An Integrated Health Management Process for Automotive Cyber-Physical Systems

Chaitanya Sankavaram, *Member, IEEE,* Anuradha Kodali, *Member, IEEE*, and Krishna Pattipati, *Fellow, IEEE*

***Abstract*— Automobile is one of the most widely distributed cyber-physical systems. Over the last few years, the electronic explosion in automotive vehicles has significantly increased the complexity, heterogeneity and interconnectedness of embedded systems. Although designed to sustain long life, systems degrade in performance due to gradual development of anomalies eventually leading to faults. In addition, system usage and operating conditions (e.g., weather, road surfaces, and environment) may lead to different failure modes that can affect the performance of vehicles. Advanced diagnosis and prognosis technologies are needed to quickly detect and isolate faults in network-embedded automotive systems so that proactive corrective maintenance actions can be taken to avoid failures and improve vehicle availability. This paper discusses an integrated diagnostic and prognostic framework, and applies it to two automotive systems, viz., a Regenerative Braking System (RBS) in hybrid electric vehicles and an Electric Power Generation and Storage (EPGS) system.**

***Index Terms*— automotive cyber-physical systems, fault diagnosis and prognosis, fault modeling, inference algorithms, test design.**

I. Introduction

RAPID advances in electronics, control, communication, and computing technologies have resulted in complex network-embedded automotive systems. Today's automotive vehicles contain more than 70 distributed electronic control units (ECUs), 100's of MegaBytes of software, 5 or more distinct communication networks, a wide variety of sensors and actuators, and 1000's of data and control signals exchanged in real-time every second. ECUs in modern vehicles perform a variety of cyber-physical functions, for example, stability control, remote monitoring (e.g., via OnStar), energy-efficient propulsion, adaptive cruise control, by-wire steering and braking, keyless entry with push button start, blind zone detection, lane departure warning, and autonomous driving [1]. Approximately 80-90% of these vehicle innovations are based on software-embedded systems, and this has resulted in an increase in the number of interactions among heterogeneous subsystems. Thus, in order to enhance vehicle performance and reliability, it is essential to model the complex interactions and failures in physical devices (e.g., sensors, batteries, motors), software algorithms running on ECUs, and the communication networks (e.g., controller area network (CAN), FlexRay, local interconnect network (LIN), & ethernet). Failures range from issues that affect a single subsystem (either hardware or software), to issues that occur as a result of coupling among multiple subsystems. Advanced detection, diagnosis and prognosis techniques are needed to infer and track the faults in complex automotive systems.

Fault diagnostic and prognostic (D&P) methods have mainly evolved upon three major paradigms, viz., physics-based modeling, data-driven and knowledge-based approaches. The *physics-based modeling approach* employs consistency checks between the sensed measurements and the outputs of a mathematical model. The expectation is that inconsistencies are large in the presence of malfunctions and small in the presence of normal disturbances, noise and modeling errors. Two main methods of generating the consistency checks are based on observers (e.g., Kalman filters, reduced-order unknown input observers, interacting multiple models, particle filters) and parity relations (dynamic consistency checks among measured variables stemming from hardware or information redundancy relations). A *data-driven approach* is preferred when the system monitoring data for nominal and degraded conditions is available. Neural network and statistical classification methods are illustrative of data-driven techniques. The *knowledge-based approach* uses graphical models such as dependency graphs (digraphs), Petri nets, multi-signal (multi-functional) flow graphs, and Bayesian networks for diagnostic knowledge representation and inference [2-4].

Model-based, data-driven and knowledge-based approaches provide the toolkit that system designers can use for diagnosis and prognosis. However, each of these techniques has its pros and cons, and ironically, no single technique in isolation can serve as the D&P approach for complex CPS. Thus, an integrated process that naturally combines the knowledge from data-driven techniques, graph-based system-level functional dependency models, and mathematical/physical models (for

Manuscript received October 18, 2012. The work reported in this paper was supported by the National Science Foundation under Cyber-Physical Systems grant ECCS 0931956. Any opinions expressed in this paper are solely those of the authors and do not represent those of the sponsor.

Chaitanya Sankavaram, Anuradha Kodali and Krishna R. Pattipati are with Electrical and Computer Engineering Department, University of Connecticut, Storrs, CT 06269-2157, USA (phone: 860-486-2890; fax: 860-486-5585; e-mail: chaitanya@engr.uconn.edu, anuradha@engr.uconn.edu, krishna@engr.uconn.edu).

978-1-4673-5288-8/13/$31.00 ©2013 IEEE

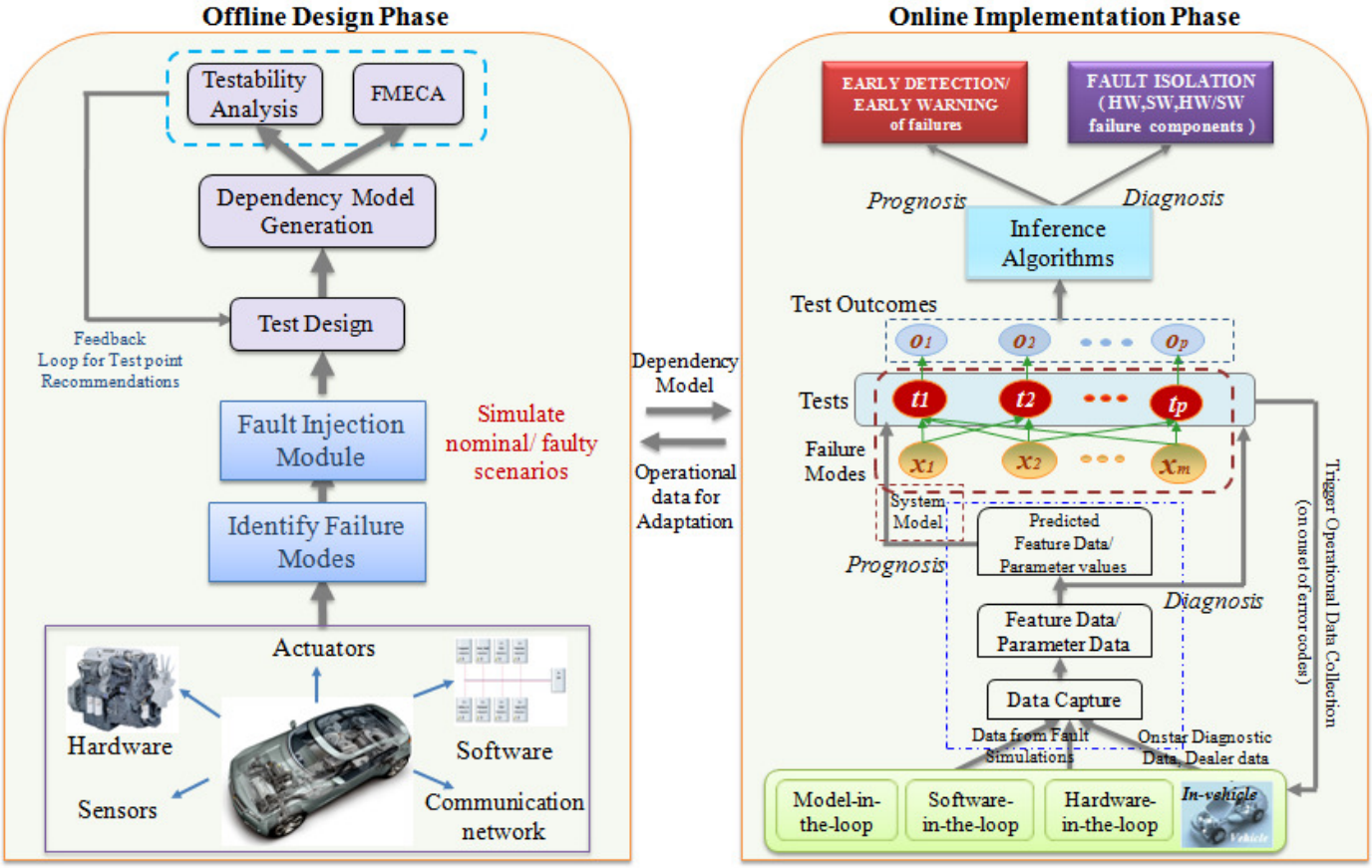

Fig 1. Integrated Diagnostic and Prognostic Framework for Networked Embedded Systems in Automotive Vehicles

components with well understood dynamics) is necessary for fault D&P of CPS. Knowledge-based graphical models [5] provide a natural means to combine fault-test dependencies from multiple sources (analytic models, fault simulations, OnStar data, dealership data, and qualitative/subjective observations) via the diagnostic dictionary (error correcting code (ECC) matrix) to perform adaptive inference, classifier fusion, and manual troubleshooting.

This paper presents an integrated (offline as well as online) diagnostic and prognostic framework and the process is validated on two automotive systems, namely, Regenerative Braking System in hybrid electric vehicles, and Electric Power Generation and Storage system. The details of the framework are discussed in Section II and the demonstration of the process on automotive systems is presented in Section III. Finally, the paper concludes with a summary in Section IV.

II. Integrated Diagnostic and Prognostic Framework

The integrated diagnostic and prognostic framework is shown in Fig. 1. The process involves *offline design* phase and *online implementation* phases.

(a) Offline Design Phase

In the *offline design* phase, as a first step, the potential failure modes associated with *software* (e.g., abnormal task termination, incomplete execution, execution at incorrect time), *hardware* (e.g., spurious current faults, stuck-at faults, open faults, bridging faults, sensor faults, actuator faults), *HW/SW interaction* (e.g., timing or synchronization failures, inappropriate update of interface value (e.g., faulty response/control of an actuator, non-specified state change), misalignment or improper connection of HW interface, etc) and *communication/network bus* faults (bus open, bus short, increase in bus wire resistance) are identified from engineering data, field maintenance data and expert knowledge.

The next step is to conduct fault simulations via model-in-the-loop, software-in-the-loop, and hardware-in-the-loop experiments. This consists of simulating the fault-free system and comparing the outputs with those of the faulty system under various modes of operation. The faulty system is created by inserting faults in hardware, software, HW/SW interfaces and network. Subsequently, monitoring mechanisms (or tests) are designed to detect faults with minimum false alarms and maximum isolation capability. The main challenge in test design is to extract useful features from sensors that are insensitive to different modes of operation of the system, and yet be able to detect/isolate faults with high detection and low false alarm probabilities and estimate their severity levels.

Tests can be designed via model-based, data-driven or knowledge-based approaches. The model-based methods use residuals as features, where the residuals are the deviations of actual measurements from the expected ones. These residuals can be generated, for instance, based on parameter estimation, observers, and parity relations. Statistical techniques/hypothesis testing (e.g., change detection

techniques [6] such as generalized likelihood ratio test, cumulative sum test, sequential probability ratio test, etc.) are then used to define thresholds to detect the presence of faults [2].

Data-driven approaches, on the other hand, are derived directly from routinely monitored system operating data. The strength of data-driven techniques is their ability to transform high-dimensional noisy data into lower dimensional features for detection and diagnostic decisions. Signal analysis methods, graphical methods, neural networks [7-8] and multivariate statistical methods [9] are illustrative of data-driven techniques. Knowledge-based approaches are based on qualitative knowledge about the system. Here, diagnostic rules are generated using domain knowledge experts [10], and faults are detected by matching historical trends of process variables under faulty conditions with the current observations [2].

Once the tests are designed, the fault-test dependency graph models are extracted through fault simulations. These dependency models can be used by existing testability analysis and FMECA tools (e.g., TEAMS [11]) to convert the dependency graph into a single global fault dictionary (or D-Matrix) via reachability algorithms. The D-Matrix contains cause-effect information needed to diagnose faults (onboard monitoring), or to minimize the troubleshooting time (maintenance in dealerships) in the *online implementation phase*. The FMECA and testability analysis (viz., fault detection, isolation, ambiguity groups, and detection and isolation delays) validate the test procedures designed for the detection of faults/errors. If the results of analysis are not satisfactory, additional and/or improved test procedures are designed to improve the diagnostic metrics, thus improving vehicle performance and avoiding costly recalls.

(b) Online Implementation Phase

In the *online implementation* phase, the data captured via onboard monitoring or telematics are transformed into features for fault detection. These features can be trended to forecast their future values for predicting component degradations. The detection/degradation results are in the form of test outcomes i.e., DTC-based triggers from on-board data or statistical test decisions derived from current sensor data or software monitors (for diagnosis) or forecasted sensor data (for prognostics). These test outcomes together with fault-test dependency model (learned from *offline design* phase) are processed through inference algorithms to infer the health status of components along with residual useful life (RUL) estimates of components. The library of inference algorithms encompasses coupled and factorial hidden Markov model-based primal-dual algorithms as well as coordinate ascent-based primal optimization techniques to deal with multiple, coupled and intermittent faults with observation delays, fault propagation delays, and imperfect test outcomes. More specifics about the algorithms can be found in [12-18].

The framework has the potential to be applicable to any networked embedded system. In fact, variants of this process have been applied to a number of engineering systems [19-21].

III. Integrated D&P Process Applied to Automotive Systems

The integrated D&P process outlined in Fig. 1 is applied to two automotive systems, namely, regenerative braking system (RBS) and electric power generation and storage system (EPGS) and the results are briefly discussed below.

a) Fault Diagnosis in a Hybrid Electric Vehicle Regenerative Braking System:

Regenerative braking is widely employed in electric and hybrid electric vehicles (HEVs) to enable energy regeneration and improve fuel economy. The primary function of a regenerative braking system (RBS) is to convert kinetic energy into electrical energy and store it in batteries during braking mode for later use in propelling the vehicle. The RBS model with series-parallel drivetrain configuration is developed using Powertrain System Analysis Toolkit, vehicle simulation software. The model consists of a driver model, a component model (physical system) and six ECUs, namely, battery control unit, engine ECU, motor1 control unit, motor2 control unit, mechanical brake control unit, and powertrain controller.

The powertrain controller is the supervisory controller making the high-level decisions that affect the general state of the powertrain (e.g. engine on/off), the operating mode of the vehicle (e.g. propelling, regenerative braking etc.), and accordingly deliver the torque requests to the component controllers. Subsequently, the torque requests are converted into component commands at the component controller. These commands are treated as the actuator commands by the individual components in the powertrain model to achieve the requested torque and consequently, report the system status (e.g., engine speed, battery state of charge) to the supervisory controller.

Figure 2 shows the simulation setup of RBS in a Vector

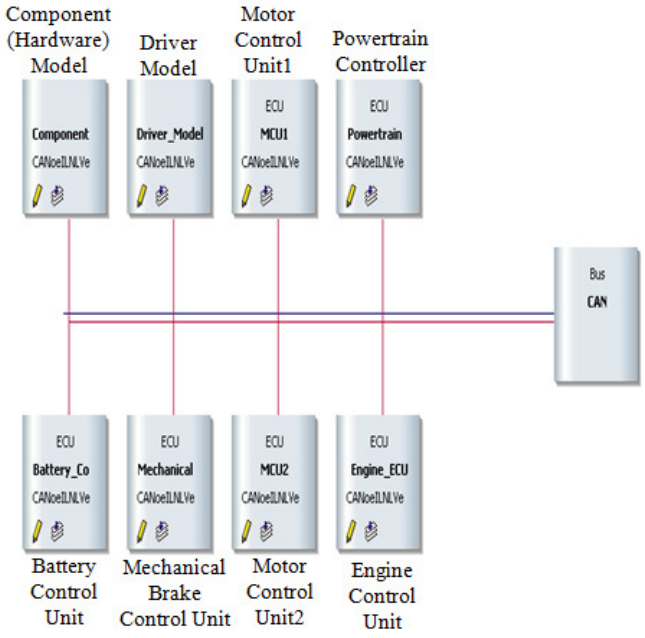

Fig 2. Simulation Setup of Regenerative Braking System in Vector CANoe Software

CANoe[1] environment. A variety of physical system faults (parametric and sensor-related faults), software logic faults, network communication faults (babbling idiot, missing message, burst loss and outdated message faults) and bus faults were injected into the system via simulation-based fault injection experiments. There are 25 signals that are monitored in the braking system including (*a*) *sensor signals*, such as temperature, speed, and current measurements from the hardware components in the powertrain model; (*b*) motor, wheel, and engine *torque demands* sent from the powertrain controller to the component controllers; and (*c*) *component commands* sent from the individual ECU's to the hardware components in the powertrain model.

Here, a data-driven approach is applied for fault detection and diagnosis. The process involves data reduction, fault detection and fault classification. In the off-line training phase, feature extraction/data reduction techniques (e.g., wavelets, multiway partial least squares (MPLS), multiway principal component analysis (MPCA) [9]) are employed on the residuals from different fault scenarios to extract salient features that capture the most information from the data. Once the features are extracted, the training of the classification techniques can be carried out in one of the following three ways:

a) Use the extracted features to train classifiers such as support vector machines (SVM), and *k*-nearest neighbor (KNN); (or)

b) Use fault detection techniques, viz., trending and thresholding to generate test results (pass/fail test outcomes), and use these to train the fault classifiers; (or)

c) Use the test results from detection techniques to learn the diagnostic matrix (Dependency Matrix), i.e., fault-test dependencies that can be later used with an inference algorithm in online phase to isolate the faults.

In the online testing phase, the trained fault classification algorithms, viz., pattern recognition techniques, or an inference algorithm are used to classify the fault based on the extracted features (reduced data). Here, the sequence employed for fault diagnosis is residuals → MPLS → fault classification via SVM and KNN. The classification accuracy was 97.06% with both classifiers. Additional details can be found in [22][23].

b) Fault Diagnosis in Electric Power Generation and Storage System (EPGS) of Automotives:

The EPGS system provides electrical power in an automotive vehicle to meet the electrical load requirements. The EPGS plant involves a drive belt, alternator, and a battery to provide the necessary power to the electrical loads of the vehicle, such as lights, fans, etc. The alternator generates voltage according to the set-point computed by the ECUs. The set-point is computed based on the load requirement and estimated battery (secondary power source) state. These systems constitute the EPGS network of embedded system, where EPGS plant model interacts with the two ECUs, viz., engine control module (ECM) and body control module (BCM). The two ECUs communicate through messages via CANbus. The BCM node takes the sensor inputs, such as battery voltage and current, from the EPGS plant node, implements the RVC control logic, and sends control signals, set-point voltage and fuel mode to the ECM node via CAN bus messages (Message 2 in Fig. 3) in CANoe. The ECM node collects and transmits sensor information, such as engine temperature and engine RPM, to the BCM node, and also receives the messages containing RVC control signals from the BCM, and sends control signals to the EPGS plant node. Fig 3 shows the EPGS experimental setup in Matlab/CANoe co-simulation environment.

In this application, the failure modes considered include component-level hardware plant faults (broken cable, belt slip fault, voltage regulator fault, battery current/voltage sensor faults), temperature related faults (out-of-range), software logic fault (initial SOC estimation fault) and global faults (BCM software error in logic, ECM software/communication fault) that can impact the entire system. Here, because of the closed-loop structure, establishing root-cause becomes difficult, for example, the ECU faults may lead to a breakdown of the whole EPGS system, while the symptoms may be very similar to the EPGS plant faults.

To address this issue, tests are designed to monitor both global variables (with system-level impact) and local variables (mainly used to isolate the root-cause within a sub-system). A 4-level hierarchical diagnostic test scheme is defined wherein the first level infers whether it is a plant fault or the ECU/ communication fault. The second level tests monitor the variables to distinguish faults between the two ECUs and the communication channel. The third level tests deal with the individual component's inputs and outputs to diagnose at that level. This includes isolating the circuit and software faults in the ECUs. At the lowest level, we isolate the failure modes at the sub-component level.

One of the global variables that is transmitted from/to a sub-system (BCM → ECM → EPGS) is the set-point voltage. Monitoring this variable at different layers of its transmission enables the bifurcation at the sub-system level, i.e., the EPGS plant or the BCM or the ECM. Similarly, monitoring battery voltage at the plant level facilitates diagnosis of the sensor or

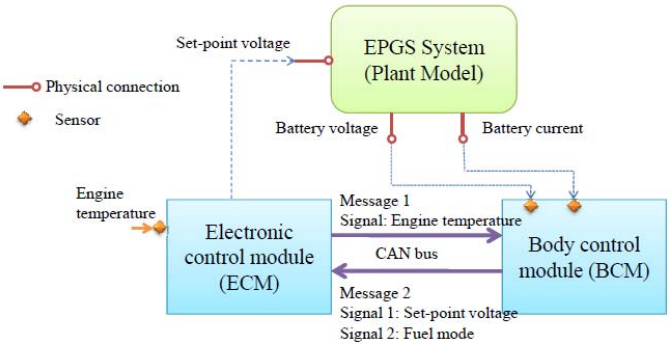

Fig. 3 Experimental setup of EPGS system in CANoe/Matlab Simulink Co-simulation

[1] CANoe is a software tool for testing and analysis of a network of ECUs.

battery problem. Table II shows the list of tests designed for EPGS system.

To generate the fault-test dependency matrix, faults are injected via fault simulations and then ascertained whether the fault is detected by each of the designed tests. Each fault is injected over time; both permanent and intermittent behaviors are captured and then the fault-test dependencies are determined. The test results are collected after fault injection for subsequent inference using the DMFD algorithm [13]. The faults are isolated with an average isolation accuracy of 97.4%. More details can be found in [24].

Table II List of Tests

TEST	
Broken cable: battery current < -20A for 10 seconds	
Belt slip: estimated pulley ratio < 3	
Voltage regulator: |Battery voltage – Set point voltage| < 1 volt	
Temp. range test: Temp. ∉ [-30 – 40 C]	ECM sensor test
	BCM sensor test
Current. range test: current. ∉ [-60 – 100 A]	Too low
	Too high
Voltage range test: voltage ∉ [9 – 16 V]	Too low
	Too high
Set-point voltage test	Test in BCM
	Test in ECM
SOC range test: Estimated SOC ∉ [40 – 100%]	

IV. CONCLUSIONS

In this paper, an integrated diagnostic and prognostic framework with offline design and online implementation phases is discussed and the process is applied to two automotive systems. With increase in the complexity of vehicular system, the work presented in this paper is critical for condition-based maintenance and to facilitate timely actions to reduce the probability of failures without any significant compromise on quality and performance. Our future research would focus on addressing the following problems: *(i)* implement a distributed fault diagnosis scheme to infer subsystem faults and in turn accurately estimate the system-level diagnostic inference; *(ii)* Prognosis and remaining useful life estimation of components and using it to accurately predict the degradation of coupled systems; and *(iii)* design of reconfiguration control strategies that facilitate fault-tolerance and recovery from adverse conditions. In addition, the D&P framework and the algorithms will be validated using hardware-in-the loop setup and in real vehicles.

REFERENCES

[1] N. Boules, "Reinventing the Automobile: The Cyber-Physical Challenge", from Embedded Systems to Cyber-Physical Systems: a Review of the State-of-the-Art and Research Needs Workshop, St. Louis, MO, April, 2008.

[2] K. R. Pattipati, A. Kodali, K. Choi, S. Singh, C. Sankavaram, S. Mandal W. Donat, S.M. Namburu, S. Chigusa, L. Qiao and J. Luo, "An integrated diagnostic process for automotive systems," in D. Prokhorov, (ed.) Studies in Computational Intelligence (SCI), Vol. 132, 2008.

[3] C. Sankavaram, A. Kodali, D. F. M. Ayala, K. Pattipati, S. Singh, and P. Bandyopadhyay, " Event-driven data mining techniques for automotive fault diagnosis", 21st Intl. Workshop on Principles of Diagnosis, Portland, OR, October 2010.

[4] C. Sankavaram, B. Pattipati, A. Kodali, K. Pattipati, M. Azam, and S. Kumar, "Model-based and data-driven prognosis of automotive and electronic systems", 5th Annual IEEE Conference on Automation Science and Engineering, Bangalore, India, August 22-25, 2009.

[5] J. Luo, H. Tu, K. Pattipati, L. Qiao, and S. Chigusa, "Graphical models for diagnostic knowledge representation and inference," IEEE Instrument and Measurement Magazine, vol. 9, pp. 45-52, 2006.

[6] M. Basseville, and A. Benveniste, "Detection of abrupt changes in signals and dynamical systems", Berlin: Springer-Verlag, (1986).

[7] R. 0. Duda, P. E. Hart, and D. G. Stork, Pattern Classification, second edition, New York: Wiley lnterscience, 2001.

[8] C. M. Bishop, Pattern Recognition and Machine Learning, Springer, 2006.

[9] P. Nomikos, "Detection and diagnosis of abnormal batch operations based on multi-way principal component analysis," ISA Transactions, Vol. 35, pp. 259–266, 1996.

[10] E. J. Henley, "Application of expert systems to fault diagnosis", in AIChE annual meeting , San Francisco, CA, 1984

[11] Qualtech Systems Inc. Website. http://teamqsi.com, accessed on July 28th, 2012.

[12] V. Raghavan, M. Shakeri and K. R. Pattipati, "Test sequencing problems arising in test planning and design for testability," IEEE Trans. Syst., Man, Cybern. Part A, (SMCA), vol. 29, pp. 153-163, Mar. 1999.

[13] S. Singh, A. Kodali, K. Choi, K. Pattipati, S. M. Namburu, S. Chigusa, D. V. Prokhorov, and L.Qiao, "Dynamic Multiple Fault Diagnosis Problem Formulations and Solution Techniques," IEEE SMCA, vol. 39, no. 1, pp. 160-176, January 2009.

[14] A. Kodali; S. Singh; K. Pattipati, "Diagnostic set-covering for real-time multiple fault diagnosis with delayed test outcomes", accepted for publication in *IEEE SMCA,* 2012

[15] A. Kodali; S. Singh; K. Choi; K. Pattipati, "Diagnostic ambiguity and parameter optimization in dynamic fusion of multiple classifier systems", submitted to Annals of Information Systems (AoIS), Springer Special Issue on Real World Data Mining Applications, 2011

[16] A. Kodali, K. Pattipati, and S. Singh, "Coupled factorial hidden Markov models for diagnosing multiple and coupled faults," IEEE Trans. on SMC: Part A (accepted), September 2011.

[17] S. Zhang, K. Pattipati, Z.Hu, X. Wen, and C. Sankavaram, "Dynamic Coupled Fault Diagnosis with Propagation and Observation Delays", submitted to SMCA (under Review), 2011.

[18] S. Zhang, K. Pattipati, Z. Hu, X. Wen, "Optimal Selection of Imperfect Tests for Fault Detection and Isolation", IEEE SMCA, 2011 (being revised).

[19] K. Choi, J. Luo, K. Pattipati, S.M. Namburu, L. Qiao, and S. Chigusa, "Data Reduction Techniques for Intelligent Fault Diagnosis in Automotive Systems", *Proceedings of the IEEE Autotestcon*, Anaheim, CA, September, 2006.

[20] K. Choi, S. M. Namburu, M. S. Azam, Jianhui Luo, K. R. Pattipati and A. Patterson-Hine, "Fault diagnosis in HVAC chillers," *Instrumentation & Measurement Magazine*, IEEE, vol. 8, pp. 24-32, 2005.

[21] W. Donat, K. Choi, W. An, S. Singh, and K. R. Pattipati, "Data Visualization, Data Reduction, and Classifier Fusion for Intelligent Fault Detection and Diagnosis in Gas Turbine Engines", *ASME Journal of Engineering for Gas Turbines and Power*, June 2007.

[22] C. Sankavaram, B. Pattipati, K. Pattipati, Y. Zhang, M. Howell, and M. Salman, "Data-driven Fault Diagnosis in a Hybrid Electric Vehicle Regenerative Braking System", *in proc.* of IEEE Aerospace Conference, March 2012.

[23] C. Sankavaram, K. Pattipati, Y. Zhang, M. Howell, and M. Salman, "Fault Diagnosis and Prognosis in Cyber-Physical Systems with an Application to Hybrid Electric Vehicle Regenerative Braking System", to be submitted to Elsevier Journal of Information Sciences, 2012.

[24] A. Kodali, Y. Zhang, C. Sankavaram, K. Pattipati, and M. Salman, "Fault Diagnosis in Cyberphysical Systems: Application to Electric Power Generation and Storage System (EPGS) of Automotives" accepted for publication in IEEE/ASME Trans. on Mechatronics, 2011.