Inductive Learning for Fault Diagnosis

Hamid R. Berenji, Jayesh Ametha, David Vengerov
Intelligent Inference Systems Corp.
Computational Sciences Division, MS: 269-2
NASA Ames Research Center
Moffet Field, CA 94035
(berenji, jayesh14, vengerov)@iiscorp.com

Abstract— **There is a steadily increasing need for autonomous systems that must be able to function with minimal human intervention to detect and isolate faults, and recover from such faults. In this paper we present a novel hybrid Model based and Data Clustering (MDC) architecture for fault monitoring and diagnosis, which is suitable for complex dynamic systems with continuous and discrete variables. The MDC approach allows for adaptation of both structure and parameters of identified models using supervised and reinforcement learning techniques. The MDC approach will be illustrated using the model and data from the Hybrid Combustion Facility (HCF) at the NASA Ames Research Center.**

I. Introduction

The problem of Fault Diagnosis and Isolation (FDI) has attracted considerable interest in recent years [1], [2]. The two main approaches to this problem are model-based and knowledge-based. Model-based methods require good analytical understanding of the system. For example, modeling a nozzle with fluid flow requires adequate understanding of the fluid dynamics involved as well as understanding the nozzle structure. Such analytical knowledge of the system can help in constructing observers (for deterministic systems) or filters (for stochastic systems) for estimating its state variables, using the observed stream of data. However, the exact models of components are often not available in complex systems, and have to be learned from data using external system's observation. In such cases, knowledge-based methods such as data clustering approach [5] can be used.

In this paper we present a novel fault diagnosis architecture for complex dynamic systems with continuous inputs and outputs. It will use a hybrid Model-Based and Data Clustering method for fault diagnosis (MDC). MDC is intended to be developed as an open knowledge-based architecture. In section II we describe the MDC architecture and in section III we discuss its main features. In section IV we give an overview of the Hybrid Combustion Facility (HCF) at the NASA Ames Research Center, which provided the data for our evaluation of the MDC architecture. The experimental results are presented in section V. Section VI describes related work and section VII concludes the paper.

II. The MDC Architecture

The first step of MDC is creation of component simulation models (CSMs) for each component using clustering-based model identification algorithm applied to the nominal input and output data, which corresponds to the component working properly. The clustering algorithm performs system identification by extracting rules that describe the input-output behavior of each component. The details of this algorithm are described in section II-A.

Next, if adequate data representing various types of faults is available, it is used in a similar manner to create a component simulation model for each fault. If data is not available or is inadequate, then the nominal CSM created earlier can be adjusted in accordance with expert's knowledge to create a faulty CSM, which can then be simulated to get the data for that fault.

The component simulation models are used to create a wide range of output values (both nominal and faulty) for each component. These output values together with corresponding inputs are used as inputs to the component diagnostic models (CDMs). The complete training data for identifying a CDM is obtained by assigning an output value of 1 to the CDM input vector if the data point represented a faulty component and assigning a value of 0 otherwise. After that, the clustering-based model identification algorithm is run on the CDM training data and RBF rules describing each CDM are obtained. The smooth Gaussian kernels used in the CDM rules imply that a new CSM input-output pair will be mapped by the corresponding CDM into a diagnostic signal between 0 and 1 (with

0-7803-7810-5/03/$17.00 ©2003 IEEE

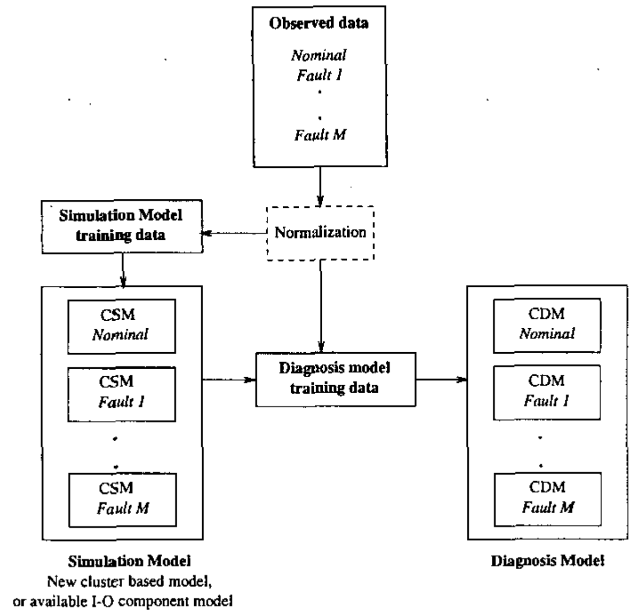

Fig. 1. Training of MDC

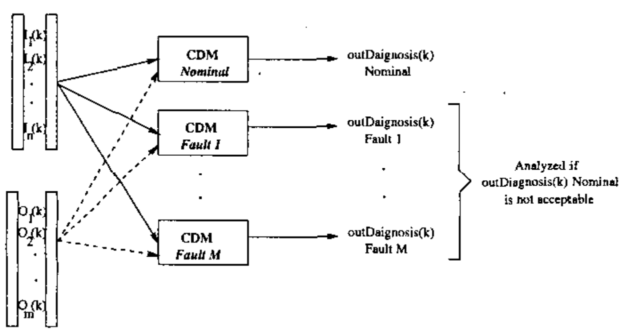

Fig. 2. Diagnosis using MDC

higher signals indicating a higher degree of faultiness) even if the training data contains either perfectly normal or completely faulty outputs. Figures 1 and 2 show the process of fault diagnosis using MDC and how the training data for MDC is obtained.

The component simulation and diagnostic models created using the data clustering approach can be trained further by supervised or reinforcement learning approach, as described in Section II-B. If accurate physical models already exist for some components, the hybrid MDC approach can utilize those models for producing a wide range of training input-output data points for CDM instead of using component simulation models described above. Also, if there is some human expert knowledge about the expected outputs of a component for certain classes of inputs, it can be added in the form of extra rules in additions to the ones identified from data.

The MDC approach can also be applied at a higher level of hierarchy for diagnosing the whole system, by using the same hybrid two-step process for constructing a system diagnostic model, SDM. After that, a faulty signal from the SDM can indicate the need for inspecting individual components in order to isolate the fault. This process can proceed as described in [3]. The authors in that work use a dynamic programming algorithm for determining the sequence in which components should be tested in order to minimize the total inspection cost. Their dynamic programming method relies on having *a priori* probabilities of each component being faulty. Our MDC approach allows us to evaluate these probabilities as a function of the diagnostic signal from each component. In the simplest case, $P(fault) = outDiagnostic$ can be used.

A. Clustering Algorithm

We first describe the mathematical form of the RBF rulebase, which is identified by our clustering algorithm. We assume a general case of a rulebase with n inputs and m outputs. The inputs to the rulebase are assumed to be normalized to fall within the range [0,1]. Each rule r has the following form:

IF (s_1 is ξ_{r1}) and ... and (s_n is ξ_{rn}) THEN $y_r k = \sum_{i=0}^{n} c_{rki} s_i$,

where c_{rki} and ξ_{r1} are tunable coefficients, $k = 1, ..., m$. The weight of rule r for a data point s is determined according to the distance between the vector **s** and the center of an n-dimensional Gaussian sphere with a mean of $(\xi_{r1}, ..., \xi_{rn})$ and a standard deviation σ_r:

$$w_r = \exp\left(-\frac{1}{2\sigma_r^2} \sum_{i=1}^{n} (\xi_{ri} - s_i)^2\right).$$

When normalized, this gives

$$\rho_r = \frac{w_r}{\sum_{k=1}^{N} w_k}.$$

Finally, the jth output of the rulebase is given by $O_j = \sum_{r=1}^{N} \rho_r y_{rj}$, where N is the number of rules. It has been shown that such a configuration can approximate any nonlinear function to any desired degree of accuracy [4] if the number, the locations, and the variances of Gaussian spheres are allowed to change.

The clustering-based approach to identifying an RBF rulebase from data is described in Figure 3. It begins by

Let $\mathbf{s}$ and $\mathbf{d}$ be the input and output parts of each data point. *For each* point $(\mathbf{s},\mathbf{d})$ do {
INFERENCE :
$\mathbf{O}$ = Rulebase output when input ($\mathbf{s}$) is presented.
Let the index of the nearest (Gaussian) rule be
$i^* = argmin_r\{\|\mathbf{s} - \xi_r\|\}$;
Let distance to the nearest rule be
$d^* = \{\|\mathbf{s} - \xi_{i^*}\|\}$;
Let the applicability of the nearest rule be
$w^* = \exp(-\|\mathbf{s} - \xi_{i^*}\|^2/2\sigma_{i^*}^2)$;
Let the error of that rule be
error = $\|\mathbf{O} - \mathbf{d}\|^2$;

ADAPT_PARAMETERS :
modify parameters of all existing rules:
$\Delta c_{jki} = \eta(d_k - O_k)\rho_j s_i$
$\Delta\xi_{jk} = \eta\,[\sum_{l=1}^{m}(O_l - d_l)y_{jl}]\,\frac{1}{\sigma_j^2}\rho_j(1-\rho_j)(\xi_{jk} - s_k)$
$\Delta\sigma_j = -\eta\,[\sum_{l=1}^{m}(O_l - d_l)y_{jl}]\,\frac{1}{\sigma_j^3}\rho_j(1-\rho_j)\|\xi - \mathbf{s}\|^2$

ADD_RULE :
if $(w^* < \delta)$ {
add rule at $\mathbf{s}$ with spread $d^*/\sqrt{2\ln 2} - \sigma_{i^*}$;
INFER and ADAPT_PARAMETERS;
else if (error $> \epsilon$) {
add rule at $\mathbf{s}$ with spread σ_{min}
INFER and ADAPT_PARAMETERS;}
}
PRUNE_RULES :
for each pair of remaining rules n_a and n_b $(a < b)$ *do* {
Let θ_{jab} = angle(hyperplane$_{aj}$,hyperplane$_{bj}$);
Let $d_{ab} = \|\xi_a - \xi_b\|$;
if $(\max_j \theta_{jab} < \omega)$ {
if $(\sigma_a > d_{ab})$ { winner := a; loser := b }
else if $(\sigma_b > d_{ab})$ { winner := b; loser := a }
else consider next pair;
move winner towards loser in ratio $\sigma_{winner}^n : \sigma_{loser}^n$;
expand winner's radius to include loser's radius;
delete loser rule;}
}

Fig. 3. Structure of the clustering algorithm [5].

creating an RBF rule with the Gaussian center coinciding with the first data point. When the next data point is encountered, the parameters of the first rule are adapted to account for both data points. If the error on the second data point is still too large, then a second rule is created centered at the second data point. The process continues until all data points have been considered. This is a very fast one-pass algorithm, which still gives very good results as our experiments indicate. For more information about the clustering algorithm, see [5].

B. Adaptation in MDC

The environment containing complex engineering systems usually changes over time. As a result, the response of each component changes, and one of the most difficult tasks in fault diagnosis is to determine whether a degradation in the diagnostic signal is due to changes of the environment or to a malfunctioning of the component. A possible way to address this problem in MDC is to inspect each component for which the fault diagnostic signal rises above a specified threshold and evaluate it to be faulty or not. If it is faulty, it is replaced, but if it is not, an adaptation phase is triggered for the nominal CSM and the CDM of that component. The procedure described next can be used when a new data set is observed for any component, which was determined to be working properly after inspection.

The nominal data set on which the CSM for that component was trained is augmented with the new data. After that, the clustering algorithm is used on the augmented data to adapt either parameters only or also the structure of the RBF rules composing the CSM.

The CDM then needs to be updated to bring the diagnostic signal back to 0. The observed component data, along with the new simulated data produced by the updated CSM can now be used to re-train and extend the nominal CDM. A new set of faulty data can also be obtained by modifying the CSM rules in accordance with expert's knowledge of faulty components and simulating the modified CSM to obtain faulty input-output data.

Reinforcement learning algorithms such as ACFRL [6] can also be used for adjusting the CDM parameters with the objective of bringing the diagnostic signal to 0, on the new simulated nominal data. An example demonstrating how ACFRL can be used for tuning the output parameters of rule-based systems is provided in [7].

III. Main Features of MDC

Continuous states: Continuously varying states can be effectively handled by the RBF rules used in MDC. Also, the RBF rules allow the diagnostic signal to vary continuously between 0 and 1 instead of having to take on binary values.

Dynamic components: Finite state automata are usually static functions, which cannot represent dynamic relationships in the evolution of a component. For time-variant systems, dynamic relationships provide a more complete representation. The proposed system can represent dynamic relationships by including time information in the rulebase antecedents and consequents.

Incorporation of expert knowledge: If expert knowledge is available or analytical relationship between some variables is known, it can be directly coded into linguistic rules and added to the rulebase, thus enriching the description of the component. In that sense, the model is scalable: it can easily be "tweaked" and extended by adding new rules as new knowledge becomes available. This is useful when real training data is not available or inadequate.

Degree of faultiness: The degree of faultiness in the component can be judged apart from identifying the type of fault. Even in absence of fault it can give a quantitative measure of healthiness of a component, or the system on the whole.

Fault prediction: Each component is represented by an RBF rulebase which, in addition to detecting faults after they have occurred, can be used to predict the occurrence of faults, by identifying the input-output data structure before the fault. Wherever this is possible, such rules will greatly improve the ability of the system to reconfigure itself. This capability of performing trend analysis is very critical in designing for safety, and will enable us to prevent some faults from occurring.

IV. Overview of Hybrid Combustion Facility (HCF)

A Hybrid Combustion Facility (HCF) was recently built at NASA Ames Research Center to investigate the combustion properties of a new fuel formulation developed by Stanford University researchers. The fuel being tested at the HCF is paraffin-based, and the oxidizer is gaseous oxygen. A brief overview of the Hybrid Combustion facility and its operation is presented below. A detailed description of HCF can be found in [8]. Figure 4 depicts our component-level model of HCF.

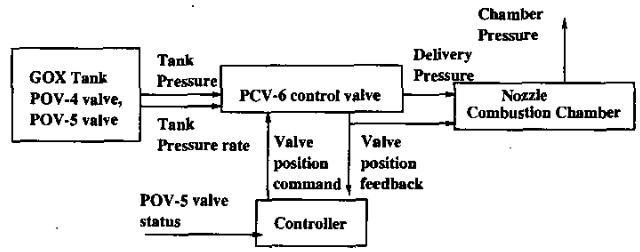

Fig. 4. Hybrid Combustion Facility

HCF is comprised of 5 main physical components: LOX tank, GOX tank, combustion, ignition, and pneumatics. Oxidizer stored in LOX tank is pumped through the vaporizer and is gasified before it enters GOX tank. Oxidizer is pumped until GOX tank pressure reaches the required level for the desired mass flow and duration of HCF operation. LOX feed is then isolated from GOX tank and desired set points such as control-valve scheduling, ignition timing, desired delivery pressure, and other configuration information are entered into the control computer. After a firing countdown, a shut off valve (POV4) is opened, allowing oxidizer to flow from the tank. The flow is controlled by varying the size of a control valve (PCV-6) with commands from the controller, ensuring constant delivery pressure. The flow chokes at the sonic nozzle, separating the delivery pressure from spikes in chamber pressure downstream, as the flow continues into the combustion chamber.

High temperature combustion products from the ignition system are injected into the combustion chamber containing paraffin fuel, vaporizing it. The oxidizer and ignition products mix with the vapor, igniting it in a self-sustaining combustion reaction.

A. HCF Modeling

For the purpose of diagnosis, the HCF system was modularized into 4 components based on the available sensor measurements. GOX tank was modeled as a finite source along with its downstream valves, together forming the first component. Its outputs are the GOX tank pressure and tank pressure rate. Control valve, PCV-6, forms the second component. It receives valve position commands from the controller as one of its inputs, while delivery pressure and actual valve position are its two outputs. The third component is the controller that receives actual valve position, and POV-5 position status as inputs. The final component comprises the sonic nozzle along with the combustion chamber. Its inputs are the delivery pressure

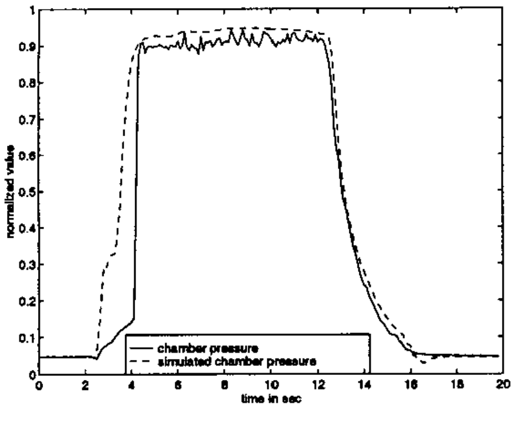

Fig. 5. Simulation of HCF combustion chamber

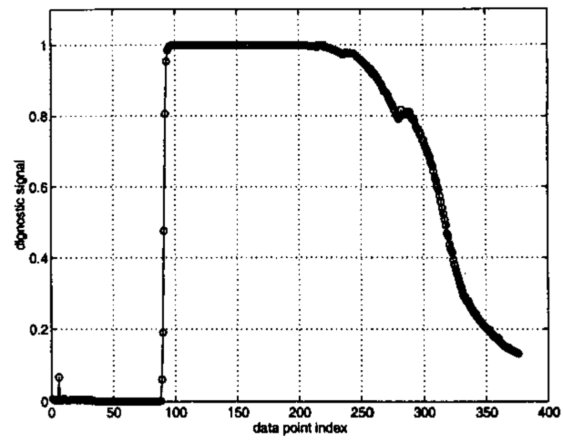

Fig. 6. Diagnosis of HCF chamber when injector catches fire midway through the run

and PCV-6 actual valve position, while its output is the chamber pressure.

V. Experimental Results and Discussion

The MDC approach to fault diagnosis has been applied to the data collected from the Hybrid Combustion Facility, HCF. In this paper we describe the results for the last HCF component, combustion chamber, where the fault has occurred in our data set. In order to build a model for the chamber pressure with our clustering algorithm, we proceeded as described in Section II. Figure 5 shows a plot of the actually observed and predicted chamber pressure on nominal test data. Figure 6 shows a plot of the output produced by the component diagnostic model, when an injector catches fire midway through the run causing rapid nozzle ablation, and the condition subsides after the control valve is closed toward the end of the run. As the plot shows, the CDM correctly diagnoses the component to be healthy before the fault occurs, then its fault diagnostic signal rises to 1 signaling a fault occurrence, and reducing slowly indicating the subsiding condition due to valve closure just before end of the run.

VI. Related Work

Trunov and Polycarpou [9] present a robust fault diagnosis scheme for detecting and approximating state and output faults occurring in a class of nonlinear MIMO dynamical systems. They model the changes in the system dynamics due to a fault with nonlinear functions of control input and measured output variables. Similar approach can train our CDMs with a class of nonlinear functions.

Li and Kadirkamanathan [10] use particle Filtering (PF) in fault diagnosis of nonlinear stochastic systems. PF is a Monte-Carlo technique for nonlinear non-Gaussian state estimation. A big advantage of Particle Filtering is its capability to handle any functional nonlinearity and system or measurement noise of any probability distributions. Li and Kadirkamanathan formulate the fault Diagnosis and Isolation (FDI) in the Multiple Model (MM) environment and then by combining the Likelihood ratio (LR) test with the PF, they develop a new FDI scheme. Our work has strong relationships with the Particle Filtering approach by treating the *outDiagnostic* signals generated by the CDMs as probabilities. Similar to PF, which does not require a complete probability distribution, we do not require complete CDMs and the system can be queried for only a limited sample similar to PF.

Shakeri et al. [3] developed sequential testing algorithms for multiple fault diagnosis. This was developed by employing concepts from information theory and graph search and exploiting the case for a single fault testing. Their work presents diagnostic strategies that generate a diagnostic directed graph (digraph). We plan to integrate CSMs and CDMs into their work as done in TEAMS software from QualTech.

Finally, our work is very much in line with the work of Diao and Passino [11] who provide an algorithm for a stable fault-tolerant adaptive Fuzzy/Neural Control for a Turbine Engine. They first generate data that is used by a Levenberg-Marquardt method to train a fuzzy system of Takagi-Sugeno type. Similar to that, we use our own clustering method to produce similar results. The resulting nonlinear model of Diao and Passino's provides an approximate representation of engine deterioration and fault effects.

VII. Conclusion

In this paper we presented a hybrid Model Based and Data Clustering (MDC) architecture for fault monitoring and diagnosis, which is suitable for complex dynamic systems with continuous and discrete variables. It was tested on data obtained from the Hybrid Combustion Facility (HCF) at the NASA Ames Research Center. When applied to a testing data set, MDC was able to diagnose correctly the fault that occurred in the combustion chamber.

The monitoring and diagnosis processes using MDC require minimal computation. As a result, the MDC architecture can be used in real time for faster monitoring and diagnosis. The MDC architecture offers a lot of flexibility because a variable can be assigned different levels of granulation in different components. The number of fault hypotheses to be checked when a fault occurs does not increase exponentially. MDC can be applied to complex non-linear systems for which the analytical models are not available (such as combustion chamber in HCF). However, the hybrid nature of MDC allows for incorporation of expert's knowledge, or known physical relationships between variables in the form of rules. MDC can adapt to changes in the environment using supervised learning when input-output data is available or using reinforcement learning when the correct outputs are not known. MDC can be applied at different levels of hierarchy of system components. Therefore, it can be used to diagnose the health of the whole system as well as to identify faults and their degree at a lower level of individual components.

VIII. Acknowledgment

We would like to thank Scott Poll and David Iverson from NASA Ames Research Center for providing us with the data and a description of the Hybrid Combustion Facility. The first author has benefited from many useful discussions with Dr. Ann Paterson-Hine and Peter Robinson from NASA Ames Research Center.

References

[1] P.M. Frank, "Fault diagnosis in dynamic systems using analytical and knowledge based redundancy–A survey and new results," *Automatica*, vol. 26, pp. 459-474, 1990.

[2] R. Isermann, "Process fault detection based on modeling and estimation methods– A survey," *Automatica*, vol. 20, pp. 387-404, 1984.

[3] M. Shakeri, V. Raghavan, K.R. Pattipati, and A. Patterson-Hine, "Sequential Testing Algorithms for Multiple Fault Diagnosis," *IEEE Transactions on SMC: Part A - Systems and Applications*, Vol. 30, No. 1, pp. 1-14, January 2000.

[4] L.X. Wang, "Fuzzy systems are universal approximators," Proceedings of IEEE International Conference on Fuzzy Systems, pp. 1163-1169, 1992.

[5] H. Berenji and P. Khedkar, "Clustering in Product Space for Fuzzy Inference" Proceedings of the IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), 1402-1407, San Francisco, 1993.

[6] H. Berenji and D. Vengerov, "On convergence of fuzzy reinforcement learning," Proceedings of the 10th IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), 2001.

[7] D. Vengerov and H. Berenji, "Using Fuzzy Reinforcement Learning for Power Control in Wireless Transmitters," Proceedings of the 11th IEEE International Conference on Fuzzy Systems, 2002.

[8] S. Poll, D. Iverson, J. Ou, D. Sanderfer and A. Patterson-Hine, "System Modeling and Diagnostics for Liquefying-Fuel Hybrid Rockets." NASA Ames Research Center internal report, 2002.

[9] A. B. Trunov and M. M. Polycarpou, "Automated Fault Diagnosis in Nonlinear Multivariable Systems Using a Learning Methodology," *IEEE Transactions on Neural Networks*, Vol. 11, No. 1, January 2000.

[10] P. Li and V. Kadirkamanathan, "Particle Filtering Based Likelihood Radio Approach to Fault Diagnosis in Nonlinear Stochastic Systems," *IEEE Transactions on Systems, Man, and Cybernetics-Part C: Applications and Reviews*, Vol. 31, No. 3, August 2001.

[11] Y. Diao and K. M. Passino, "Stable Fault-Tolerant Adaptive Fuzzy/Neural Control for a Turbine Engine," *IEEE Transactions on Control Systems Technology*, Vol. 9, No. 3, May 2001.