2006 IEEE International Conference on
Systems, Man, and Cybernetics
October 8-11, 2006, Taipei, Taiwan

Application of Signal Analysis and Data-driven Approaches to Fault Detection and Diagnosis in Automotive Engines

Setu Madhavi Namburu, Shunsuke Chigusa, Liu Qiao, Mohammad Azam, and Krishna R. Pattipati

Abstract—**The modern era of sophisticated automobiles is necessitating the development of generic and automated embedded fault diagnosis tools. Future vehicles are expected to contain more than one hundred complex Electronic Control Units (ECUs) and data acquisition systems to control and monitor large number of system variables in real-time. There exists an abundant amount of literature on fault detection and diagnosis (FDD). However, these techniques are developed in isolation. In order to solve the problem of FDD in complex systems, such as modern vehicles, a hybrid methodology combining different techniques is needed. Here, we apply an approach based on signal analysis that combines various signal processing and statistical learning techniques for real-time FDD in automotive engines. The data under several scenarios is collected from an engine model running in a real-time simulator and controlled by an ECU.**

Key Words: Fault detection and diagnosis, data-driven approach, signal analysis, partial least squares, wavelets

I. INTRODUCTION

On Board Diagnosis (OBDII) regulations for reduced fuel consumption and emissions are the major drive in the development of more sophisticated automobiles. As the complexity of systems increase, detecting and diagnosing the faulty components also becomes a very challenging task. FDD has evolved upon three major paradigms, viz., model-based, data-driven and knowledge-based approaches. Even though FDD has been an active research in many domains for the past four decades, it is still an open area of research calling for the development of automated diagnostic procedures for providing "smart services". Luo et al. [14] proposed a generic intelligent diagnostics process that requires analytical and qualitative knowledge [18] about the system. Model-based approach is already used in automotive industry for control development, as dynamic models of the vehicle sub-systems are usually available. However, extensive amount of system knowledge and expertise are needed to derive dynamic relations for FDD. Incorporating the model-based approach for on-board diagnosis is questionable as the modern vehicles are using complex system dynamics and control loops. Presently, the vehicle diagnosis is performed by service technicians by referring to service manuals based on the diagnostic codes generated by the OBD system. Significant amount of research is ongoing for developing user-friendly and generic embedded diagnostic modules in modern vehicles for on-board diagnosis. The future vehicles are expected to contain more than 100 ECUs and sophisticated data acquisition tools and sensors to control and monitor several system variables continuously. Modern automotive industries are also producing vehicles that have mobile agents for remote communication with service centers. This gives a scope for collecting large amount of monitored data and analyzing it continuously, making the system a data rich environment.

Data-driven approaches have proven to be very promising in data rich environments, such as aerospace [1][2], power systems [3], chemical industries, and HVAC systems [4][5]. Signal processing tools, neural networks and statistical learning techniques are some of the examples of data-driven techniques. Some researchers applied signal-processing techniques for automotive fault diagnosis [7][8][9][10]. It was observed from the previous research that a single approach is not adequate to solve the FDD problem in non-linear, complex systems, such as automotive engines.

Here, we apply a mode-invariant signal analysis-based data-driven procedure that seamlessly employs combination of several signal-processing and statistical learning techniques for detecting and diagnosing faults in automotive engines. This procedure requires only the monitored signals without the need for any system models. We demonstrate the procedure on an engine model running in the real-time simulator, CRAMAS® (ComputeR Aided Multi-Analysis System) and controlled by a prototype ECU. The procedure is also validated under several operating conditions of the engine.

The rest of the paper is organized as follows. Section II provides a brief description of the engine model under consideration. In Section III, we describe the proposed generic fault detection and diagnostic procedure for automotive engines. Section IV presents the detailed experimental FDD results on the engine model considered. We conclude the paper in Section V with future research directions.

Manuscript received March 30, 2006. This work was supported by Toyota Motor Engineering and Manufacturing North America.

S. M. Namburu, S. Chigusa, and L. Qiao are with the Toyota Motor Engineering and Manufacturing North America, 1555 Woodridge, Ann Arbor, MI 48105 USA (e-mail: setumadhavi.namburu@tema.toyota.com).

M. Azam, and K. R. Pattipati are with the Electrical and Computer Engineering Department, University of Connecticut, 371 Fairfield Road, Storrs, CT 06269 USA (e-mail: krishna@engr.uconn.edu).

II. Engine Model Description

The SIMULINK® engine model considered in our experiments is an approximate model of Toyota Camry 554N engine. A detailed description of this model is provided in [16]. This model consists of five primary subsystems, namely, air dynamics, fuel dynamics, torque generation, rotational dynamics, and the exhaust system. The model is simulated in a custom-built CRAMAS® and controlled by a prototype ECU to achieve real-time performance. CRAMAS® is a high-speed, multi-purpose expandable system for vehicle control development. Here, we utilized it for vehicle diagnostic development. Fig. 1. depicts the CRAMAS® set-up for engine simulation at the Toyota Technical Center research facility in Cambridge, MA. This platform provides a means for rapid prototyping, and Hardware-In the-Loop simulations. We simulated the engine model in CRAMAS®, and collected the experimental data under several scenarios for our analysis.

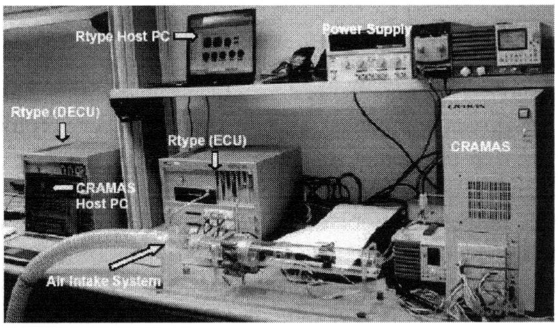

Fig. 1. CRAMAS® set-up for engine simulation

A. Engine Monitored Variables and Fault Universe

This engine system is a highly non-linear, complex model. There are more than 100 variables calculated continuously in the model. Deriving dynamic relations to apply model-based approach requires a significant amount of expertise. Another challenging issue is that the engines operate at different operating conditions (such as various speeds, water temperature, pedal angle, etc.) depending on the load torque. Consequently, the development of a mode-invariant diagnostic procedure is a cumbersome task. Here, we explored eight engine faults [16] and their effect on seven monitored variables. The measurement data is collected at three operating conditions, namely, three pedal positions: 15, 18 and 20 degrees respectively. In each operating condition, all eight faults are simulated at ten severity levels: 6% - 15% with a step size of 1%. Table I presents the list of monitored variables and the fault universe considered in our experiments.

III. Fault Detection and Diagnosis Approach Overview

The proposed FDD procedure for automotive engines is depicted in Fig. 2. It consists of an off-line training module and an on-line testing module. The following sub-sections describe the process in detail.

A. Off-line FDD Module

In the off-line phase, the experimental data is collected from the Simulink model running in CRAMAS® at several operating conditions and fault scenarios. Here, we performed our analysis on steady-state data. The following sub-sections describe each block of the off-line module in detail.

TABLE I
Engine Fault Universe and Monitored Variables

Fault List		Monitored Variables	
Name[1]	Notation	Name[2](Units)	Notation
Air Flow Sensor Fault	AFS	Air Pressure (KPa)	P_m
Leakage in Air Intake System	AIS_leak	Amount of Fuel Injected (mg)	FC
Blockage of Air Filter	AF_blockage	Air to Fuel Ratio	A/F
Throttle Angle Sensor Fault	TAS	Engine Speed (rpm)	NE
Less Fuel Injection	LFI	Vehicle Speed (km/h)	SPD
Added Engine Friction	AEF	Throttle Angle (deg)	TA
Air/Fuel Sensor Fault	AFuel_S	Air Flow Meter (Volts)	VG2
Engine Speed Sensor Fault	ESS		

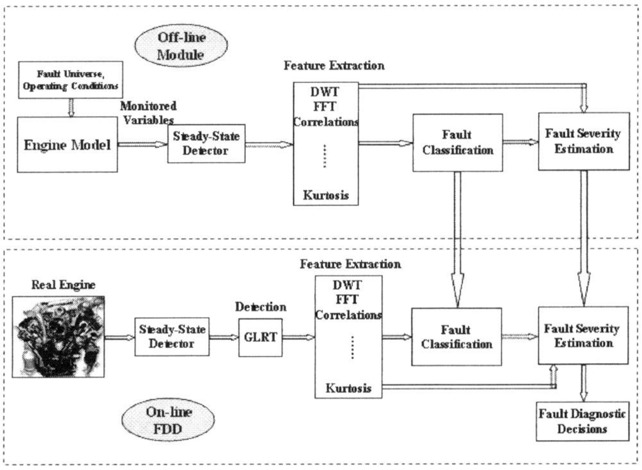

Fig. 2. Generic Fault Detection and Diagnosis Scheme for Automotive Engines

1) Steady-State Detector: Since the engine's operating region varies depending on the load torque, it takes considerable amount of time for the monitored variables to achieve steady-state. As shown in the diagram, a steady-state detector is used to detect the steady-state condition by estimating the time derivatives of the monitored variables over a sliding window.

2) Feature Extraction: In the next step, mode-invariant features are extracted via signal processing techniques such as wavelets, FFT, correlations, kurtosis, etc. As mentioned earlier, with the wide operating regions of the engine,

[1] These are denoted as faults numbered from 1 through 8 in later sections
[2] These are denoted as sensors numbered from 1 through 7 in later sections.

extracting the features from data that are insensitive to operating modes is a very challenging task. Wavelets have proven to be very useful feature extractors in recovering mode-invariant, fine-grained features from the data [1][2][7][8]. Here, we employed wavelet-based feature extraction on the seven monitored variables.

3) Fault Classification: In the fault classification block, two methods are used to classify faults into pre-determined classes. In the first method, tests are performed on the extracted wavelet feature set based on trending, thresholding etc. A D-matrix (Diagnostic-matrix) is generated from the test outcomes. This is imported by the on-line module for real-time inference. The following section describes the procedure for testing the wavelet coefficients and generating the D-matrix.

a) Test Design for Classifying Faults: The challenging task in test design is the extraction of features from sensors that are insensitive to modes of operation of the engine. For the test design, we consider only those features from the monitored variable m, which satisfy the following criterion:

$$\theta\left(S_m(k/O_i)\right) \approx \theta\left(S_m(k/O_j)\right) \tag{1}$$

In the above equation O_i refers to the engine operating condition, such as pedal angle position, engine speed, etc. The signal S_m is the value of monitored variable m. These features are approximate wavelet coefficients that are obtained from raw sensor data. Trend testing is performed on the wavelet coefficients to classify faults into different categories.

The following test procedure is implemented to classify the faults [1]

Define $A_{ij}(S_m)$, $i = 1,2,3,\ldots\ldots$, and $j = 1,2,3,\ldots$ as the i^{th} approximate coefficient at level j of the wavelet transform of the time series signal from the monitored variable m. The trend function at the level j is defined as

$$T_{d_{m_j}} = \bigcup_i \operatorname{sgn}\left(A_i(S_m) - A_{i-1}(S_m)\right);\ i = 1,2,3,\ldots \tag{2}$$

Two trend tests are used, namely a positive trend test $\left(T_{pos}(m_j)\right)$ and a negative trend test $\left(T_{neg}(m_j)\right)$ for the isolation process. The test outcomes are evaluated as follows

$$T_{pos}(m_j) = \begin{cases} 1 \text{ if } T_{d_{m_j}} = \{1\} \\ 0 \text{ otherwise} \end{cases} \tag{3}$$

$$T_{neg}(m_j) = \begin{cases} 1 \text{ if } T_{d_{m_j}} = \{-1\} \\ 0 \text{ otherwise} \end{cases} \tag{4}$$

A fault codebook (D-matrix) is prepared for all the faults based on these tests. Then, these tests are applied on simulated data in the testing phase and the binary codes are generated. Hamming distance criteria is used to measure the distance between the code generated from the test pattern and each entry of the code book. The test pattern is classified as the fault that has the nearest distance with the entry in the codebook.

In the second approach, the extracted wavelet coefficients are used to train the Partial Least Squares (PLS) network for fault classification. The description of PLS is provided in Appendix. The trained PLS weights are imported by the on-line module for real-time classification.

4) Fault Severity Estimation: In the next step, the fault severities are estimated via PLS regression. The trained weights are imported by the on-line estimation block for real-time diagnosis.

B. On-line FDD Module

In the on-line FDD module, the data is monitored from a real-engine. Once steady-state is detected, the Generalized Likelihood Ratio Test (GLRT) is used to detect faults. The feature extraction block, classification block, and estimation block perform the same operations as in the off-line module. The diagnostic decision block gives the details about the fault and its severity level. Here, we used this phase for testing on the simulated data.

IV. Experimental Results

As explained in Section II, the collected data from three operating conditions and eight fault scenarios with different severity levels constitute a total of 240 patterns. The data collection sampling time is 5msec. 2000 time samples are collected from each sensor for all the scenarios for the sake of consistency. The faults are inserted at 500th sampling point in each run. The FDD of the engine consists of detection, isolation, and severity estimation for these eight faults. The following subsections describe the experimental results in each phase.

A. Fault Detection

The faults are detected via a GLR test on the 7 monitored variables. Fig. 3. presents the GLRT response on air pressure sensor to blockage of air filter fault at PA: 15 degrees. It is observed that 100% fault detection is achieved via GLRT, irrespective of the operating mode.

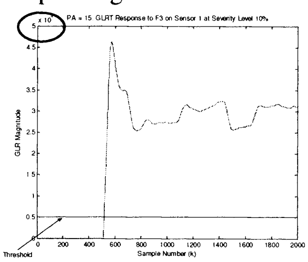

Fig. 3. GLRT response on pressure sensor signal to 10% AF_blockage fault

B. Feature Extraction

The data matrix has the dimension 240x7x2000 where 240 is the number of patterns from eight faults, 7 is the number of monitored variables and 2000 is the length of time series. The mode-invariant features from the seven monitored variables are extracted using wavelets, for each pattern. Daubechies (DB) 1, Daubechies 2 mother wavelets are used for wavelet transformation and the level 9 and 10 approximate coefficients are of more interest in this case study.

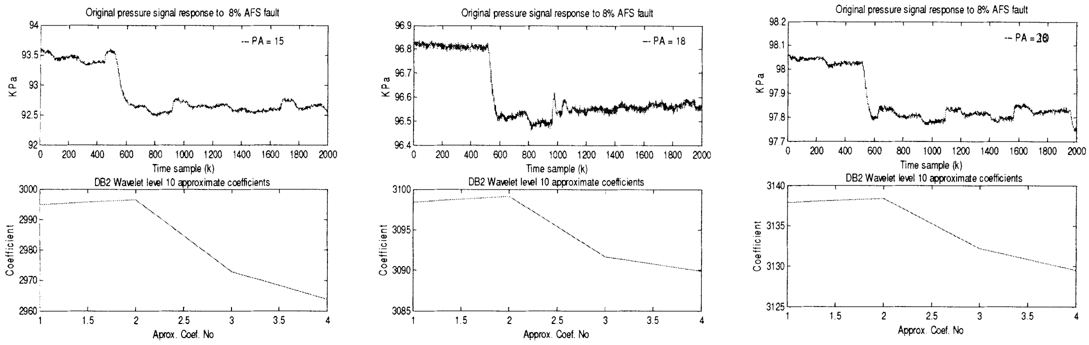

Fig. 4. Original pressure signal and DB2 wavelet level 10 approximate coefficients response to 8% AFS fault at $k = 500$.

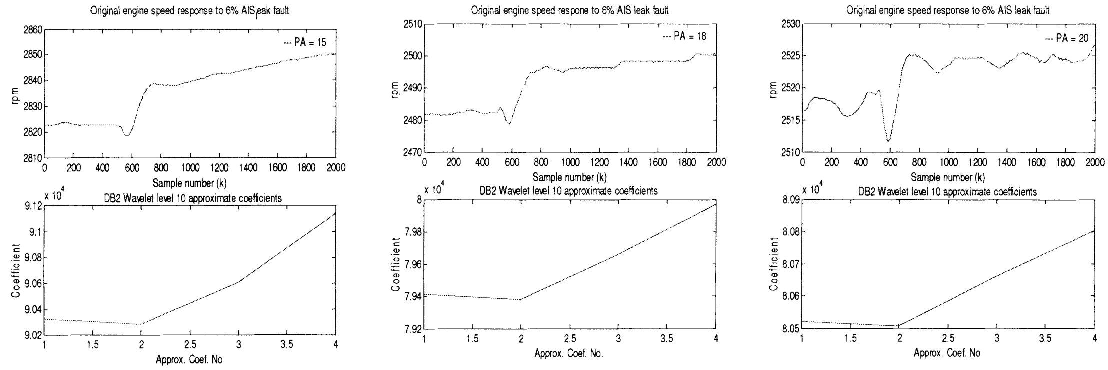

Fig.5. Original engine speed and DB2 wavelet level 10 approximate coefficients response to 6% AIS_leak fault at $k = 500$.

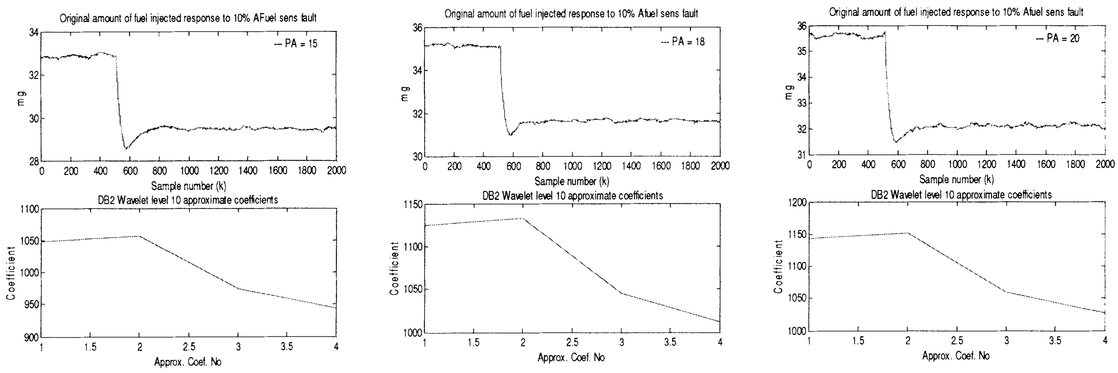

Fig. 6. Amount of fuel injected and DB2 wavelet level 10 approximate coefficients response to 10% AFuel_sens fault at $k = 500$.

TABLE II
D-MATRIX FOR ISOLATING ENGINE FAULTS VIA TREND-TESTING

Fault\Test	S1T1	S1T2	S2T1	S2T2	S3T1	S3T2	S4T1	S4T2	S5T1	S5T2	S6T1	S6T2	S7T1	S7T2
AFS (F1)	1	0	0	1	0	0	0	0	0	0	0	0	0	1
AIS_leak (F2)	0	1	0	0	0	0	0	1	0	1	0	0	1	0
AF_block (F3)	1	0	1	0	0	0	0	0	0	0	0	0	1	0
TAS (F4)	1	0	1	0	0	0	1	0	0	0	1	0	1	0
LFI (F5)	0	0	0	1	0	0	0	0	1	0	0	0	0	0
AEF (F6)	0	0	0	0	0	0	1	0	1	0	0	0	1	0
AFuel_S (F7)	0	0	1	0	0	1	1	0	1	0	0	0	1	0
ESS (F8)	1	0	0	0	0	0	1	0	1	0	1	0	1	0

TABLE III
CONFUSION MATRIX OF EIGHT ENGINE FAULTS VIA TREND TESTING AND PLS

	Via Trend Testing								Via PLS							
	F1	F2	F3	F4	F5	F6	F7	F8	F1	F2	F3	F4	F5	F6	F7	F8
F1	86.67	0	0	0	0	0	0	13.33	95.83	0	0	0	0	4.17	0	0
F2	0	100	0	0	0	0	0	0	0	100	0	0	0	0	0	0
F3	0	0	80	20	0	0	0	0	0	0	99.58	0	0	0.42	0	0
F4	0	0	13.33	86.67	0	0	0	0	0	0	0	100	0	0	0	0
F5	10	0	0	0	90	0	0	0	0	0	0	0	100	0	0	0
F6	0	0	0	3.33	0	73.33	10	13.33	0	0	0	0	0	100	0	0
F7	0	0	0	10	0	0	90	0	0	0	0	0	0	0	100	0
F8	0	0	0	3.33	0	0	0	96.67	0	0.83	1.25	0	0	0	0	97.92

Figs. 4 - 6 show some of the mode-invariant features to pedal positions from some of the faults. The positive or negative trends are observed based on the effect of fault on the signal for longer duration irrespective of its magnitude.

C Fault Classification Results

Here, two approaches are applied to classify faults. In the first approach, trend testing is performed on 7 monitored variables. Table II presents the D-matrix developed for isolating the eight faults. The rows in the D-matrix represent the number of faults. The columns represent the tests. Each element in the matrix represents a test outcome: either pass or fail. Here *SiT1* represents the negative trend test on sensor *i*, and *SiT2* represents the positive trend test on sensor *i*.

In the second approach, the extracted DB2L10 approximate wavelet coefficients are fed to a PLS classifier as a feature set. The data is divided into 60% training and 40% testing, and PLS is applied to train and test the patterns. The procedure is repeated 20 times.

The accuracies obtained are as follows:

Total classification accuracy via trend testing on wavelet coefficients: **87.92%**

Total classification accuracy via PLS on wavelet coefficients (mean over 20 runs): **99.17%**

The D-matrix obtained is based on hard trend testing on the monitored data. The trends are ignored for some of the signals that are affected by faults for only short duration. We can clearly see the drastic improvement of classification accuracy by applying classification technique on more significant features extracted via wavelets. Table III presents the classification accuracies obtained via both approaches in the form of a confusion matrix [5]. As can be seen from the table, the classification accuracy for each fault is above 95% via PLS.

D Fault Severity Estimation via PLS

After a fault is classified into one of the classes, the fault severity is estimated via the PLS estimation block. One PLS network is trained for each fault separately. For each fault, different wavelet coefficients are selected as features for estimation depending on the error. Table IV presents the average estimation errors over 20 runs for each fault on the test set, and the wavelet coefficients used. It can be observed that the average errors are within 10%, except for fault 6.

TABLE IV
FAULT SEVERITY ESTIMATION AVERAGE % ERRORS OVER 20 RUNS VIA PLS

Fault Name	Wavelet Details	Average Errors (%)
AFS (F1)	DB2L10	2.98
AIS_leak (F2)	DB2L10	14.58
AF_blockage (F3)	DB1L10	6.06
TAS (F4)	DB1L10	9.94
LFI (F5)	DB2L10	11.46
AEF (F6)	DB1L10	26.11
AFuel_S (F7)	DB1L9	6.74
ESS (F8)	DB2L10	3.06

V. CONCLUSION

In this paper, we applied a signal analysis-based data-driven approach for real-time diagnosis of automotive engines. Eight engine faults are detected and diagnosed via the proposed FDD approach. GLRT is used for fault detection, and for classification we employed two approaches viz., trend testing on extracted wavelet features and application of PLS. Finally the fault severities are estimated using PLS.

Our future research would involve testing the proposed approach on a real-engine. We plan to develop a demo system and investigate intelligent testing procedures that are applicable to transient data also. We also plan to combine data-driven and knowledge-based approaches.

APPENDIX

Partial least squares [19] is successfully applied in supervised classification tasks for different applications [4][5][18]. It is a good alternative technique to multi linear regression due to its robustness. It is a dimensionality reduction technique for maximizing the covariance between the $n \times m$ independent training data matrix X, and the $n \times d$ dependent matrix Y (corresponding to d fault classes) for each component of the reduced space. Here, n represents the number of patterns from d classes, and m is the number of features (wavelet coefficients). It builds a regression model between X and Y. PLS can be used for both classification and estimation tasks (e.g., to estimate fault severity levels). For classification, while forming the data sets, the fault class label, say j, corresponding to the pattern in a row of X is represented by 1 in the corresponding row of column j of Y. For estimation, the values in Y represent the fault severities corresponding to the pattern in X.

PLS, a variant of conjugate gradient method, generates uncorrelated latent variables, which are linear combinations of the original features. The basic idea is to select the weights of the linear combination to be proportional to the covariance between the features and pattern classes. Once the latent variables are extracted, a least squares regression is performed to estimate the fault class. Both matrices X and Y are decomposed into a number of components, which is known as the model reduction order, plus residuals. Each component captures certain amount of variation in the data. The reduction order is determined by cross-validation. The PLS decompositions are given by

$$\begin{aligned} X &= \sum_{i=1}^{k} \underline{t}_i \underline{p}_i^T + E = TP^T + E \\ Y &= \sum_{i=i}^{k} b_i \underline{t}_i \underline{q}_i^T + F = UQ^T + F \\ U &= TDiag(b_i) \in R^{n\times k} \end{aligned} \tag{A1}$$

Here, $T \in R^{n\times k}$ and $U \in R^{n\times k}$ are score matrices (latent vectors); $P \in R^{m\times k}$ and $Q \in R^{d\times k}$ are loading matrices; $E \in R^{n\times m}$ and $F \in R^{n\times d}$ are residual matrices, and k is the model reduction order or number of PLS components retained.

PLS can be viewed as a two-phase optimization problem. For simplicity, assume that the response matrix Y is a single column, $\underline{y} \in R^n$. In the first phase, for a given latent vector $\underline{t} \in R^n$, PLS seeks to find a rank 1-matrix $\underline{t}\,\underline{p}^T$ that is closest to X in Forbenius norm, i.e., $\min_{\underline{p}} \left\| X - \underline{t}\,\underline{p}^T \right\|_F$. The solution is given by: $\underline{p} = \frac{X^T \underline{t}}{\underline{t}^T \underline{t}}$. In the second phase, PLS seeks to find $\underline{t}$ and a weight vector $\underline{w}$ such that the covariance between $\underline{t}$ and $\underline{y}$ is maximized:

$$\begin{aligned} &\max_{\underline{t},\underline{w}} \underline{t}^T \underline{y} \\ &\text{s.t. } \underline{t} = X\underline{w} \quad \& \quad \underline{w}^T \underline{w} = 1 \end{aligned} \tag{A2}$$

The result is:

$$\underline{w} = \frac{X^T \underline{y}}{\left\| X^T \underline{y} \right\|_2} \propto \underline{p} \text{ and } \underline{t} = \frac{XX^T \underline{y}}{\left\| X^T \underline{y} \right\|_2} \tag{A3}$$

Since $\underline{t}$ and $\underline{w}$ (or $\underline{p}$) depend on each other, iteration is required. The score and loading vectors are determined using the nonlinear iterative partial least squares algorithm [19].

References

[1] Azam, M., K. Pattipati, J. Allanach, S. Poll, and A. Patterson-Hine, "In-flight Fault Detection and Isolation in Aircraft Flight Control Systems", *Proceedings of the IEEE Aerospace Conference,* Big Sky, Montana, March 2005.

[2] Azam, M., D. Pham, F. Tu, K. Pattipati, A. Patterson-Hine, and L. Wang, "Fault Detection and Isolation in the Non-Toxic Orbital Maneuvering System and the Reaction Control System", *Proceedings of the IEEE Aerospace Conference,* Big Sky, Montana, March 2004.

[3] Azam, M., F. Tu, K. Pattipati, and R. Karanam, "A Dependency Model-based Approach for Identifying and Evaluating Power Quality Problems", IEEE *Transactions on Power Delivery,* Vol. 19, No.3, pp. 1154 -1166, July 2004.

[4] Namburu, S. M., M. Azam, J. Luo, K. Choi and K. Pattipati, "Data-driven Modeling, Fault Diagnosis and Sensor Selection in HVAC Chillers", accepted in *IEEE Transactions on Automation Science and Engineering,* February, 2006.

[5] Choi, K., M. Namburu, M. Azam, J. Luo, K. Pattipati and A. Patterson-Hine, "Fault Diagnosis in HVAC Chillers: Adaptability of a Data-driven Fault Detection and Isolation Approach", *IEEE Instrumentation and Measurement Magazine,* pp.24-32, August 2005.

[6] Crossman J.A., H. Guo, Y.L. Murphy, and J. Cardillo, "Automotive Signal Fault Diagnostics – Part I: Signal Fault Analysis, Signal Segmentation, Feature Extraction and Quasi-Optimal Feature Selection", *IEEE Transactions on Vehicular Technology,* Vol. 52, No. 4, July 2003.

[7] Murphey, Y. L., J. A. Crossman, Z. Chen, and J. Cardillo, "Automotive Faullt Diagnosis – Part II: A Distributed Agent Diagnostic System", *IEEE Transactions on Vehicular Technology,* Vol. 52, No. 4, pp. 1076-1098, July 2003.

[8] Guo, H., J. A. Crossman, Y. L. Murphy, and M. Coleman, "Automotive Signal Diagnostics Using Wavelets and Machine Learning", *IEEE Transactions on Vehicular Technology,* Vol. 49, No. 5, September 2000.

[9] Zhang, Q. J., and Y. Yan, "A Wavelet-Based Appraoch to Abrupt Fault Detection and Diagnosis of Sensors", *IEEE Transactions on Instrumentation and Measurement,* Vol. 50, No. 5, October 2001.

[10] Kimmich, F., A. Schwarte, and R. Isermann, "Fault Detection for Modern Diesel Engines using Signal- and Process Model-based Methods", *Control Engineering Practice,* Vol. 13, pp. 189-203, 2005.

[11] Capriglione, D., C. Ligouri, and A. Pietrosanto, "On-Line Sensor Fault Detection, Isolation, and Accommodation in Automotive Engines", *IEEE Transactions on Instrumentation and Measurement,* Vol. 52, No. 4, pp. 1182-1189, August, 2003.

[12] Manders., E-J., G. Biswas, P.J. Mosterman, L. A. Barford, and R. J. Barnett, "Signal Interpretation for Monitoring and Diagnosis, A Cooling System Testbed", *IEEE Transactions on Instrumentation and Measurement,* Vol. 49, No.3, June 2000.

[13] Antory, D., U. Kruger, G. Irwin, and G. McCullough, "Fault Diagnosis in Internal Combustion Engines using Non-linear Multivariate Statistics", *Proc. IMechE, Part I: Systems and Control Engineering,* Vol.219, No. 4, pp. 243-258, 2005.

[14] Luo, J., F. Tu, M. Azam, K. Pattipati, L. Qiao, and M. Kawamoto, "Intelligent Model-based Diagnostics for Vehicle Health Management", *Proc. Of the SPIE,* pp. 13-26, Florida, April 2003.

[15] Luo, J., M. Namburu, K. Pattipati, L. Qiao, and S. Chigusa, "Integrated Model-based and Data-driven of Automotive Anti-Lock Braking Systems", Revised version submitted to *IEEE Transactions on Systems, Man, Cybernetcis, Part – A,* 2005.

[16] Luo, J., K. Pattipati, L. Qiao, and S. Chigusa, "An Integrated Diagnostic Development Process for Automotive Engine Control Systems", *IEEE Transactions on Systems, Man, Cybernetics, Part – C,* March 2006.

[17] Luo, J., K. Pattipati, L. Qiao, and S. Chigusa, "Agent-based Real-time Fault Diagnosis", Proceedings *of the IEEE Aerospace Conference,* Big Sky, Montana, March 2005.

[18] Qualtech Systems Inc. www.teamqsi.com

[19] Namburu, S. M., H. Tu, J. Luo, and K. R. Pattipati, "Experiments on Supervised Learning Algorithms for Text Categorization", *Proceedings of the IEEE Aerospcace Conference,* Big Sky, Montana, March 2005.

[20] Geladi, P., and B. R. Kowalski, "Partial Least-Squares Regression: A Tutorial", *Analytica, Chemica, Acta,* 1986.