ELEVATOR TRAFFIC PATTERN RECOGNITION BY ARTIFICIAL NEURAL NETWORK

Albert T.P. So*, J.R. Beebe**, W.L. Chan*, S.K. Liu***

* Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong
** Omnia Ltd., England
*** Chevalier (HK) Ltd.

ABSTRACT

Elevator control is ordered hierarchically: Single Car Control (SCC); Group Supervisory Control (GSC) and Building Supervisory Control (BSC). The BSC is well suited to the application of Artificial Neural Networks (ANNs). They are used to learn the traffic patterns of the building for the subsequent tuning of GSC parameters. Several parameters, such as number of car stops, number of landing calls and their ratios and changes in car weights etc., can be used as input variables to the ANN. Initially, supervised learning mode is used and later unsupervised learning is employed. The ANN provides settings for "weightings" that describe the proportion of each type of traffic pattern within a mixed case which normally exists in real practice in a modern commercial building. Five types of traffic patterns, namely up-peak, down-peak, peak demand floor, four-way traffic and off-peak, can be recognised by the ANN.

1 INTRODUCTION

It is generally aware that vertical transportation within a building is one of the most important means of transportation for almost every citizen. A highly efficient and intelligent supervisory control system for a lift group is a necessity for the excellent services provided within a modern building and in particular, the commercial buildings. However, as a rule of thumb around the world, the up-peak mode of traffic is usually used for the initial design of an elevator system because if the up-peak traffic pattern is sized correctly, generally all other traffic patterns will be adequately served [1]. Therefore, most existing elevator systems can perform well during the up-peak traffic situation and they fail to be optimal during other traffic conditions. A comprehensive supervisory control system is the only way to make the system adapt to different kinds of traffic patterns during the operational period whole day long so that the system can work optimally. However, it is very difficult to identify the existence of a particular traffic pattern since different patterns appear simultaneously though one may be the dominating factor. This uncertainty and fuzziness were handled manually in the past. In this paper, we describe an ANN based system that can automatically identify major traffic patterns so that both the quality and quantity of services can be enhanced.

Elevator control is imposed at different levels which are ordered hierarchically. At the lowest level, the Single Car Controller (SCC) is responsible for starting and stopping the lift car, opening and shutting the car door and moving the car from floor to floor. At an intermediate level, the Group Supervisory Controller (GSC) is responsible for co-ordinating the operation of a group of lifts so as to optimally service the landing calls placed by passengers wishing to use the lifts. At the highest level, the Building Supervisory Controller (BSC) is responsible for monitoring the traffic in the building or site and making adjustments on the basis of information gathered in the long term so as to "fine-tune" the operation of the lower level controllers. The BSC is well suited to the application of ANNs. They can be used to learn the traffic patterns of a building and the optimal tuning of GSC parameters to respond to those patterns. The ANNs can be trained to provide settings for "weightings" to be applied to a "cost-based" GSC algorithm. With a deeper knowledge of the traffic patterns, intelligent control algorithms can be implemented, such as the dynamic zoning methodology [2] etc.

2 ARTIFICIAL NEURAL NETWORKS

2.1 What are neural networks?

Building intelligent systems that can model human behaviour has captured the attention of the world for years. So, it is not surprising that a technology such as neural networks has generated great interest. Neural networks are information processing systems. In general, neural networks can be though of as "black box" devices that accept inputs and produce outputs. Some of the operations that neural networks perform include [3]:

a) Classification - an input pattern is passed to the network and the network produces a representative class as output.
b) Pattern matching - an input pattern is passed to the network and the network produces the corresponding output pattern.
c) Pattern completion - an incomplete pattern is passed to the network and the network produces an output pattern that has the missing portions of the input pattern filled in.
d) Noise removal - a noise-corrupted input pattern is presented to the network and the network removes some or all of the noise and produces a cleaner version of the input pattern as output.
e) Optimization - an input pattern representing the initial values for a specific optimization problem is presented to the network and the network produces a set of variables that represents a solution to the problem.
f) Control - an input pattern represents the current state of a controller and the desired response for the controller and the output is the proper command sequence that will create the desired response.

Neural networks consist of processing elements, i.e. neurons, and weighted connections. The processing elements (PEs) are formatted in layers. Each PE in a neural network collects the values from all its input connections, performs a predefined mathematical operation (typically a dot product followed by a PE function) and produces a single output value. The Sigmoid (S-shaped) function, f(x), which is a bounded, monotonic, non-decreasing function that provides a graded, nonlinear response within a pre-specified range, has been the most popular PE function used for ANNs.

$$f(x) \triangleq \frac{1}{1+e^{-\alpha x}} \qquad \textit{where } \alpha > 0 \ (\textit{usually} = 1) \tag{1}$$

2.2 Neural learning by backward propagation

Arguably, the most appealing quality of neural networks is their ability to learn. Learning is defined as a change in connection weight values that results in the capture of information that can later be recalled. Several procedures are available for changing the values of connection weights. Hebbian learning [4] is the simplest form of adjusting connection-weight values in an ANN. Some ANNs have learning algorithms designed to produce, as a set of weights, the principal components of the input data patterns [5], the learning mode being entitled principal component learning. Hebbian learning has later been extended to capture the temporal changes that occur in pattern sequences. This learning law, called differential Hebbian learning, has been independently derived by Klopf [6] in the discrete-time form and by Kosko [7] in the continuous-timeform. Competitive learning, introduced by Grossberg [8,9] is a method of automatically creating classes for a set of input patterns. There are others such as Min-Max Learning [10], Reinforcement Learning [11], Stochastic Learning [12] and Hardwired Systems [13] etc. One very popular learning method that is employed in our project is called Error-Correction Learning and it has another name called backward propagation.

A section of a multiple layers trainable ANN is shown in Fig. 1 for illustration, including the (M-1)th, Mth layers and the output layer which will be connected to the (M+1)th layer. Each link has a weight that can be changed so as to improve the net's ability to produce the correct outputs for input combinations in the training set. The desirable outputs, t_j's, are compared with the actual outputs, y_j's, and by minimising this error term, E, it is possible to tune the weights of the neural network.

$$E = \frac{1}{2}\sum_j (t_j - y_j)^2$$

$$\textit{where} \quad y_j = \gamma_j\left[\sum_i \left(w_{ji}^{(M)} u_i^{(M-1)} + \theta_j^{(M)}\right)\right] = \gamma_j\left(n_j^{(M)}\right) \tag{2}$$

$$n_j^{(M)} = \sum_i \left[w_{ji}^{(M)} u_i^{(M-1)} + \theta_j^{(M)}\right]$$

$$\gamma_j(x) \triangleq \frac{1}{1+e^{-x}}$$

In equation (2), w_{ji} is the weight between node n_j at the Mth layer and node u_i at the

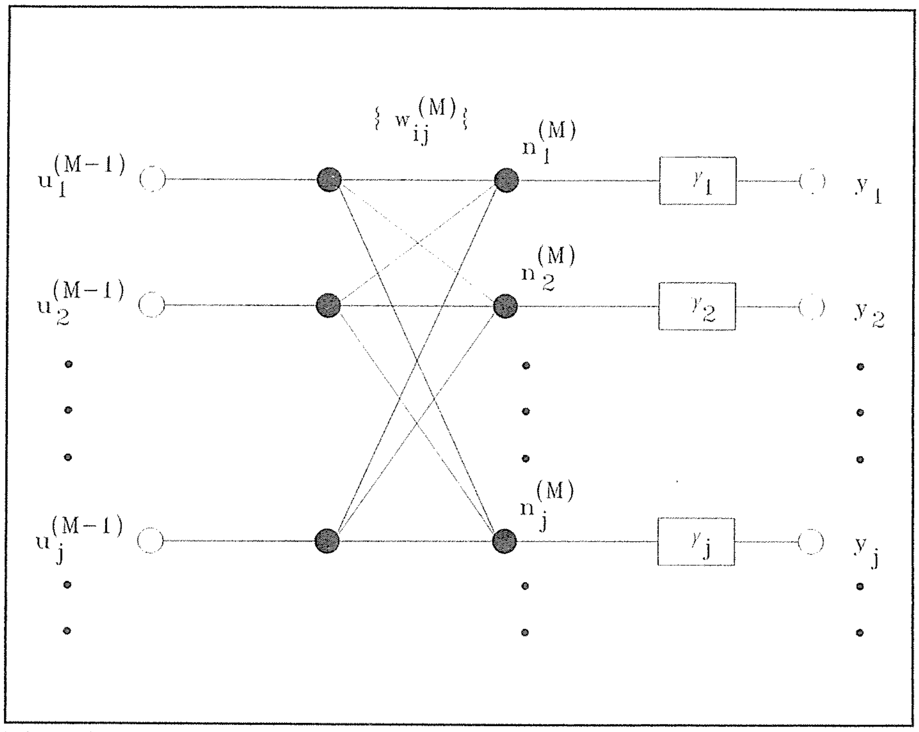

Fig. 1

(M-1)th layer while θ_j is the bias associated with node n_j at the Mth layer. The goal of training is to minimise E. The partial derivative of E with respect to each weight is obtained as shown below.

$$\frac{\partial E}{\partial w_{ji}^{(M)}} = \frac{\partial E}{\partial n_j^{(M)}} \frac{\partial n_j^{(M)}}{\partial w_{ji}^{(M)}} = \delta_j^{(M)} u_i^{(M-1)}$$

$$where \quad \delta_j^{(M)} = \frac{\partial E}{\partial n_j^{(M)}} = \frac{\partial E}{\partial y_j} \frac{\partial y_j}{\partial n_j^{(M)}} = (y_i - t_j) \gamma_j' (n_j^{(M)}) \tag{3}$$

The well-known "Steepest Slope Descent" method [14] can be applied here to search for the best $w_{ji}^{(M)}$. At the kth iteration, we have the following equation.

$$w_{ji}^{(M)} (k+1) = w_{ji}^{(M)} (k) + \Delta\, w_{ji}^{(M)} (k+1)$$

$$where \quad \Delta\, w_{ji}^{(M)} (k+1) = \eta\, \delta_j^{(M)} u_i^{(M-1)} \tag{4}$$

In order to adjust the converging rate, an accelerating factor, α, can be incorporated into the iterative equation.

$$\Delta\, w_{ji}^{(M)} = \eta\, \delta_j^{(M)} u_j^{(M-1)} + \alpha\, w_{ji}^{(M)} (k) \tag{5}$$

3 THE ANN BASED ELEVATOR TRAFFIC PATTERN RECOGNITION SYSTEM

3.1 The real-time lift system simulator

In order to generate suitable data records for training the ANN, a real-time simulation program on a standard 486 PC has been prepared, entitled "sim.exe". The following parameters can be freely chosen for different simulation conditions:
a) number of lifts in the system
b) number of storeys in the building
c) interfloor height of the building
d) contract capacity of each lift car (P)
e) contract speed of each lift car
f) acceleration and deceleration of each lift car
g) door opening/closing time
h) passenger transfer time
i) floor queue maximum length
j) stop coefficient
k) distance coefficient
The last two items are used for lift car assignment in response to a landing call. This simulation program generates data files, entitled "raw.dat", for each lift car and each floor with a frequency of 1 second. Four parameters are associated with each lift car and two parameters are associated with each landing:
a) lift position, correct to integral floor number, 0 for main terminal
b) lift direction (-1 for down, 0 for stationery, 1 for up)
c) lift weight (0 to P*75 kg)
d) door status (0 for closed and 1 for open)
e) up landing call status for each floor (0 for no call and 1 for up)
f) down landing call status for each floor (0 for no call and 1 for down)
Based on "raw.dat", another program entitled "ana.exe" is prepared to convert the raw data into statistical parameters that can be fed into the input nodes of the ANN. It should be noted that the design format of "raw.dat" is not merely for simulation. Site measurement of a real elevator system can produce similar files, as discussed later.

3.2 The ANN structure

A fully connected ANN with nine input nodes, nine hidden nodes and five output nodes is configured. The five output nodes represent up-peak traffic, down-peak traffic, heavy demand floor, peak four-way traffic and off-peak traffic. All statistical data are normalised to within a range of [0, 1] before they are fed into the input nodes and all output nodes assume values within the same range. A value near to one at node 1 implies an up-peak traffic mode exists in the system.

Supervised learning mode is employed before the ANN is put into operation. All learning methods can generally be classified into two categories: supervised learning and unsupervised learning. Supervised learning is a process that incorporates an external teacher and/or global information. Unsupervised learning, also referred to as self-organization, is a process that incorporates no external teacher and relies upon only local information during the entire learning process. Unsupervised learning organizes presented data and discovers its emergent collective properties. Supervised learning is aided by the real-time simulation software package. The five types of traffic modes are arbitrarily arranged in the passenger data files of "sim.exe" alternatively. Initially, all weights are assigned random real numbers. Following the execution of "sim.exe", "raw.dat" files are available and they are turned into parameters that are compatible with the nine input nodes of the ANN. Each record within the "raw.dat" file represents the measurement of 1 second. Totally, 300 records (i.e. five minutes) are required to form one learning case for the ANN because what is required is statistical data rather than real-time raw data. During real operation, 120 records (i.e. 2 minutes) are needed to form one case to produce the results.

3.3 The input nodes

From the six parameters in "raw.dat", nine parameters can be derived for the nine input nodes. These nine parameters are believed to represent the five types of traffic modes distinctively.

3.3.1 Up-peak identification

For the identification of up-peak traffic condition, the ratio of the number of up-stops divided by down-stops within a prescribed time interval, say 5 minutes, reaches a marked peak during an up-peak condition, as suggested by Beebe [15,16]. This ratio is divided by the maximum value of the peak for a practical elevator system for normalisation and is then fed to the first input node of the ANN. The ratio of up-stops to up-landing calls times the number of cars within the prescribed time interval during an up-peak condition [15] yields the value of expected number of stops per round trip, S, used in the conventional up-peak traffic design calculation. The ratio is thus divided by S for normalisation and fed to the second input node of the ANN.

3.3.2 Down-peak identification

For the identification of down-peak traffic condition, the number of down-stops divided by the number of up-stops within the prescribed time interval reaches a peak during a down-peak condition, as suggested by Beebe [15,16]. This ratio is divided by the maximum value of the peak for a practical elevator system for normalisation and is then fed to the third input node of the ANN. The ratio of all stops divided by all landing calls within a prescribed time interval reaches the value 1 [15]. The reciprocal of this ratio is fed to the fourth input node of the ANN.

3.3.3 Off-peak identification

For the identification of off-peak traffic condition, the number of total stops per minute for the whole system is a good indicator. The minimum value is equal to the number of lift cars and thus the reciprocal of this number is fed to the fifth input node of the ANN.

3.3.4 Two-way and four-way traffic identifications

For the identification of two-way and four-way traffic, four new parameters need to be derived from "raw.dat". Upon any car stoppage, the monitoring system measures the change of weight as sensed by the linear transformer underneath the floor of each car. It is assumed that embarking passengers will only enter the car after all departing passengers have left the car. In this way, the weight first falls until a minimum value is attained and then it rises up again. In this way, it is possible to measure both the changes of weight when passengers leave and enter the car. Four types of weight change are derived, namely Up-in (UI), Up-out (UO), Down-in (DI) and Down-out (DO). Four registers are assigned to record the floor number where the four types of weight change are maximum within a prescribed period of time, say 3 minutes. As an example, an up-car initially stops at the ith floor and its change of weight with respect to Up-in is UI_i. Another up-car then stops at the jth floor and its change of weight with respect to Up-in is UI_j. If $UI_i > UI_j$, the register of UI will save the value of i or it will save the value of j vice versa. After the prescribed period of time, the four registers are frozen and four new registers are formed. When four sets of registers are available, the means and standard deviations of corresponding registers are calculated, resulting in eight parameter, UI_M, UO_M, DI_M, DO_M, UI_{SD}, UO_{SD}, DI_{SD} and DO_{SD}. The remaining four input nodes are fed with the following values:

$$
\begin{aligned}
Node\ 6 &= \frac{UO_M - DI_M}{2\ N} + 0.5 \\
Node\ 7 &= \frac{UI_M - DO_M}{2\ N} + 0.5 \\
Node\ 8 &= \frac{UO_{SD} + UI_{SD} + DI_{SD} + DO_{SD}}{4\ N} \\
Node\ 9 &= \frac{Overall\ SD}{N}
\end{aligned}
\qquad (6)
$$

where N = *Total number of floors*

$Overall\ SD$ = *Standard Deviation of* UI_M, UO_M, DI_M, DO_M

It is anticipated that inputs to node 6 and node 7 will approach 0.5 and input to node 8 will approach zero under a four-way traffic condition. Input to node 9 will approach zero under two-way traffic condition.

The advantage of using mean and standard deviation is to eliminate noisy data. Since the algorithm only considers floors with maximum passenger flow, it will not be disturbed by other floors with less passenger flow. Another advantage of this method is the identification of the floor numbers where two-way or four-way traffic exists because the registers actually record the floor numbers with maximum passenger flow. If the ANN recognises the existence of the two types of traffic patterns, the registers are then consulted to find out the actual floor numbers concerned.

4 IMPLEMENTATION OF THE SYSTEM

The simulation program is first executed on a pseudo building with 25 floors excluding the main terminal. Four cars with contract capacity equal to sixteen each are used to service the whole building. Interfloor distance is fixed at 4 m. Contract speed is fixed at 2 m/s. Door opening time is 0.8 s while closing time is 2 seconds. Passenger transfer time is 1.2 s. For each type of traffic, 20 minute simulation is carried out and the data is fed to the ANN. Totally, 6000 records are used to train up the ANN. Training is complete when the average squared error of each output node is smaller than 10^{-3}. It takes about 35 hours to train up the ANN by using the 6000 records. Afterwards, the ANN is used to identify patterns from new data sets and the accuracy can be up to 60%.

A monitoring system has been installed in a modern commercial building in Hong Kong. Four lift cars service the high-rise zone, from 16/F to 23/F and the main terminal. The operational data is continuously grabbed by a 8255 I/O card installed on a standard 486 PC. Signals of the car weights are grabbed by a 12-bit AD converter card. All the raw data are recorded on a 540 MB hard disk and they are converted to parameters feeding the nine input nodes of the ANN. Off-line supervised learning mode is conducted with manual recognition of the traffic patterns. It has been found that the accuracy of the resultant ANN can be up to 35% since the training has not been comprehensive enough. Variations in traffic patterns are limited for a real system.

5 CONCLUSION

The configuration and learning method of a standard artificial neural network has been described in this paper. Such ANN forms the foundation of automatic elevator traffic pattern recognition. A 9 x 9 x 5 ANN has been constructed and algorithms for converting raw data recorded from an elevator system, either operational or by simulation, to feed the input nodes of the ANN are discussed in details. It is our belief that perfect supervisory control can only be possible by a good knowledge of the traffic patterns. Thus, this ANN based recognition system can lead to a more intelligent supervisory control system. One shortfall of the ANN is the speed of learning. By using high speed supercomputer, such as i860 etc., it is possible to turn the system into real-time mode. Otherwise, off-line learning is the only choice. It is desirable that a more comprehensive ANN can be formulated to identify all traffic patterns of an elevator system although different pattern may co-exist and pollute the recognition process. The accuracy of identification is another aspect that needs improvement.

6 REFERENCES

[1] CIBSE, *CIBSE Guide D - Transportation Systems in Buildings*, CIBSE, 1993, pp. 3-3.

[2] W.L. Chan, A.T.P. So, Dynamic zoning in elevator traffic control, in G.C. Barney Ed. *Elevator Technology 6, Proc. of Elevcon'95*, IAEE, 1995 (in press).

[3] P.K. Simpson, Foundations of neural networks, in E. Sanchez-Sinencio and C. Lau Ed. *Artificial Neural Networks - Paradigms, Applications and Hardware Implementations*, IEEE, New York, 1992, pp. 3-24.

[4] J. Anderson, Two models for memory organization suing interactive traces, *Math. Biosci.*, Vol. 8, 1970, pp. 137-160.

[5] E. Oja, A simplified neuron model as a principal component analyzer, *J. Math. Biol.*, Vol. 15, 1982, pp. 267-273.

[6] A. Klopf, Drive-reinforcement model of a single neuron function: an alternative to the hebbian neuron model, in J. Denker Ed., *AIP Conf. Proc. 151: Neural Networks for Computing*, American Institute of Physics, New York, 1986, pp. 265-270.

[7] B. Kosko, Differential hebbian learning, in J. Denker Ed., *AIP Conf. Proc. 151: Neural Networks for Computing*, American Institute of Physics, New York, 1986, pp. 277-2182.

[8] S. Grossberg, Neural pattern discrimination, *J. Theoret. Biol.,* Vol. 27, 1970, pp. 291-337.

[9] S. Grossberg Ed., *Studies of Mind and Brain*, Reidel, Boston, 1982.

[10] P. Simpson, Fuzzy min-max classification with neural networks, *Proc. Int. Joint conf. Neural Networks*, Singapore, 1991, 1658-1669.

[11] R. Williams, Reinforced learning in connection to networks: a mathematical analysis, *University of California, Institute for Cognitive Science, Technical Report No. 8605*.

[12] H. Szu, Fast simulated annealing, in J. Denker Ed., *AIP Conf. Proc. 151: Neural Networks for Computing*, American Institute of Physics, New York, 1986, pp. 420-425.

[13] J. Hopfield and D. Tank, Neural computation of decisions in optimization problems, *Biiol. Dybernet.*, Vol. 52, 1985, pp. 141-152.

[14] B.D. Bunday and G.R. Garside, Eds. *Optimization Methods in Pascal*, Edward Arnold, 1987, pp. 54-56.

[15] J.R. Beebe, *Lift management*, PhD Thesis, Control Systems Centre, UMIST, 1980, pp. 114-116.

[16] A.T.P. So, H.S. Kuok, S.K. Liu, W.S. Chan and T.Y. Chow, New developments in elevator traffic analysis, in G.C. Barney, Ed. *Elevator Technology 5, Proc. Elevcon'93*, 1993, IAEE, pp. 212-221.

7 ACKNOWLEDGEMENTS

The project has been financially support by the Croucher Foundation Research Grant #905046 "Application of Neural Network on Recognition of Elevator Traffic Patterns for Intelligent Group Supervisory Control". The authors are also indebted to the support from IAEE (HK Branch).

BIOGRAPHICAL DETAILS

Dr. Albert T.P. So obtained BSc(Eng), MPhil and PhD from University of Hong Kong. After his graduation, he had been with Hong Kong Government for seven years and joined the Department of Building & Construction of City Polytechnic of Hong Kong. Now, he is a University Lecturer in the Department of Building Services Engineering of Hong Kong Polytechnic University. Dr. So is a Chartered Engineer, Members of I.E.E., C.I.B.S.E. and H.K.I.E. He is also the Regional Coordinator of IAEE, the General and Academic Secretary of IAEE (HK Branch).

Dr. J.R. Beebe obtained BSc, MSc and PhD from University of Manchester Institute of Science & Technology. After his graduation, he had been with LDP(Lift Designers), UMIST, Lift Innovations Ltd. and Unibit (Holdings). Now he is a Design Authority of Omnia Ltd. Dr. Beebe is a Chartered Engineer and Members of I.E.E. and B.C.S.

Mr. W.L. Chan obtained BSc(Eng) and MPhil from University of Hong Kong. He joined the China Light & Power Company Ltd. after his graduation. Now, he is a Lecturer in the Department of Electrical Engineering of Hong Kong Polytechnic University. Mr. Chan is a Chartered Engineer, Member of I.E.E., Member of I.E.E.E., Publication Secretary and Honourable Advisor of IAEE (HK Branch).

Mr. S.K. Liu is Assistant General Manager of Chevalier (HK) Ltd. He has received FMBA and Chartered Manager. He is a Registered Lift & Escalator Engineer and Registered Safety Officer.