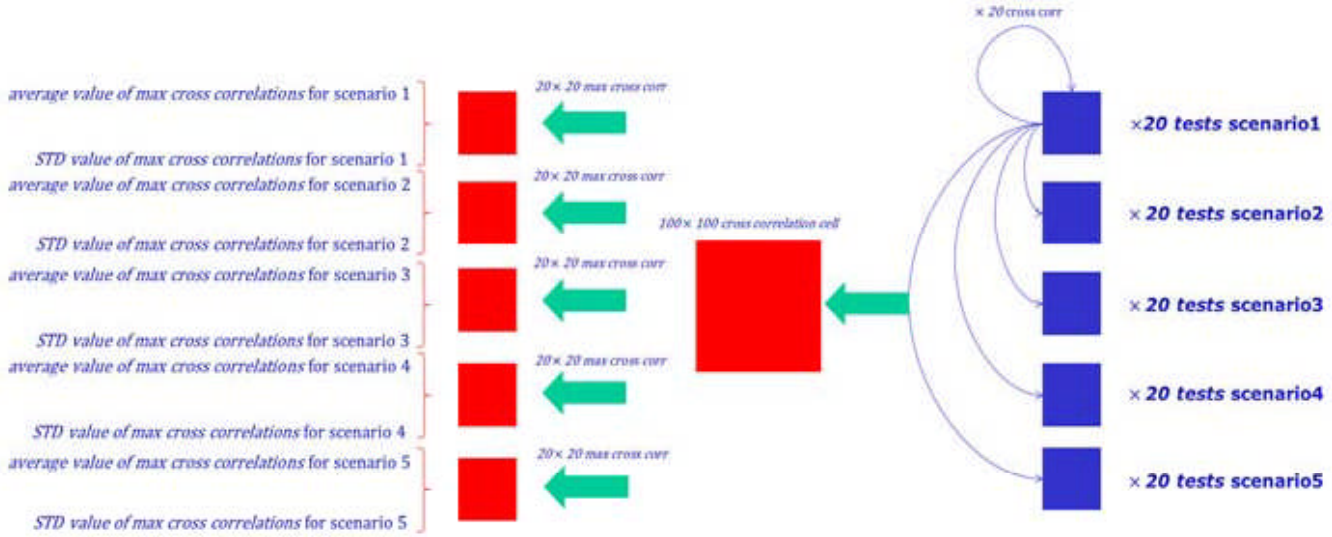

Fig. 5 Procedure of creating training set to be used for DPS method for classification for first scenario

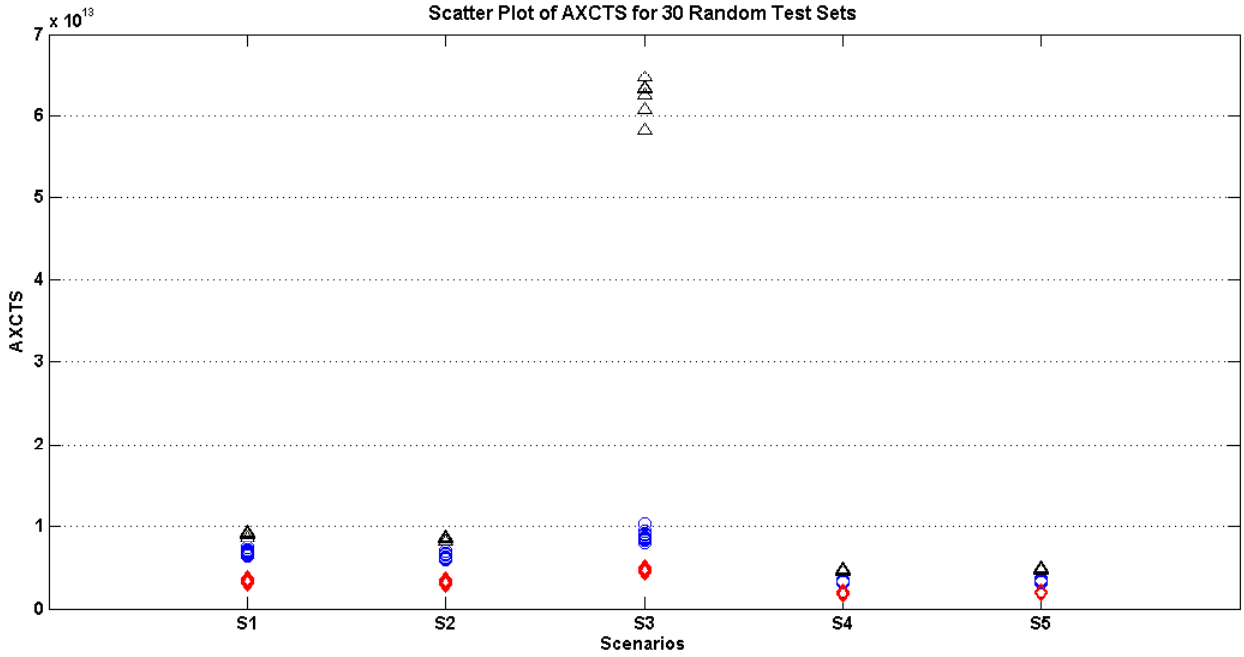

Fig. 6 Results, showing the matrix of AXCTS classified the 30 test sets in 3 clusters; Diamond: Fault scenario, Circle: Normal Scenario 1, Triangle: Normal Scenario 2

TABLE V. CONFUSION MATRIX FOR CLASSIFYING TEST DATA

	Scn.1	Scn.2	Scn.3	Scn.4	Scn.5
Group 1	6	6	0	0	0
Group 2	0	0	6	0	0
Faults	0	0	0	6	6

VI. CONCLUSION

In this paper, development and diversity of networks in vehicles was studied. The main focus was the Controller Area Network (CAN) protocol as a communication tool. Moreover, it was discussed that fault diagnosis in electronic systems and networks of a vehicle is becoming an increasingly important factor.

For the fault detection, a data mining technique was applied and, from comparison to DNA, the sequencing nature of the data was considered as a feature that could be used as a classifier. The algorithm used here as a diagnosis tool was developed and coded using MathWorks MATLAB R2009a.

Signal processing methods where used to derive the density power spectrum of the binary indicator sequences from the messages sequences in each test. It can be concluded that the place of occurrence of the peaks in the density power spectrum are different among the scenarios and it can be used as the classification feature. According to this feature, three clusters were identified: the normal scenario 1 cluster, the normal scenario 2 cluster and the fault scenario cluster.

Furthermore, instead of just considering the peaks the whole sequence of the power spectrum density was considered. The density power spectrum is a discrete signal in frequency domain and as comparison tool for the classification. The available data was split into training sets and the test sets. The cross correlation of the training sets and the test sets was utilised as a classification feature. This hybrid classifier was able to distinguish the fault scenarios from the normal scenarios in 100% of cases, showing that this method is an effective classifier for these data sets.

This technique was developed on the CAN network of a vehicle, but application of it is not limited to vehicle industry. It can also be used on other backgrounds which use the CAN protocol or even on other networks in a vehicle such as LIN and MOST. Also new methods which adopt real time fault detection and on-board fault diagnosis and use the large amount of information available from the system network to pinpoint the cause of detected faults can use the technique introduced here. The real time fault detection system requires a generic electronic control unit (generic ECU) to monitor data at fast speeds in order to find faults. This requires algorithms with a small numbers of variables in order to respond quickly to faults. This extracted feature may be a useful variable for this purpose.

This method is most effective when a sequence of messages is expected in response to an event (such as network wake-up and shutdown), with the advantage of not needing to understand the underlying network functions. However, other types of faults may not be as clearly defined as this. Network fault detection and diagnosis is a complex issue with no single answer, requiring multiple approaches to categorise faults. This method can provide an additional tool to further research in this area.

REFERENCES

[1] Navet, N., Song, Y., Simonot-Lion, F., Wilwert, C., "Trends in Automotive Communication Systems," Proceedings of the IEEE, vol.93, no.6, June 2005, pp.1204-1223.

[2] Suwatthikul J., and McMurran, R., Peter Jones, R., "Automotive Network Diagnostic Systems," Industrial embedded systems, IES '06 international symposium, IEEE, 2006, pp.1-4, 1-4244-0777-X.

[3] Corno, F., Tosato, S., Gabrielli, P., "System-level analysis of fault effects in an automotive environment," *Defect and Fault Tolerance in VLSI Systems, 2003. Proceedings, 18th IEEE International Symposium on*, vol.3, no.5, Nov. 2003, pp.529-536.

[4] Johansson, K., Törngren, M., and Nielsen, L., "Handbook of Networked and Embedded Control Systems," D. Hristu-Varsakelis and W. S. Levine, Eds. Boston, MA: Birkhäuser, 2005.

[5] Arlat, J., Costes, A., Crouzet, Y., Laprie, J., and Powell, D., "Fault injection and dependability evaluation of fault-tolerant systems," IEEE Transactions on Computers, vol. 42, no.8, 1993, pp.913-923.

[6] NXP Semiconductors. "TJA1042 High-speed CAN transceiver with Standby mode". [Online]. (URL: http://www.nxp.com/documents/data_sheet/TJA1042.pdf) 2011. (Accessed 4th Aug 2011).

[7] stillerb infineon Company. "TLE6251-2G High Speed CAN-Transceiver with Wake and Failure Detection," [Online]. (URL: http://www.infineon.com/dgdl/TLE6251-2G_DS_rev10.pdf?folderId=db3a3043163797a6011666d32a0c0de1&fileId=db3a304320d39d5901215451b99c05dc) 2009. (Accessed 4th Au 2011)

[8] Simonot-Lion, F., "In-car embedded electronic architectures: how to ensure their safety," presented at the 5th IFAC Int. Conf. Fieldbus Systems and Their Applications, Aveiro, Portugal, 2003.

[9] Anastassiou, D., "Genomic signal processing," *Signal Processing Magazine, IEEE* , vol.18, no.4, Jul 2001, pp.8-20.

[10] MacQueen, J. B. "Some Methods for classification and Analysis of Multivariate Observations", Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability, Berkeley, University of California Press, vol.1, 1967, pp.281-297.

[11] Towell, G. G., and Shavlik, J. W,. "Knowledge-based artificial neural networks, Artificial Intelligence," vol.70, nos. 1-2, October 1994, pp.119-165. ISSN 0004-3702, DOI: 10.1016/0004-3702(94)90105-8.