A STOCHASTIC APPROACH TO ACCIDENT IDENTIFICATION IN NUCLEAR POWER PLANTS

Kee-Choon Kwon, Soon-Ja Song, Won-Man Park, Jae-Chang Park, Chang-Shik Ham
Korea Atomic Energy Research Institute
P.O. Box 105, Yusong, Taejon, 305-600, Korea
Tel. 82-42-868-2926 Fax. 82-42-868-8357
E-mail:kckwon@nanum.kaeri.re.kr

ABSTRACT

Identification of the types of accidents and proper actions is required at an early stage of an accident in nuclear power plants. The accident of the plant can be identified by their symptom patterns related to the principal variables and operating status of major equipment. The patterns are identified by the Self-Organizing Feature Map (SOFM), unsupervised artificial neural network, for feature mapping algorithm and the Hidden Markov Model (HMM), a stochastic technique for solving the time series problem. The off-line data from a compact nuclear simulator are vector quantized by SOFM clustering algorithm. The HMM is created for each accident from a set of training data which are the result of vector quantization. The accident identification is decided by calculating which model has the highest probability for given test data. The system uses a left-to-right model including 6 states and 16 input variables to identify 7 types of accidents and the normal state. The HMM is trained by the maximum-likelihood estimation method which uses forward-backward algorithm and Baum-Welch re-estimation algorithm. The optimal path for each model at the given observation is found by Viterbi algorithm, and then the probability of optimal path is calculated. The simulation results show that the proposed system identifies the accident types correctly. It is also shown that the diagnosis is performed well for incomplete input observation caused by sensor fault or malfunction of certain equipment.

I. INTRODUCTION

The term diagnosis, as applied to an engineering system or process, means the determination of the cause which brought about an undesirable state or failure of the system. The diagnosis can be done at several different levels, e.g. component, subsystem, function or event.[1] At the proposed accident identification system, diagnosis is made at event level to determine which accident has occurred. This system is intended to support operator's decision-making by interpreting major plant variables and operating status of the plant. The accident identification system has been developed using the techniques of rule-based system, artificial neural network, and fuzzy theory. But most of them are under test with test facility, not applied to operating Nuclear Power Plant (NPP).[2,3,4]

Accidents in a NPP are associated with unique patterns of major variables and equipment status; hence, diagnosis can be treated as pattern classification. A pattern is a quantitative description of objects, events, or phenomena. The classification may involve spatial and temporal patterns. Temporal patterns usually involve ordered sequences of data appearing in time. The goal of pattern classification is to assign a physical object, event, or phenomenon to one of the prespecified classes[11]. The same accident may occur under different conditions such as full power or half power, so it is possible to adopt a stochastic approach for classification of the patterns. Hidden Markov Model (HMM) is such a stochastic technique to solve the identification problems associated with time series, such as speech signal or plant process signal.

II. DEVELOPMENT OF ACCIDENT IDENTIFICATION SYSTEM

1. An Outline of the System

Identification of unknown patterns X corresponds to finding optimal model $\hat{W}$ that maximizes the conditional probability $P(W|X)$ over the types of accident W. We can apply Bayes rule,

$$P(\hat{W}|X) = \max_{W} \frac{P(X|W)P(W)}{P(X)}$$

The conditional probability $P(X|W)$ comes from comparing shapes of the accident models with input observations while *a priori* probability $P(W)$ comes from the accident model which represents how often the accident appears in NPP. Since $P(W)$ is independent of W, we get

$$\begin{aligned} P(\hat{W}|X) &\propto P(X|\hat{W})P(\hat{W}) \\ &= \max_{W} [P(X|W)P(W)] \end{aligned}$$

It is difficult to calculate *a priori* probability $P(W)$ in NPP, so we can assume the probabilities of all accidents occurring are equal. In this paper, HMM is used to estimate the conditional probability $P(W|X)$. An HMM is trained for each accident from a set of training data. Incoming observations are recognized by calculating which model has the highest probability for producing that observation.

The training and test data are provided off-line from a compact nuclear simulator. Major variables and equipment status are combined for input symptom vector in each accident when the simulator emulates an accident situation. The collected input symptoms are vector quantized for feature extraction, which means N-dimensional measurement space transmitted to 1-dimensional feature space. Vector quantized code book is the input of the HMM identifier which is already trained. The final decision is the result of comparing with an HMM identifier to determine which model has the highest probability. <Fig. 1> shows a block diagram of the accident identification system.

The accidents are simulated in a compact nuclear simulator by activating malfunctions during normal operation. We then get parameters, such as temperature, pressure, flow, pump status, or valve open/close. The accidents that can be diagnosed by the accident identification system are:

Accident 1 ; ATWS (Anticipated Transient Without Scram)
Accident 2 ; LOCA (small break Loss Of Coolant Accident)
Accident 3 ; SGTR (Steam Generator Tube Rupture)
Accident 4 ; RCPT (Reactor Coolant Pump Trip)
Accident 5 ; FWLB (FeedWater Line Break inside containment)
Accident 6 ; MSLB (Main Steam Line Break)
Accident 7 ; PORV (Power Operated Relief Valve stuck open)

The input symptom vectors are a collection of principal variables and status of major equipment from accident in the compact nuclear simulator. The following 16 major variables and equipment status are used to identify the seven different types of accidents and one normal state.

(1) pressurizer pressure
(2) pressurizer level
(3) reactor coolant average temperature
(4) reactor coolant flow
(5) steam generator pressure
(6) steam generator level
(7) main steam flow
(8) main steam pressure
(9) reactor power
(10) turbine power
(11) main feedwater pump flow
(12) main steam enthalpy
(13) fuel temperature
(14) status of reactor coolant pump
(15) reactor trip signal
(16) status of main steam isolation valve.

2. Self-Organizing Feature Map[10,11]

Feature extraction is an important task both for classification or recognition and is often necessary as a preprocessing stage of data. In this way data can be transformed from high-dimensional pattern space to low-dimensional feature space. Our goal is to identify a neural architecture that can learn feature mapping without supervision. The feature mapping algorithm is supposed to convert patterns of arbitrary dimensionality into the response of one-dimensional array of neurons.[11]

Suppose that an input pattern has N features and is represented by a vector x in an N-dimensional pattern space. The network maps the input patterns to an output space. The output space in this case is assumed to be one-dimensional array of output nodes, which possess a certain topological orderness. The question is how to train a network so that the ordered relationship can be preserved. Kohonen proposed to allow the output nodes to interact laterally, leading to the Self-Organizing Feature Map (SOFM), in other words Kohonen network. A simple configuration of the SOFM is illustrated in <Fig. 2>.

The most prominent feature of the SOFM is the concept of excitatory learning within a neighborhood around the winning neuron. The size of the neighborhood

slowly decreases with each iteration. A more detailed description of the training phase is provided below:

1) First, a winning neuron is selected as the one with the shortest Euclidean distance

$$\| x - w_{i^*} \| = \min_i \| x - w_i \|$$

between its weight vector and the input vector, where w_i denotes the weight vector corresponding to the ith output neuron.

2) Let i^* denote the index of the winner and let I^* denote a set of indices corresponding to a defined neighborhood of winner i^*. Then the weights associated with the winner and its neighboring neurons are updated by

$$\Delta w_j = \eta(x - w_j)$$

for all the indices $j \in I^*$, and η is a small positive learning rate. The amount of updating may be weighted according to a preassigned "neighborhood function", $\Lambda(j, i^*)$.

$$\Delta w_j = \eta \, \Lambda(j, i^*)(x - w_j)$$

for all j. For example, a neighborhood function $\Lambda(j, i^*)$ may be chosen as

$$\Lambda(j, i^*) = \exp(-|r_j - r_{i^*}|^2 / 2\sigma^2)$$

where r_j represents the position of the neuron j in the output space. The convergence of the feature map depends on a proper choice of η. One plausible choice is that $\eta = 1/t$. The size of neighborhood should decrease gradually.

3) The weight update should be immediately succeeded by the normalization of w_i.

In the retrieving phase, all the output neurons calculate the Euclidean distance between the weights and the input vector, and the winning neuron is the one with the shortest distance.[12]

3. Application of Hidden Markov Model

3.1 Hidden Markov Model [6,7,8]

The basic theory of HMM was introduced in late 1960s and implemented for continuous speech recognition in mid 1970s. After this application, HMM has been successfully applied to real problems which can not be solved by conventional Markov models. HMM has advantages that provide proper solutions by modeling and learning by itself even if it does not have exact knowledge about problem solving.

HMM is represented by a graph structure which consists of N nodes called state and arcs that means a directional transition between nodes. In a graph, the observation symbol probability distribution which models spatial characteristics and initial state probability distribution stored in a node, and state transition probability distribution which models time characteristics stored in an arc. HMM states are not directly observable, and can be observed only through a sequence of observed symbols.

To describe the HMM formally, the following model notation for an HMM can be used.

N : the number of states in the model

L : the number of distinct observation symbols per transition, denote the set of individual symbols $V = \{v_1, v_2, \cdots, v_L\}$

T : the lengths of the observation sequence, $O_1, O_2, \cdots, O_T$

S = { s_t } : a set of states, $s_t \in \{1,2,\cdots,N\}$, $t = 1,2,\cdots,T$,
state i at time t may be denoted by $s_t = i$

$A = \{a_{ij}\}$: the state transition probability distribution, $i, j = 1, 2, \ldots N$,
$a_{ij} = P(s_{t+1} = j | s_t = i)$: the transition probability from state i to state j, the parameter should satisfy the stochastic constraint $\sum_i a_{ij} = 1$

$B = \{b_j(k)\}$: the observation symbol probability distribution,
$k = 1,2,\cdots,L$, $b_j(k) = P(v_k | s_{t+1} = j)$:
observation probability of kth symbol v_k in transition a_{ij}, the parameter should satisfy the stochastic constraint $\sum_k b_j(k) = 1$

$\pi = \{\pi_i\}$: the initial state distribution, where
$\pi_i = \Pr(s_i = i)$, $i = 1,2,\cdots,N$ satisfying $\sum_i \pi_i = 1$

S_I : a set of initial states,
S_F : a set of final states,
N_I : the number of initial states,
N_F : the number of final states.

An HMM can be represented by the compact notation $\lambda = (A, B, \pi)$. Specification of an HMM involves the choice of the number of states, N, the number of discrete symbols, L, and the specification of three probability densities with matrix form, A, B and π.

3.2 Training and Identification

Training means that the characteristics of input patterns to be modeled by the parameter of $\lambda = (A, B, \pi)$. An HMM is applied to an identification problem under the assumption that we can precisely determine the model parameters for given observations. But it is difficult that this assumption is exactly realized because of the complexity of problem and having local optimal not global optimal. At present, we are satisfied to find local optimal in parameter optimization methods. In this paper, we use the maximum likelihood estimation for training. This method maximizes the following equation given input observations O.

$$P(O|\lambda) = \sum_{S} \pi_{s_0} \prod_{t=1}^{T} a_{s_{t-1}s_t} b_{s_{t-1}s_t}(O_t)$$

It calculates the probability of given observation symbols at all paths from initial to final state. The model parameter which maximizes the above equation is efficiently computed by forward-backward algorithm and Baum-Welch re-estimation algorithm. The forward variable $\alpha_t(i)$ can be defined :

$$\alpha_t(i) = P(O_1, O_2, \cdots O_t, s_t = i|\lambda)$$

This is actually the probability of the partial observation sequence to time t and state i which is reached at time t for a given λ. We can calculate $\alpha_t(i)$ by forward algorithm as follows:[7]

Step 1: $\alpha_1(i) = \pi_i b_i(O_1)$, for all states i (if $i \in S_I$ $\pi_i = \frac{1}{N_I}$; otherwise $\pi_i = 0$)

Step 2: Calculating $\alpha()$ along the time axis, for $t = 2, \cdots, T$, and all states j, compute:

$$\alpha_t(j) = [\sum_i \alpha_{t-1}(i) a_{ij}] b_j(O_t)$$

Step 3: Final probability is given by: $P(O|\lambda) = \sum_{i \in S_F} \alpha_T(i)$

The backward variable $\beta_t(i)$ which is used to optimize the model parameter with forward variable, can be defined as :

$$\beta_t(i) = P(O_{t+1}, O_{t+2}, \cdots O_T | s_t = i, \lambda)$$

i.e. the probability of the partial observation sequence from $t+1$ to the final observation T, given state i at time t and the model λ and $\beta_t(i)$ can be calculated by backward algorithm as follows:[7]

Step 1: $\beta_T(i) = \frac{1}{N_F}$, for all states $i \in S_F$, otherwise $\beta_T(i) = 0$

Step 2: Calculating $\beta()$ along the time axis, for $t = T-1, T-2, \cdots, 1$ and all states j, compute:

$$\beta_t(j) = [\sum_i a_{ji} b_i(O_{t+1}) \beta_{t+1}(i)]$$

Step 3: Final probability is given by: $P(O|\lambda) = \sum_{i \in S_I} \pi_i b_i(O_1) \beta_1(i)$

The most difficult problem in HMM is how to adjust the model parameters (A, B, π) to maximize the probability of the observation sequence given a model. The iterative algorithm used in HMM-based recognition is known as the Baum-Welch algorithm. The *a posterior* probability of transitions, $\gamma_t(i, j)$, will be defined as the probability of a path being in state i at time t and making a transition to state j at time $t+1$, given the observation sequence and the particular model. It can be computed as :

$$\gamma_t(i,j) = P(s_t = i, s_{t+1} = j|O,\lambda)$$
$$= \frac{\alpha_t(i)a_{ij}b_j(O_{t-1})\beta_{t-1}(j)}{P(O|\lambda)}$$
$$= \frac{\alpha_t(i)a_{ij}b_j(O_{t-1})\beta_{t-1}(j)}{\sum_{k\in S_T}\alpha_T(k)}$$

Similarly, *a posterior* probability of being in state i at time t, $\gamma_t(i)$, given the observation sequence and model is

$$\gamma_t(i) = P(s_t = i|O,\lambda) = \frac{\alpha_t(i)\beta_t(i)}{\sum_{k\in S_T}\alpha_T(k)}$$

At this point, $\overline{a_{ij}}$, $\overline{b_j}$, $\overline{\pi_i}$ of re-estimated new model $\overline{\lambda}$ can be computed as:

$$\overline{a_{ij}} = \frac{\sum_{t=1}^{T-1}\gamma_t(i,j)}{\sum_{t=1}^{T-1}\sum_j\gamma_t(i,j)} = \frac{\sum_{t=1}^{T-1}\gamma_t(i,j)}{\sum_{t=1}^{T-1}\gamma_t(i)}$$

$$\overline{b_j}(k) = \frac{\sum_{t\in O_t=v_k}\gamma_t(j)}{\sum_{t=1}^{T}\gamma_t(j)}$$

$$\overline{\pi_i} = \gamma_1(i)$$

Thus, if $\overline{\lambda}$ is iteratively to replace λ and repeat the above re-estimation calculation, it can be guaranteed that $P(O|\lambda)$ can be improved until some limiting point is reached.

Identification or recognition means to find the best path in each model and select the one which maximizes the path probability for a given input observation. There are several possible ways to find the optimal state sequence associated with the given observation sequence. One possible optimality criterion is to choose the states, s_t, which are in the best path with the highest probability, i.e. with maximum $P(O,S|\lambda)$. A formal technique for finding this single best state sequence is called the Viterbi algorithm, which works as follows: [7,8]

Step 1: Initialization. For all states i,
$\delta_1(i) = \pi_i b_i(O_1)$
$\psi_1(i) = 0$

Step 2: Recursion. From time t=2 to T, for all states j,

$$\delta_t(j) = \max_i[\delta_{t-1}(i)a_{ij}]b_j(O_t)$$

$$\psi_t(j) = \arg\max_i[\delta_{t-1}(i)a_{ij}]$$

Step 3: Termination. (* indicate the optimized results)

$$P^* = \max_{s \in S_T}[\delta_T(s)]$$

$$s^*_T = \arg\max_{s \in S_T}[\delta_T(s)]$$

Step 4: Path (state sequence) backtracking. From time T-1 to 1

$$s^*_t = \psi_{t+1}(s^*_{t+1})$$

4. Design of a Prototype Accident Identification System

At first, we collect 10 training and test data per accident from the compact nuclear simulator. The collected training and test data are vector quantized by the SOFM clustering algorithm as shown in <Fig. 1>. The SOFM, unsupervised artificial neural network, clustered input vector into M disjoint sets. In our implementation, we chose 84 clusters for an optimal solution after several attempts. It means every input vector is assigned to one of 84 clusters. <Fig. 3> shows the distribution of 84 clusters given 682 input vectors. The code book size is 15, this means this system receives 15 time interval input vectors.

A left-to-right HMM has been considered appropriate for processing those signals whose properties change over time.[8] The underlying state sequence associated with the model has the property that as time increases the state index increases (or stays the same), i.e. the state proceeds from left to right. This model consists of 6 states which have less than 2 direct transitions to the right state; the state transition probability a_{ij} satisfies the following condition :

$a_{ij} = 0$ for $j < i$ or $j \geq i+3$

Few initial conditions are given to this model, and these initial condition are equivalent to all accident models. When the transition is occurred to 3 ways, the initial value of a_{ij} are 0.333, to 2 ways like just before last state, the initial values of a_{ij} are 0.5, to 1 way like last state, the initial value of a_{ij} is 1.0. By assuming that the observation symbol probability is equivalent to each state, observation symbol probability $b_j(v_k) = 0.0119$ when satisfy the equation

$$b_j(k) = \frac{1}{L}, \quad \sum_{k=1}^{L} b_j(k) = 1 \text{ and } L = 84.$$

Initialized HMM is illustrated in <Fig. 4>. In this system, we use 8 models for 7 types of accidents and 1 normal model.

The training is performed by calculating α value from forward algorithm, β value from backward algorithm, and re-estimate from Baum-Welch algorithm in each model given multiple input observations. The re-estimation is done until the convergence condition, $P(O|\hat{\lambda}) \geq P(O|\lambda)$, is satisfied in each model.

The probability, $P(O|\lambda)$, is calculated by the optimal path which is obtained by Viterbi algorithm for given input observations in each model. We identify the accident by examining which model has the highest probability for given input observations.

5. The Result of Experiment

The experiments were carried out off line in HP715/33 workstation the programming done with "C" language. <Table 1> shows the best path probability of each model which is the result of Viterbi algorithm when the input observation is the same as training data. In this case, model #2 accident, ATWS, has the highest probability. In the case of one sensor fault or equipment malfunction, the accident identification system also exactly identified the accident as shown in <Table 2> for model #4 accident, SGTR. However, it could not correctly identify the accident when more than 2 sensor faults or equipment malfunctions have occurred.

III. CONCLUSION

We proposed a prototype accident identification system based on the stochastic modeling approach of HMM. We can identify accidents by recognizing the patterns of accidents, which is a new attempt to expand the application area of HMM to accident identification. After a proper training using the train vector, the prototype system exactly identifies the accidents from input observations. At present, this prototype system is implemented off-line, but it should be implemented on-line to accept continuous time series data in real time. There are a lot of problems which should be solved before its actual application to an operating plant. However, this system have advantages such as an easy expansion of accident types and observation symbol sequences, and a relatively short time for training. Further effort is being made to improve the system.

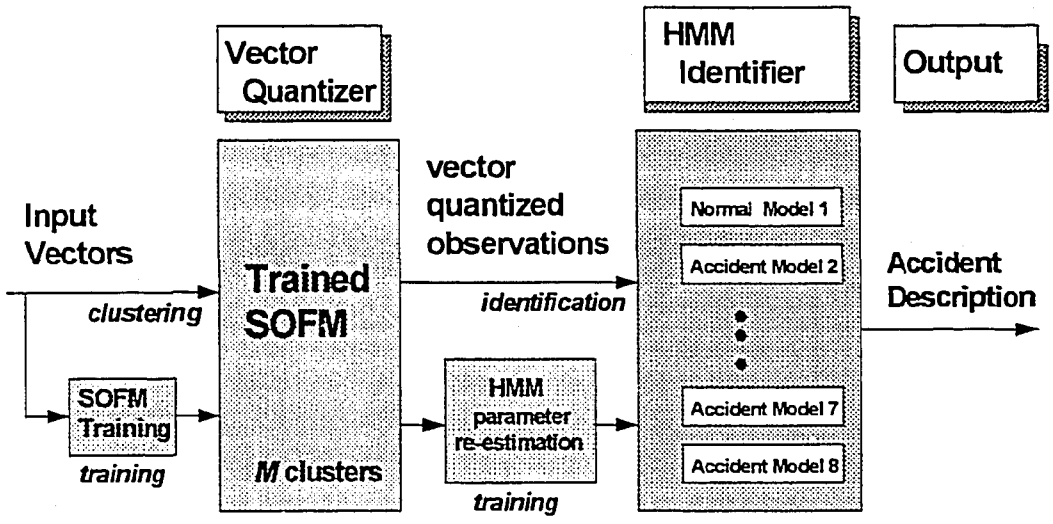
ACKNOWLEDGMENTS

We acknowledge the financial support of the Ministry of Science and Technology (MOST) for this work.

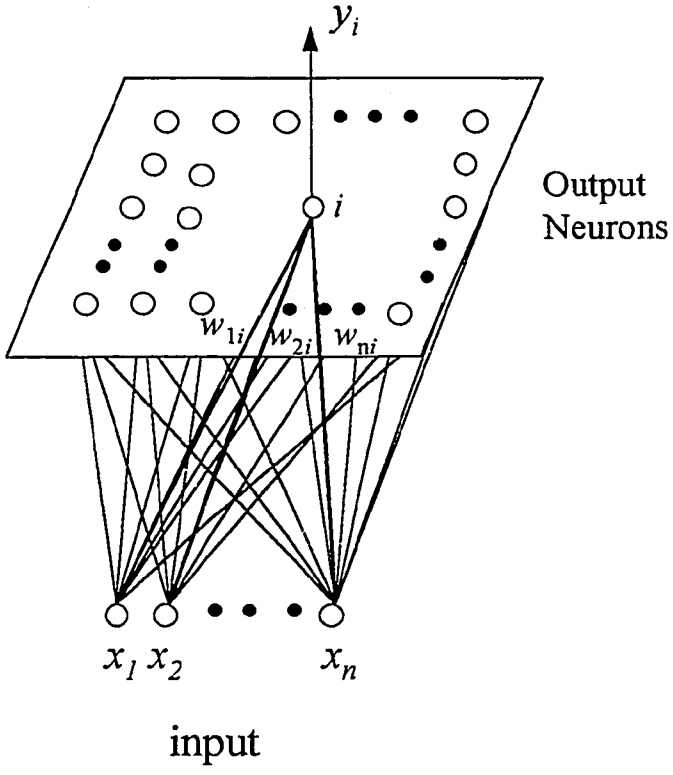
REFERENCES

1. I. S. Kim, "Computerized Systems for On-line Management of Failures: A State-of-the-Art Discussion of Alarm Systems and Diagnostic Systems Applied in the Nuclear Industry," *Reliability Engineering and System Safety*, Vol. 44, pp. 279-295, 1994.
2. S. W. Cheon, *A Study on the Application of Expert Systems and Neural Networks for the Development of Operator Support Systems in Nuclear Power Plants*, Ph. D Thesis, Korea Advance Institute of Science and Technology, 1993.
3. A. Ikonomopoulos, et al., "A Hybrid Neural Network-Fuzzy Logic Approach to Nuclear Power Plant Accident Identification," *AI91 Frontiers Innovative Computing for the Nuclear Industry*, Jackson, Wyoming, pp. 217-226, Sep. 15-18, 1991.

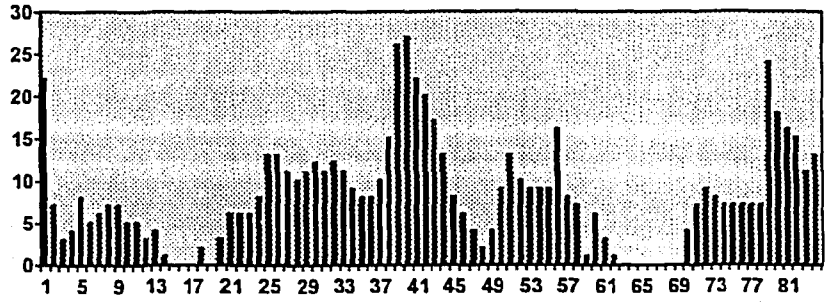
4. S.W. Cheon, S.H. Chang, "Application of Neural Networks to a Connectionist Expert System for Accident Identification in Nuclear Power Plants," *Nuclear Technology*, Vol. 102, pp. 177-191, May 1993.
5. K.C. Kwon, et al., "An Application of Hidden Markov Model to Transient Identification in Nuclear Power Plants," ANS Embedded Topical Meeting on Computer-Based Human Support Systems, pp. 275-280, Philadelphia, June 25-29, 1995.
6. J.Y. Ha, *Unconstrained Handwritten Word Recognition with Interconnected Hidden Markov Models*, Ph.D Thesis, KAIST Department of Computer Science, 1994.
7. X.D. Huang, Y. Ariki & M.A. Jack, Hidden Markov Models for Speech Recognition, Edinburgh University Press, 1990.
8. L.R. Rabiner, "A Tutorial on Hidden Markov Models and Selected Application in Speech Recognition," *Proceedings of the IEEE*, Vol. 77, No. 2, pp. 257-285, 1989.
9. P. Smith, "Hidden Markov Models for Fault Detection in Dynamic Systems," Pattern Recognition, Vol. 27, No. 1, pp. 149-164, 1994.
10. T. Starner, *Visual Recognition of American Sign Language Using Hidden Markov Models*, MS Thesis, MIT, Feb. 1995.
11. J. Zurada, Intriduction to Artificial Neural Systems, West Info Access, 1992.
12. Y. Kung, Digital Neural Networks, Prentice Hall, 1993.
13. J. Freeman, D. Skapura, Neural Networks Algorithms, Applications, and Programming Techniques, Addison-Wesley, 1991.



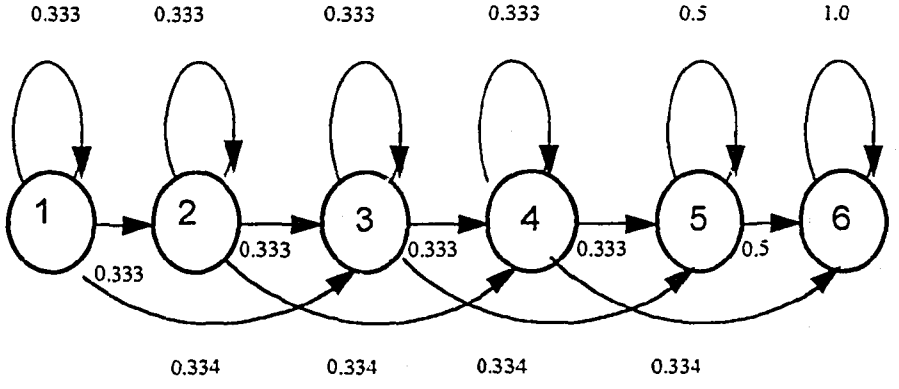
<Fig. 1> Block Diagram of the Accident Identification System



<Fig. 2> A Network for Self-Organizing Feature Map



<Fig. 3> Cluster Distribution by SOFM



<Fig. 4> Initialized HMM with 6 states

<Table 1> Likelihood probability (Case I)

Accident Model	Likelihood Probability
Normal	0.0
ATWS	9.506e-30
LOCA	3.547e-35
SGTR	2.232e-39
RCPT	1.398e-42
FWLB	1.447e-43
MSLB	8.668e-43
PORV	3.283e-44

<Table 2> Likelihood probability (Case II)

Accident Model	Likelihood Probability
Normal	0.0
ATWS	0.0
LOCA	6.699e-43
SGTR	2.153e-35
RCPT	2.675e-38
FWLB	2.209e-39
MSLB	2.131e-41
PORV	5.993e-43