N94-35066

AN INTELLIGENT CONTROL SYSTEM FOR FAILURE DETECTION AND CONTROLLER RECONFIGURATION

SAROJ K. BISWAS
Intelligent Systems Application Center
Department of Electrical Engineering
Temple University
Philadelphia, PA 19122
Phone: (215)-204-8403, FAX: (215)-204-6936
E-mail: SBISWAS@TEMPLEVM.BITNET

ABSTRACT

We present an architecture of an intelligent restructurable control system to automatically detect failure of system components, assess its impact on system performance and safety, and reconfigure the controller for performance recovery. Fault detection is based on neural network associative memories and pattern classifiers, and is implemented using a multilayer feedforward network. Details of the fault detection network along with simulation results on health monitoring of a dc motor have been presented. Conceptual developments for fault assessment using an expert system and controller reconfiguration using a neural network are outlined.

I. INTRODUCTION

With the increased demand for reliability and safety, there is a need for an intelligent restructurable control system which has the ability to detect a system fault as early as possible and restructure the controller in the event of a failure. In addition such intelligent control systems must operate in real time or near real time so as to be able to predict faults before their actual occurrence. The intelligent control systems should also be able to classify a fault in terms of its type and severity for the human supervisor or *an expert system* supervisor monitoring the performance of the system.

In this paper we propose an Intelligent Restructurable Control System consisting of three functional modules or subsystems: a) a Fault Detection and Isolation module, b) a Fault Assessment and Strategy Development module, and c) a Control Reconfiguration module. The fault detection subsystem utilizes a neural network that detects and classifies faults by associating the faults with patterns of sensor data. Fault assessment and strategy development require analyses of linguistic phrases and decision making so that an expert system is the best tool for the implementation of this module. Since control reconfiguration must be carried out in real time or near time, suitable table lookup or parallel computational facility is essential for controller reconfiguration. Thus this intelligent control system integrates three of the emerging technologies that have become synonymous with intelligent control, namely, neural network, expert system, and parallel processing.

The paper is organized as follows: Section II presents the intelligent restructurable control system architecture which is followed by development of the FDI neural network and simulation results in Section III. The paper is concluded with some remarks in section IV.

II. AN INTELLIGENT RESTRUCTURABLE CONTROL SYSTEM

A restructurable control system must have the capability to accommodate automatically for any unanticipated failure of system components, and to reconfigure the controller by way of a real time redesign of the control system. The key feature that is of fundamental concern in this respect is that the response time is limited. The complete restructurable control problem may be broken into three distinct but interrelated problems:

1) Fault detection and isolation (FDI),
2) Assessment of the FDI results and strategy development
3) Control reconfiguration, and

In this paper we propose an Intelligent Restructurable Control System as shown in Figure 1 consisting of three functional modules or subsystems: a) a fault detection and isolation module, b) a fault assessment and strategy development module, and c) a control reconfiguration module. Once the controller is redesigned following a failure, implementation of the controller must be carried out through available hardware/software. The following subsections outline the underlying principles of operation of each of the above modules.

2.1 Fault Detection and Isolation

Two key requirements for the fault detection system are:

a) It must operate in real time or near real time

b) It must classify the fault in terms of its type and severity.

The proposed diagnostic neural network methodology for failure detection is based on the analyses of patterns of plant sensor data. In this approach, a fault is conceptualized as a mapping or association of a pattern of the process data (measured through sensors) to a fault condition. Since neural networks are known to perform best as pattern classifiers, a neural network fault identifier is used in this FDI subsystem.

2.2 Fault Assessment and Strategy Development

Once the FDI subsystem has detected and identified a failure, an assessment of the FDI results must be made. As one may expect, this task may require analysis of linguistic phrases and decision making; thus an expert system is required in this subsystem. The expert system accepts information from the fault identification (FDI) neural network, and classifies the fault as 1) a survivable fault, or 2) a catastrophic fault. In the event of a survivable fault, the expert system will prescribe immediate notifications to the Control Reconfiguration subsystem.

2.3 Control Reconfiguration

The Control Reconfiguration subsystem actually redesigns the controller for the faulty system. For high speed dynamic systems, computation time for control redesign is an important factor in choosing the method to be used. A two stage control reconfiguration strategy is proposed in this paper: 1) a fast inner loop **stabilizing controller** for immediate stabilization of the faulty system implemented through neural network, and 2) a more accurate outer loop **performance recovery controller** implemented through vector/parallel processor. The inner loop controller stabilizes the system and brings it to safety. The outer loop controller recovers as much of the prefault system performance as possible [10].

In this paper we present the details of the fault detection and isolation subsystem. Research for implementation of the other two subsystems is in progress.

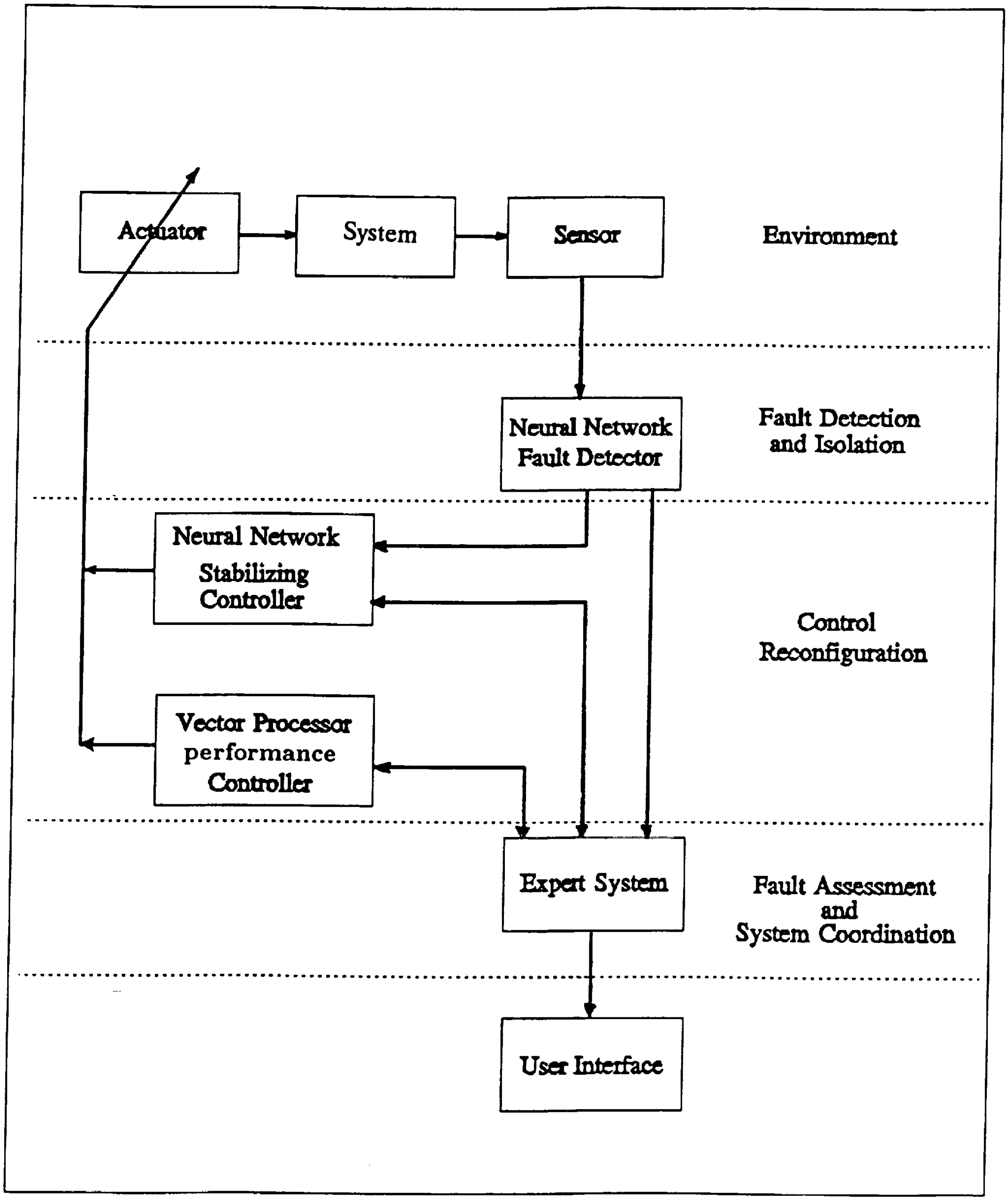

Figure 1. Architecture of the Intelligent Restructureable Control System

III. FAULT DETECTION AND ISOLATION

3.1 Background

Traditionally, human evaluation has been used as a primary means of fault detection, however this is grossly inadequate and prone to misjudgments. The earliest methods of failure detection have been based on limit violation and/or trend violation [1] of sensor data. Since human evaluation could be unrealiable, analysis of sensor data [2,3] to extract certain characteristic features of the process has been suggested. These analyses include computation of process efficiency, estimation of frequency spectrum, and the autocorrelation function, etc. requiring the use of Kalman filters, observers, and FFTs.

Another method of identifying a fault is by measurement or estimation of parameters [4,5,6,7] that govern the input-output mapping of the process. Least square estimation, Kalman filter [5], maximum likelihood estimation [6] are carried out to estimate the unknown parameters. However, this method suffers from lack of uniqueness which may lead to false alarms and/or wrong diagnosis of faults.

The use of neural network as an effective alternative for failure detection and classification to overcome the inherent drawbacks of the traditional methods has become a subject of current research interest. An application of neural networks for failure detection has been reported in [8,9] for the detection of incipient fault in a single phase induction motor.

3.2 Neural Network Based Failure Detection and Isolation

This section presents the proposed Failure Detection and Isolation (FDI) subsystem based on artificial neural network. The primary function of the FDI system is fault identification and determination of the *type of fault* and its *level of severity*. Fundamentally, a fault can be considered as a mapping or association from fault scenarios to patterns of sensor data. Neural networks are known to perform best as associative memories and pattern classifiers, and hence is used as the basic building block of the FDI system.

3.2.1 Fault Representation

One of the functions of the FDI subsystem is to classify a fault in terms of its type and its level of severity. The types of faults that may occur in a dynamical system are very much system specific. Levels of severity can be classified in several discrete conditions such as severe fault, mild fault, normal operation etc. In any case we can designate typical fault conditions in terms of a set, say $F_1, F_2, \cdots F_n$. This set of faults includes both *the type of fault* as well as *the severity of the fault*. For example: F_1 may represent mild (level of severity) damage of a steam valve (type of fault) while F_2 may represent a more severe version of the same fault or a different kind of fault altogether.

3.2.2 Network Architecture

Since the Failure Detection and Isolation system is based on associative memories and pattern classifiers, a multilayer feedforward network is considered as the basic network topology for the FDI system. The network is arranged into an input layer, an output layer, and one or more hidden layers. The input layer receives sensor data information from the process, and the output layer provides the fault diagnostic information.

The number of neurons in the input layer is equal to the number of sensors that carry out the measurement of various quantities in the process. The neurons in the input layer have linear input-output mapping. The neurons in the hidden layer(s) have a sigmoidal input-output map. The number of neurons in the hidden layer(s) must be sufficiently large so that the fault library (or the stored fault scenarios) of the network is of an acceptable size for a given system.

The number of neurons in the output layer is equal to the number of bits of the binary number corresponding to the number of faults that must be identified by the FDI network. The neurons in the output layer have sigmoidal input-output mapping so that each neuron can take discrete values 0 or 1. The on-off status of the neurons in the output layer is translated to a fault code or fault number corresponding to certain type of fault at certain level of severity. For example, consider a system in which 16 faults $F_1, F_2, \cdots F_{16}$, are to be identified. Then the output layer must have 4 nodes.

3.2.3 Training

Training of a neural net is the process by which the values of the weights are determined based on historical fault data. For the purpose of training it is essential that historical or simulated fault data be available representing various types of faults along with their possible levels of severity. Training of the FDI network may be done using the back propagation [11,12] algorithm or other adaptive training algorithms [13]. Also, pruning algorithms can be used for the purpose of reducing the number of interconnection of the neurons in the network.

3.2.4 Example: Fault Monitoring of a DC Motor

The Failure Detection and Isolation (FDI) system described above has been used for monitoring of the health status of a dc motor. The state of the dc motor can be described in terms of three state variables, such as speed of rotation, armature current, and temperature. These are also physically measurable variables through appropriate sensors. The control input to the motor is the armature voltage.

It was desired to monitor the performance the motor in terms of five different types faults, such as

a) Faulty Controller
b) Partially shorted winding
c) Excessive bearing friction
d) Motor overload
e) Blocked ventilating system

at three levels of severity, such as

a) Mild
b) Moderate
c) Severe

Thus corresponding to the five different types of faults at three levels of severity, a total of 15 possible (single) faults must be identified along with one normal operating state. Each of these 16 faults are then assigned a unique binary number ranging from 0000 to 1111 with 0000 being the *no fault* and 1111 representing a severely faulted motor. Clearly this requires 4 nodes in the output layer of the FDI network. Since there are three measurable signals available from the motor, the input layer has 3 nodes, one for each sensor signal. The FDI network was assumed to have two hidden layers with eleven nodes in each layer. The number of nodes in the hidden layers were determined after some trial and error simulation.

In order to train the FDI neural networks to give correct responses a large data base of fault data was generated through computer simulation of the motor dynamics given in the appendix. Training of the network was completed after several training sessions. Figure 2 shows reduction of the rms error of the output nodes of the network as a function training runs. Convergence of a few of the synaptic weights are shown Figure 3.

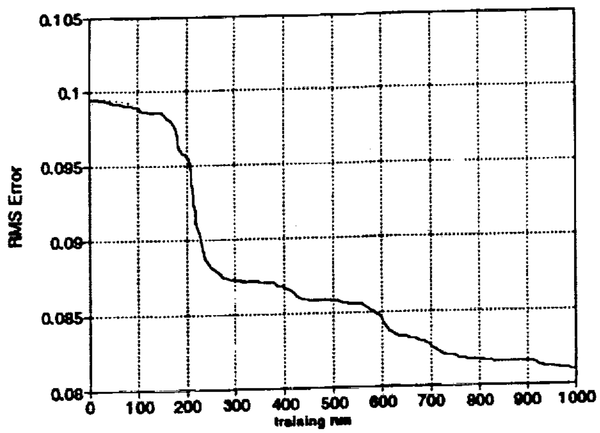

Fig. 2 Convergence of Training

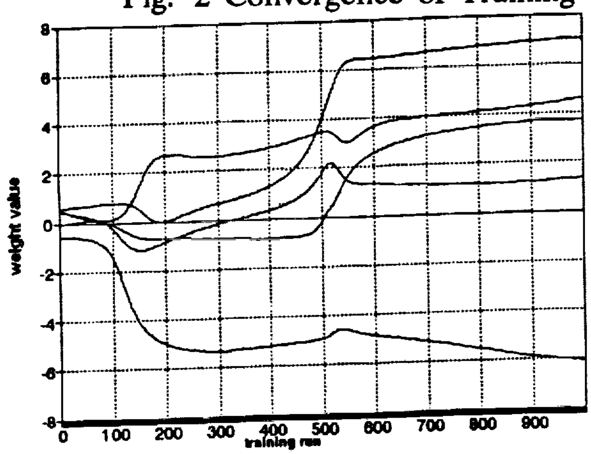

Fig. 3 Convergence of Synaptic Weights

After the completion of training, the FDI system was tested to monitor the health status of the motor. Various faults were simulated on the motor, and the fault data from the motor was applied to the input layer of the FDI network. The network compares the input pattern with the fault patterns it has been trained to recognize. The network was able to recognize the faults with 95% accuracy.

IV. CONCLUSIONS

We present the architecture of an intelligent restructurable control system that automatically identifies the occurrence of a fault, and restructures the controller for performance recovery.

The failure detection (FDI) subsystem is implemented using a neural network. The FDI network identifies the type of fault as well as its level of severity. A major advantage of this neural network based method is that it identifies faults that are characterized by changes in the numerical values of process variables as well as faults that are characterized by changes in certain physical attributes of the process.

APPENDIX

The dynamic model of a dc motor driving a load is given by

$$\frac{dI_a}{dt} = \frac{1}{L_a}V_a - \frac{R_a}{L_a}I_a - \frac{K_b}{L_a}\omega$$
$$\frac{d\omega}{dt} = \frac{K_i}{J_m}I_a - \frac{B}{J_m}\omega - \frac{1}{J_m}T_{\mathrm{load}}$$
$$\frac{dT}{dt} = \frac{R_a}{C_{\mathrm{Th}}}I_a^2 - \frac{1}{R_{\mathrm{Th}}C_{\mathrm{Th}}}T + \frac{1}{R_{\mathrm{Th}}C_{\mathrm{Th}}}T_{\mathrm{ambient}}$$

where

R_a, L_a : Armature resistance, Inductance
I_a : Armature current
ω, T : Motor Speed, Temperature
K_i, K_b : Constants
C_{Th} : Thermal capacitance
J_m, B : Motor Inertia, friction coefficient
T_{Load} : Motor Load
R_{Th} : Rotor Surface heat transfer coefficient

VI. REFERENCES

[1] Isermann R., "Failure Detection Methods for the Supervision of Technical Processes", *Process Automation*, Vol. 1, pp. 36-44, 1981.

[2] Saedtler E., "Hypothesis Testing and System identification Methods for On-line Vibration Monitoring of Nuclear Power Reactors", *5th IFAC Symp. on Identification and System Parameter Estimation*, 1979.

[3] Williams M.M. and R. Sher, "*Progress in Nuclear Energy*, Vol. 1, Pergamon Press, Oxford, 1979.

[4] Mehra R.K. and J. Peschon, "An Innovation Approach to Fault Detection and Diagnosis in Dynamic Systems" *Automatica*, Vol. 7, 1971.

[5] Montgomery R.C. and A.K. Caglyan, "A Self Reorganizing Digital Flight Control System for Aircraft", *AIAA 12th Aerospace and Science Meeting*, Washington, 1974.

[6] Willisky A.S. and H.L. Jones, "A Generalized Likelihood Ratio Approach to State Estimation in Linear Systems subject to Abrupt Changes", *Proc. IEEE Conf. on decision and Control*, Phoenix, Arizona, 1974.

[7] Pau L.F., *Failure Diagnosis and Performance Monitoring*, Marcel Dekker, New York, 1981.

[8] Chow M. and S.O. Yee,"Application of Neural Networks to Incipient Fault Detection in Induction Motors", *Journal of Neural Network Comput.*, Vol. 2, pp. 26-32, 1991.

[9] Chow M., Peter M. Mangum, and S.O. Yee, "A Neural network Approach to Real-Time Condition Monitoring of Induction Motors", *IEEE Trans. on Industrial Electronics*, Vol. 38, pp. 448-453, 1991.

[10] N. Palumbo, S.K. Biswas and B.P. Butz, "Recovery of Close-to-Nominal Pre-Fault Performance Using the Pseudo-Inverse Eigenstructure Assignment Method", *1992 American Control Conference*, 1992.

[11] Lippmann R.P., "An Introduction to Computing with Neural Nets", *IEEE ASSP Magazine*, pp. 4-22, April 1987.

[12] Kohonen, T., An Introduction to Neural Computing, *Neural Networks*, Vol. 1, pp. 3-16, 1988.

[13] Helferty J.J., Biswas S.K. and Maund Mark,"Experiments in Adaptive Control using Neural Networks", *Proc. IEEE International Conf. on Systems Engineering*, Pittsburgh, PA, August, 1990.

Knowledge Engineering