Self-organizing maps for automatic fault detection in a vehicle cooling system

Magnus Svensson, Stefan Byttner and Thorsteinn Rögnvaldsson

N.B.: When citing this work, cite the original article.

Svensson M, Byttner S, Rögnvaldsson T. Self-organizing maps for automatic fault detection in a vehicle cooling system. In: 4th International IEEE Conference Intelligent Systems, 2008. IS '08. Piscataway, N.J.: IEEE; 2008. p. 24-8-24-12.

DOI: http://dx.doi.org/10.1109/IS.2008.4670481

Self-Organizing Maps for Automatic Fault Detection in a Vehicle Cooling System

Magnus Svensson, Stefan Byttner and Thorsteinn Rögnvaldsson

Abstract— **A telematic based system for enabling automatic fault detection of a population of vehicles is proposed. To avoid sending huge amounts of data over the telematics gateway, the idea is to use low-dimensional representations of sensor values in sub-systems in a vehicle. These low-dimensional representations are then compared between similar systems in a fleet. If a representation in a vehicle is found to deviate from the group of systems in the fleet, then the vehicle is labeled for diagnostics for that subsystem. The idea is demonstrated on the engine coolant system and it is shown how this self-organizing approach can detect varying levels of clogged radiator.**

Index Terms— **Anomaly detection, Diagnostics, Self-Organizing Maps**

I. Introduction

Fleet operators want to maximize the availability and use of their vehicles while minimizing the cost. Among other things, this means that the maintenance of a vehicle should be scheduled with as long intervals as possible. This, however, requires that service intervals are based on the health status of the vehicle (i.e. a corrective maintenance strategy), which can vary significantly between similar vehicles, and not on static mileage intervals or something similar (i.e. a preventive maintenance strategy). This means that the health status of each vehicle, and each subsystem on the vehicle, must be monitored so that, e.g., a check-engine light is lit when a vehicle needs service.

The traditional method for doing health status based fault detection and diagnosis on vehicles is to perform a large number of experiments in the lab, prior to production, and try to find characteristics that can be used to detect a coming (or existing) fault. This requires several engineering hours of work and use of expensive engine laboratories. The process is so costly that no manufacturer builds on-board error detection software for all possible errors that can occur on a vehicle. Only the most important malfunctions are included in the on-board-diagnostics tools. Also, the on-board computers are insufficient for running all diagnostics tools. Another problem is that many of the error detection systems on-board the vehicle are never triggered because the errors never occur, and at the same time is it too frequent that some unexpected part breaks on the vehicles once they are in production, e.g. a part that is not meeting the quality requirements. It would be terrific if the effort spent on error detection systems that are never triggered could instead be used on designing error detection systems for those errors that actually occur once production starts. This, however, requires the ability to predict which errors that can occur by continuously surveilling vehicles, looking for anything that behaves out of the ordinary (assuming that an error corresponds to an "out of the ordinary" behavior of the system). It requires a system that continuously collects data on the fleet vehicles and performs data mining in real time on this data, looking for interesting patterns, signatures of malfunctions.

This paper proposes exactly this; a telematic based fault detection scheme for enabling on-board fault detection by collecting data on a fleet of vehicles, performing data mining in real time and finding errors by looking for deviations from the norm, where the norm is defined dynamically from the population of vehicles, not in a static way from experiments in a lab. The basic idea is to create low-dimensional representations of subsystems in a vehicle, representations that capture the important correlations and relationships between sensor values and other signal for this subsystem, and to compare these low-dimensional representations between a large group (a fleet) of similar vehicles. If a low-dimensional representation (a model) in a vehicle is found to deviate compared to the group of models from a fleet of vehicles, then the vehicle is judged to need diagnostics for that subsystem. A deviation means that the relationships between sensor values (and other signals) in the subsystem for that particular vehicle are not what they should be, which indicates a possible fault. It is for now assumed that the deviation in the model cannot be attributed to e.g. a different driver behavior (this will be dealt with in future work). It is also possible to detect variations on a specific system on a single vehicle and follow the progress of this subsystem.

The representation is low-dimensional; hence it is possible to have the parameters for it transferred over a limited wireless communication channel to a communications center where the comparison of models is made. The alternative would be to transfer raw data that have been collected on the vehicle, which requires too much bandwidth.

Magnus Svensson is with Volvo Technology, 405 08 Göteborg, Sweden (e-mail: Magnus.Svensson@volvo.com).
Stefan Byttner is with the Intelligent System Laboratory, University of Halmstad, Box 823, 301 18 Halmstad, Sweden (e-mail: Stefan.Byttner@hh.se).
Thorsteinn Rögnvaldsson is with the Intelligent System Laboratory, University of Halmstad and with the Learning Systems Laboratory, AASS, Örebro University, 701 82 Örebro (e-mail: Thorsteinn.Rognvaldsson@hh.se).

978-1-4244-1739-1/08/$25.00 © 2008 IEEE

II. METHODOLOGY

There are basically two different approaches for doing system diagnostics (which includes error detection): Model Based Diagnostics (MBD) and Data Based Diagnostics (DBD). In the MBD case a model of a system is made from physical knowledge and this model is compared to the actual (real) system. If the behavior of the real system deviates from the model of the system, this indicates a fault (assuming that the physical model is accurate). In the DBD case the system is instead modeled from previous operation history and pattern recognition methods are then used to detect deviations, either by having examples in the history of faulty operation (which are recognized) or by defining patterns that have not been seen before as potential fault patterns. An example of DBD approach is presented by Martin and Marzi who use neural networks to detect faults in an engine cooling system [5].

A pro of the MBD approach is that it builds on an actual physical model and is thus easier to understand. It is also easier to fit, and should provide better results, since it has fewer free parameters; only the "necessary" parameters need to be tuned, which minimizes the number of experiments that need to be done. A con of the MBD approach is that the true physical process may be too complex to model accurately, given time constraints, so it is an approximate simplification that is used and the conclusions may therefore be wrong or the method may not be powerful enough (i.e. be able to detect all faults).

A pro of the DBD approach is that it uses the true empirical data and therefore focuses on what is actually observed, which may include error states which are too complex to simulate. A con is that it requires a lot of experimental data, preferably data that covers all possible error states (which of course is impossible to provide). Another con is that it is difficult to interpret.

The method described here is a DBD method that avoids the problem of having to create all error states in the laboratory in order to detect them [8]. Instead we aim to automatically find a deviating vehicle by doing continuous monitoring of it and data mining on the data collected on it. A faulty vehicle, or a vehicle that is about to get a fault, is detected by comparing its behavior against a fleet of similar vehicles; if it does not behave like the group then we flag it as potentially faulty. This can be used to start a diagnostics process, either remotely or on the vehicle. In this way we ensure that we only focus on those errors that actually occur on the vehicles.

The full method can be split into two steps:

- Look for interesting relationships among sensor values and control signals on the vehicle; encode these relationships in a suitable model. (This is done in a local self-organized way, i.e. on each individual vehicle.)
- Compare the models from the vehicles and look for deviations. (This is done in the back office with a global perspective, knowing the models from each vehicle.)

The first step requires a method for generating models on the vehicle and a measure of "interestingness" that does not build on a global perspective. The model describes the relationship between the on-vehicle signals. The second step requires a method (a metric) for comparing models and a method for generating a statistical model that can be used to flag abnormal models. The models are sent to the back-office with a dynamic update rate. Since it is only the model parameters that are sent the amount of data is low.

We have recently demonstrated how projective and linear models can be used for both the first step and the second step. For example, the sum square error can be used to rank linear models with different variable combinations without having to know the global behavior, and the distance between models can be measured using a Mahalanobis distance [1], [2]. Or, we can use explained variance for principal component based models and measure the angular distance between the resulting subspaces [6].

In this paper we extend our work to the nonlinear submanifold domain, presenting a method where submanifolds based on clustering approaches with topographic ordering are compared. The clustering method that has been used is the Self-Organizing Map (SOM), which is a neural network that configures a map of nodes to a data set [4], [3]. A SOM model is used to represent the data on each vehicle and we introduce a measure $D_{A,B}$ for the distance between two SOM models A and B, and demonstrate that this distance can be used to group vehicles into different fault groups. The method is demonstrated on an engine cooling system with injected faults.

The proposed method for comparing systems works as follows:

1. A SOM model is fitted to the data from each vehicle. These SOM models have the same topology and the same number of nodes (N). (In the demonstration case we use different fault states on the same vehicle and not different vehicles.)
2. The Euclidean distances d_{ij}^{AB} between all nodes in two SOM models, denoted by A and B, are computed.
3. The nodes in the two SOM models are then paired so that the summed distance between the two SOM models is minimized. This minimal distance is used for $D_{A,B}$, the distance between SOM model A and SOM model B. The minimization is done with a random search method. The nodes in the two SOM models are then paired so that the summed distance between the two SOM models is minimized. This minimal distance is used for $D_{A,B}$, the distance between SOM model A and SOM model B. The minimization is done with a random search method.

We introduce the assignment matrix $\mathbf{S}^{AB}$ with elements

$$S_{ij}^{AB} = \begin{cases} 1\, if\ node\ i\ in\ A\, is\ connected\ to\ node\ j\ in\ B \\ 0\, otherwise \end{cases} \tag{1}$$

The matrix $\mathbf{S}$ is such that there is only one nonzero element in each row and column, i.e. a node in one SOM model can only be connected to one node in the other SOM model. The distance between the two SOM models A and B can then be written as

$$D\left(\mathbf{S}^{AB}\right)=\sum_{i=1}^{N}\sum_{j=1}^{N} s_{ij}^{AB} d_{ij}^{AB} \qquad (2)$$

The distance between the two SOM models is then defined as the minimum of this, i.e. $D_{A,B} = min\, D\left(\mathbf{S}^{AB}\right)$. The lower the min distance is, the more similar are the two SOM models. The minimization of the distance is done iteratively and approximately with a random search method described below. The number of possible $\mathbf{S}$ matrices becomes $N!$ where N is the number of nodes in the SOM model.

A. Initialization

Eight initialization matrices are randomly selected. (The matrices are $\mathbf{S}$ matrices.)

B. Iterations

The two $\mathbf{S}$ matrices that give the shortest distances $D_{A,B}$ are saved for the next iteration and parts of the best and the second best $\mathbf{S}$ matrices are used in three new matrices. Three new randomly generated matrices are generated at each iteration.

C. Selection

The shortest distance between the nodes is selected $min\, D\left(\mathbf{S}^{AB}\right)$. Figure 1 show the minimum and maximum distances during 100 iterations for two SOM-model nodes.

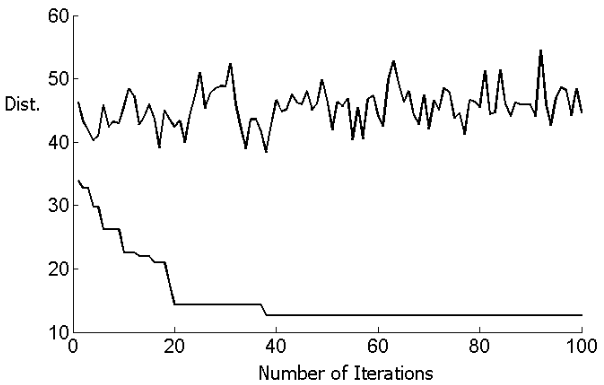

Fig. 1. Number of iterations and total distance between paired nodes of two SOM models, top line shows the maximum distance and bottom line shows the minimum distance at each iteration.

III. DATA COLLECTION

We use data from a bus in inter-city traffic with different levels of clogged radiator to demonstrate the method. The clogging levels are 0%, 25% and 50%.
The anomaly detection method was applied on the engine cooling system, where the most important parts are:

- Engine block - where the coolant mixture heats up
- Radiator - where the coolant mixture cools down
- Fan - increase cooling of coolant mixture in the radiator
- Thermostat valve - controls the amount of coolant mixture through the radiator

The main purpose of the cooling system is to keep the coolant temperature at a predefined constant level. An accurately performing cooling system decreases fuel consumption, emissions and wear of vehicle components. The amount of heat that is removed from the engine by the coolant mixture is approximately 30% of the energy from the fuel [7].

The bus was driven approximately 35 minutes in inter-city traffic, including a steep hill, on each soil level and the following signals were used for analysis:

- Engine Fan Speed
- Turbo Pressure
- Coolant Temperature
- Engine Speed
- Engine Load
- Oil Temperature
- Pedal Position
- Road Speed

All signals were normalized, i.e. rescaled to zero mean and unit standard deviation, before any analysis was done. Each soil level measurement was divided into two data sets, i.e. there are six data sets in total.

IV. SELECTION OF SIGNALS

In this paper the self-organizing maps are based on two, three and four signals available on the vehicle. There are many signals that affect the cooling system and we choose the signals on the vehicle that represent the cooling system data as well as possible. This is done by measuring the distance to the closest node from each sample. This should preferable be done when the vehicle is new. In the column "Data sample to node" in table I to table III are the values typed for the different signal combinations.

By selecting signals only depending on how well the nodes represents the data may not always be the best method to extract the relevant signals to use. If the selected signals are constants the distance to node will be low although the selected signals might not be useful for anomaly detection of the engine cooling system.

V. RESULTS

A self-organizing map with $5x5$ nodes was fitted to the data and the total minimal Euclidean distance between the nodes placements have been calculated by a random search algorithm. Fig. 2 to fig. 7 show data with corresponding node placement for the three different soil levels.
Table I to table III show the different mean distances between the nodes. The tables also show the "Data sample to node" distance that indicates which signals are the most suitable for clogged radiator detection.

TABLE I

DISTANCES FOR DIFFERENT SOIL LEVELS, FOUR SELECTED SIGNALS

Selected Signals	Distances: Data sample to node	Soil 0% to Soil 0%	Soil 0% to Soil 25%	Soil 0% to Soil 50%	Soil 25% to Soil 50%
Fan Speed, Boost Press, Coolant Temp, Engine Speed	0,34	1,61	1,50	1,53	1,51
Fan Speed, Boost Press, Coolant Temp, Road Speed	0,25	1,72	1,87	1,88	1,81
Fan Speed, Coolant Temp, Pedal Pos, Road Speed	0,37	1,81	1,67	1,74	1,77
Fan Speed, Boost Pres, Coolant Temp, Engine Load	0,43	1,46	1,58	1,48	1,47

TABLE II

DISTANCES FOR DIFFERENT SOIL LEVELS, THREE SELECTED SIGNALS

Selected Signals	Distances: Data sample to node	Soil 0% to Soil 0%	Soil 0% to Soil 25%	Soil 0% to Soil 50%	Soil 25% to Soil 50%
Fan Speed, Coolant Temp, Engine Load	0,12	1,10	1,28	1,24	1,21
Boost Press, Coolant Temp, Road Speed	0,39	1,53	1,44	1,34	1,34
Engine Speed, Pedal Pos, Road Speed	0,35	1,33	1,37	1,34	1,23
Engine Load, Oil Temp, Pedal Pos	0,56	1,78	1,60	1,62	1,60
Coolant Temp, Engine Speed, Road Speed	0,52	1,37	1,41	1,43	1,48
Fan Speed, Coolant Temp, Engine Speed	0,22	1,04	1,26	1,21	1,28

TABLE III

DISTANCES FOR DIFFERENT SOIL LEVELS, TWO SELECTED SIGNALS

Selected Signals	Distances: Data sample to node	Soil 0% to Soil 0%	Soil 0% to Soil 25%	Soil 0% to Soil 50%	Soil 25% to Soil 50%
Fan Speed, Boost Pres	0,06	0,71	0,82	0,93	0,93
Fan Speed, Coolant Temp	0,13	1,01	1,01	1,08	0,95
Fan Speed, Engine Speed	0,17	0,96	0,94	0,99	0,97
Fan Speed, Road Speed	0,09	0,96	1,00	0,97	1,05
Boost Pres, Coolant Temp	0,15	1,16	1,09	1,12	1,06
Coolant Temp, Engine Load	0,29	0,91	0,98	1,07	1,06
Coolant Temp, Engine Speed	0,09	1,16	1,10	1,14	1,11

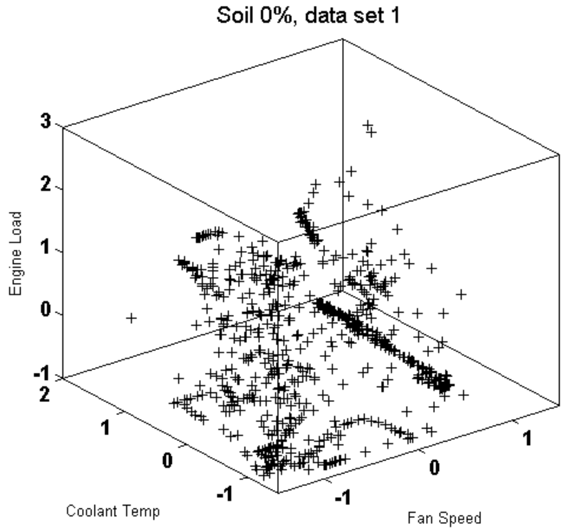

Fig. 2. Data samples from 0% soli level with Fan Speed, Coolant Temperature and Engine Load

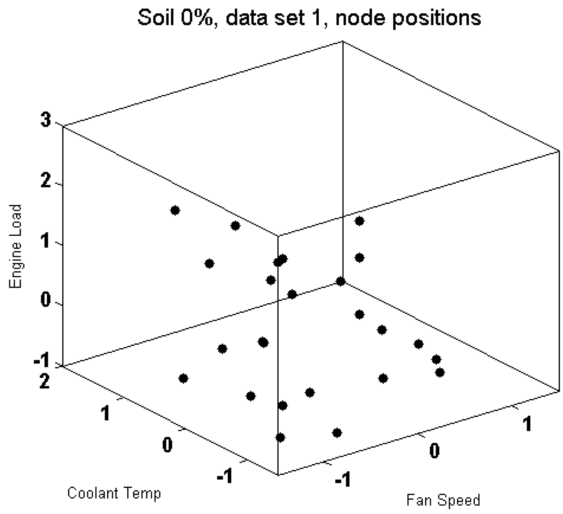

Fig. 3. Node placement, 0% soli level with Fan Speed, Coolant Temperature and Engine Load

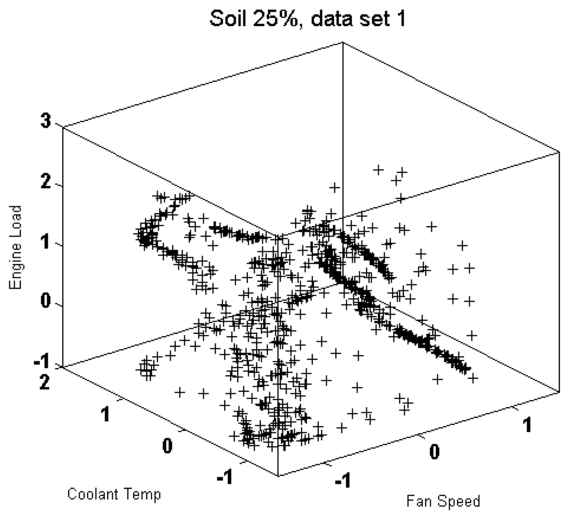

Fig. 4. Data samples from 25% soli level with Fan Speed, Coolant Temperature and Engine Load

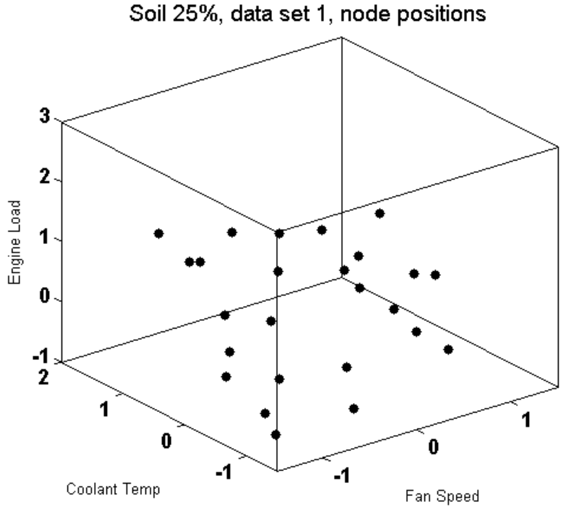

Fig. 5. Node placement, 25% soli level with Fan Speed, Coolant Temperature and Engine Load

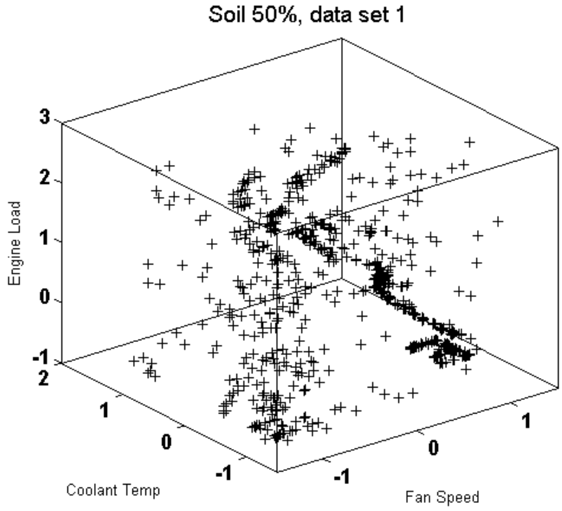

Fig. 6. Data samples from 50% soli level with Fan Speed, Coolant Temperature and Engine Load

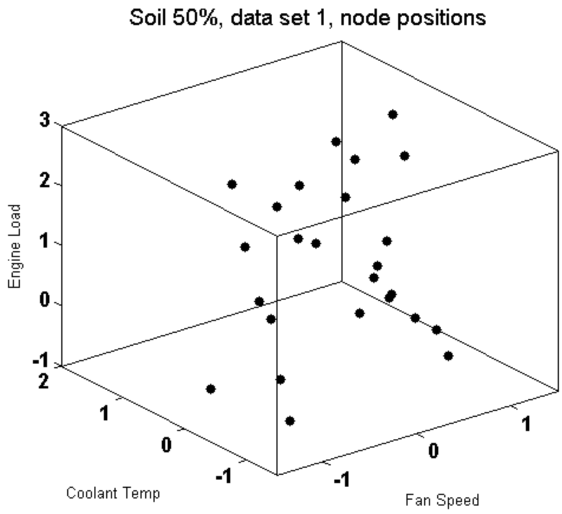

Fig. 7. Node placement, 50% soli level with Fan Speed, Coolant Temperature and Engine Load

VI. SUMMARY AND CONCLUSION

The need for improved diagnostic functions of vehicles is high and the need increases even further due to the increased complexity and variants of vehicles. The method described utilizes an on-vehicle model, telematics and an off-vehicle server to detect if the cooling system on a bus in a fleet of buses works normally compared with the rest of the vehicles in the fleet. The proposed method can be described in five steps:

- Collect selected data from the vehicle
- Search for signals on the vehicle that feed a SOM
- Send SOM properties to back-office
- Compare SOM properties from different comparable vehicles
- Evaluation; find a vehicle or vehicles with deviating SOM parameters among the vehicles

This method has been tested on real data on an engine cooling system to detect soil anomaly. The results show that by measuring the distance difference between the node positions from different self-organizing, it is possible to detect changes on clog levels of the radiator. Table I to table II show that there is a measurable distance difference between different soil levels of the radiator. Fig. 2 to fig. 7 shows the differences on the data sets with corresponding nodes. It is apparent from the figures that there is a difference between the data with 50% clogged radiator and the other soil levels (25% soil and no soil). However, there is a measurable difference on the data from the two lowest soil levels. To verify the method it is also required that a fleet of test-buses in traffic is used. Moreover, to be able to identify the exact root cause of the problem on the cooling system (e.g. soiled radiator or leakage); further analyses have to be performed either at a workshop or automatically with separate downloadable or on-vehicle diagnostic functions.

The results show that it is possible to use a self-organizing map to detect if a vehicle cooling system behaves differently with a clogged radiator.

REFERENCES

[1] S. Byttner, T. Rögnvaldsson and M. Svensson. Modeling for vehicle fleet remote diagnostics. Technical paper 2007-01-4154, Society of Automotive Engineers (SAE), 2007.

[2] S. Byttner, T. Rögnvaldsson and M. Svensson. Self-organized modeling for vehicle fleet based fault detection. Technical paper 2008-01-1297, Society of Automotive Engineers (SAE), 2008.

[3] T. Kohonen. The self-organizing maps. *Proceedings of the IEEE,* 78(9):1461-1480, 1990

[4] T. Kohonen. *Self-Organizing Maps.* Springer-Verlag, Berlin, 3 edition, 2001.

[5] K. F. Martin and M.H. Marzi. Diagnostics of a coolant system via neural networks. *Journal of Systems and Control Engineering, 213:229-242,1999*

[6] Georg Panholzer. Identifying deviating systems with unsupervised learning. Technical Report IDE-0801, Halmstad University, 2008.

[7] J. A. Twiddle and N. B. Jones. Fuzzy model-based condition monitoring and fault diagnosis of a diesel engine cooling system. *Journal of Systems and Control Engineering,* 216:215-224,224, 2002.

[8] G. Vachkov. Intelligent data analysis for performance evaluation and fault diagnosis in complex systems. *IEEE International conference on fuzzy systems*, pages 6322-6329, July 2006