

# Artificial Neural Networks Application in Duplex/Triplex Elevator Group Control System

C. Erdem Imrak

Istanbul Technical University, Turkey

*Artificial neural networks can offer the better solution to the passenger call distribution problem when compared to the conventional elevator control systems. Therefore, the application of neural networks in elevator group control system is discussed. The significance of introducing artificial neural networks is presented. Elevator group control systems with neural networks can predict the next stopping floors to stop by considering what has been learnt by processing the changes in passenger service demand pattern. This paper deals with the use of artificial neural networks for the distribution of the most suitable cars to the floors by considering the passenger service demand. Artificial neural networks are applied in Duplex/Triplex group control systems for improving passenger waiting time. The backpropagation algorithm is used for training neural networks. The elevator traffic analysis and simulation results are presented and compared to conventional elevator control systems.*

© 2008 Journal of Mechanical Engineering. All rights reserved.

**Keywords:** elevators group control systems, neural networks, backpropagation

## 0 INTRODUCTION

The service provided by the elevator system of any modern building is necessary for the efficient functioning of the building. This service has to be not only reliable, but also satisfactory to the passengers. Elevator group control systems can be utilized to control the cars as a group and convey the passengers to their destinations comfortably and promptly. It is viewed as a combination of on-line scheduling, resource allocation and stochastic optimal control problems similar to robotics or automated manufacturing systems. However, elevator control systems are harder to deal with than other systems because the states of the system are dynamically changed in large state space and coming events are unpredictable in many cases [1]. In offering a solution to elevator control problem, the traditional control systems often yield unsatisfactory results because they lack in considering number of technical characteristics and possibilities to be taken in to account. They also possess limitations and their flexibility is still restricted even if they are adapted to utilize computers. The selection and distribution of the most suitable cars in the building is a function of the call assignment. Landing and car calls are often allocated to suitable cars by taking into account of

the minimum cost concept that operates by performing a trial allocation to all available cars and allocating the call to the car giving the lowest cost. The criteria for determining a suitable cost function depends on either quantity of service and/or quality of service. The quality of service is a measure of the elevator capacity consumed to serve a specific set of calls, indicated by total journey times of all the cars [2].

Elevator control problem has been studied for a long time: in the last two decades artificial intelligence techniques such as fuzzy logic, neural networks, genetic algorithms and evolutionary algorithms were introduced. Although considerable research studied has been carried out to explore artificial intelligence applications in elevator control area [3] to [9], a little work has been done in neural network application in elevator group control.

Neural networks have been applied to tackle the selection and distribution of the most suitable cars which is a function of assignment of calls. Artificial neural networks which simulate the way that human neural networks operate and learn, and mimic the hardware structure of the brain in a simplified way consist of artificial neurons that perform a simple task and are fully interconnected [1]. A neural network is used to predict the number

\*Corr. Author's Address: Istanbul Technical University, Faculty of Mechanical Engineering, Güneşsuyu, TR-34437, Istanbul, Turkey, imrak@itu.edu.tr

of landing calls likely to be made and the cars' likely destination for the time of service in order to optimize the car allocation efficiency. The prediction is made by combining predictions from historic data learnt during corresponding periods of previous days and real time data learnt during a short interval. They can be envisaged as a black box acting in response to a set of inputs of position, direction and calls produces an output representing the next stopping floor. Artificial neural networks can be placed in the conventional control system to improve the elevator performance [11].

In this study, the artificial neural networks has been inserted into an elevator group control systems for minimizing the average passenger waiting time and developed to model the behavior of the building population and also to adapt the algorithm to changes in passenger demand. For training the neural networks, the backpropagation method was used in the present work. The aim of this application is to optimize some of the parameters that a controller uses in order to minimize the average passenger waiting time. The data used here are from a simulator program developed by the author [9]. The elevator traffic analysis has been carried out using the software written in Turbo Pascal version 7.0 by examining the simulation results obtained. Simulations are run for several levels of interfloor traffic demand for the improved Duplex/Triplex control system (ICS). In the simulation an existing building having 3 elevators with 10 floors was selected as a model case. The normalized performance figures, with respect to the interfloor demand, are compared to other traffic control algorithms which are priority timed fixed sectoring algorithm (FSPTS), the fixed sectoring common sector system (FSCSS).

## 1 STRUCTURE OF THE ELEVATOR GROUP CONTROL

Building traffic becomes more and more diversified and complicated as building has more various functions and intelligence. Elevator group control algorithm, therefore, become more complicated as they become more intelligent. The goal of elevator control is to provide operational management of a group of elevators, by selecting cars to meet landing calls and achieve passengers' destinations pleasantly and promptly. An efficient elevator control system has four properties [2]: to

provide even service to every floor in a building, to minimize the passenger journey time in the car, to minimize the passenger waiting time, and to serve as many passenger as possible given time. The basic function of elevator control system is to assign an appropriate car to occurred landing calls to quickly respond them. The car to be assigned will be selected with consideration of many factors such as passenger waiting time, passenger journey time and energy consumption. Among them passenger waiting time is the dominant factor in elevator group control [8].

In this study, the Duplex/Triplex group control system is selected as an example of the conventional elevator group control system. The chosen system is suitable for groups of two or three cars in low-rise buildings and serves car calls and landing calls according to the directional distributive control principles by considering the current position, distance between landing call floor and car position, and the direction of car travel [2] and [9]. In order to perform directional distributive control, the nearest car continuously is to be searched with the call is allocated. Therefore it provides a single control algorithm designed to serve interfloor traffic in low-rise buildings.

Initial design of the elevator control system in building the main concern is to fulfill its requirements in conditions of predominantly up-peak traffic. Under up-peak traffic conditions, the Duplex/Triplex group control algorithm presents poor performance. To investigate improvement of this control algorithm, artificial neural networks were introduced into the algorithm and optimization was carried out to many parameters such as passenger waiting time and performance figure [10] and [11].

Landing calls, car calls and car positions are the inputs of the Duplex/Triplex control system. The elevator configuration such as number of floors, arrival rates, car capacities etc. is the parameters of the control algorithm. The outputs of the group control are the car direction and the next stopping floors. In this study, the next stopping floor problem in elevator traffic control is the explicit problems which has been tackled. It is stated that a given car system, with a known position, direction commitment and registered up and down landing calls as inputs find the next stopping floors as an output of the network. To solve this problem artificial neural network is developed



and applied to the Duplex/Triplex elevator group control algorithm.

## 2 NEURAL NETWORK APPLICATIONS ON THE NEXT STOPPING FLOOR PROBLEM

The quality of service in multi-stories building is indicated by the average passenger waiting time. Thus, the aim of this study is reduce the average waiting time for the different traffic pattern. For this reason, it is proposed that neural networks learn the likely destinations from one floor to another and then use this information to allocate cars to calls and have been employed to predict elevator demand on a day-to-day basis. This study is an exploration in possibilities of the various fields in which neural networks can be applied in elevator control. Feedforward neural networks embedded in the car control and call distribution module is shown in Figure 1.

Neural network embedded elevator control module is situated in the simulation program and allocates the both landing and car calls to suitable car according to the control algorithm. Inverse-Stop Passenger method was proposed by Al-Sharif [12]. This module can continuously learn passenger arrival rate patterns throughout the day of the elevator system, and predict the passenger arrival rates for each floor and destination in the building. In an elevator system, when a landing call is registered at a given floor, the information the group control system receives is the floor identification and desired direction of travel. The next stopping floor as the destination is not known. The information could be used by the elevator control algorithm to send the best car to answer the landing

call. Conservative approaches can not provide such flexibility and autonomous behavior. As a preferred embodiment of this technique, population behavior is modeled using neural network approach as described by Rumelhart and McClelland [13]. Artificial neural network application in identification of vertical traffic pattern uses a feedforward neural network to perform this task [14]. Another neural network application in elevator control intends to be a new method using artificial neural networks to predict the response time of an elevator to a landing call.

### 2.1 The Structure of Artificial Neural Networks

Artificial neural networks consists several units that carry out a simple task and are fully interconnected. This layout allows reproducing some features of the brain that are not usual on computers [15]. A typical simulation of the action of a single neuron is shown in Figure 2. The strength of each synaptic junction is represented by a weight, with a positive sign for excitatory connections, and negative otherwise. The output of a neuron is easily computed by using vector multiplication of the input and weights, and adding the trainable bias to each neuron for speeding up convergence. A bias is included in the neurons to allow the activation functions to be offset from zero. This feature can be easily incorporated by adding an additional weight connected to +1, to each neuron. The additional bias term also determines the spontaneous activity of the neuron. This can also be achieved as setting the threshold values for the sudden onset of a high firing rate, hence the term non-linear threshold element.

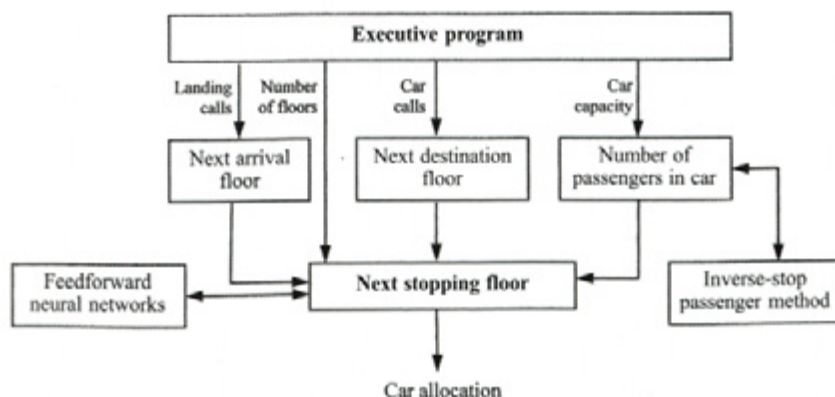


Fig. 1. Overall diagram of an elevator control system

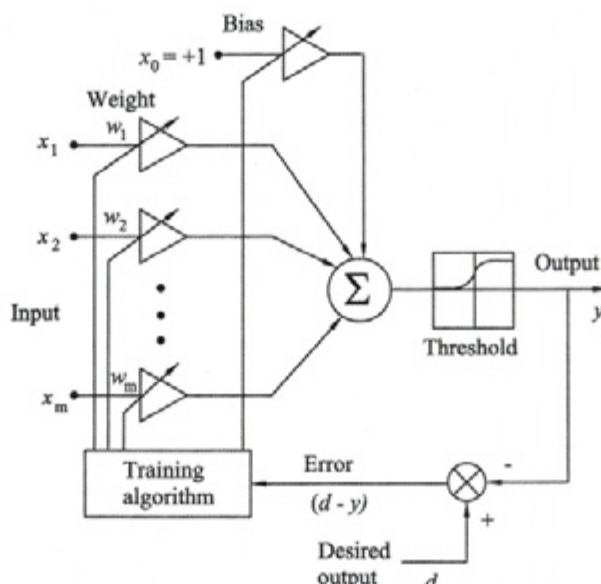


Fig. 2. A typical model of a neuron

After an input has been applied as a stimulus, it is propagated through each unit on output is generated. This output is then compared the desired output, and an error signal is computed for each output. The net input vector  $a_j$  to neuron  $j$  is a non-linear function of the inputs  $x_i$  of the neurons that are connected to  $j$  and the weights  $w_{ij}$  on these conditions is:

$$a_j = \sum_{i=1}^m y_i w_{ij} \quad i \neq j \quad (1)$$

Using sigmoid non-linearity at each node, the neuron has a real-valued output  $y_j$  that is a nonlinear function of the net input is:

$$y_j = \frac{1}{1 + e^{-\alpha(a_j - \theta_j)}} \quad (2)$$

where  $\theta_j$  it the threshold value for the artificial neurons. Improving the performance sigmoid temperature  $\alpha$  is used and slightly higher than one which makes the middle portion of the curve comparatively flatter. The node characteristics of an artificial neuron are thus determined by Equation (2).

The presented system can be adapted to do this purpose because of all the input values such as number of floors are defined as parametrically. A typical multilayer neural network used consists of input and output layers are configured by number of floors and two hidden layers as depicted in Figure 3. An increase the number of hidden layers

may improve generalization capacity. In the present case, after a few trials with the network with one hidden layer, good convergence could not be achieved. The network used, therefore, has two hidden layers as this guarantees it can learn any pattern. There are also no formulae to determine the number of neurons to arrange in hidden layers. The most used procedure is to try several configurations. To improve the generalization, the number of nodes in hidden layers are double that of the number of input layer nodes [16] and [17]. The development of successful neural networks requires considerable time and effort.

In fully connected neural networks, the inputs of the network gets in are: number of car calls, number of up landing calls, number of down landing calls and car directions. The hidden layers are computational layers and fully connected to input layer. There are no direct connections between the input and output layers. Case information flows forward from the input layer to the output layer through the hidden layers. Each layer consists of processing element called neurons [18].

The program uses the multilayer backpropagation method for determining the next stopping floors and calculates the outputs by multiplying the input matrix by the weight matrix to produce the output vectors. Then it sets the highest output to "1", while setting all other outputs to "0". It compares this output with the desired output, and then adjusts the weights.



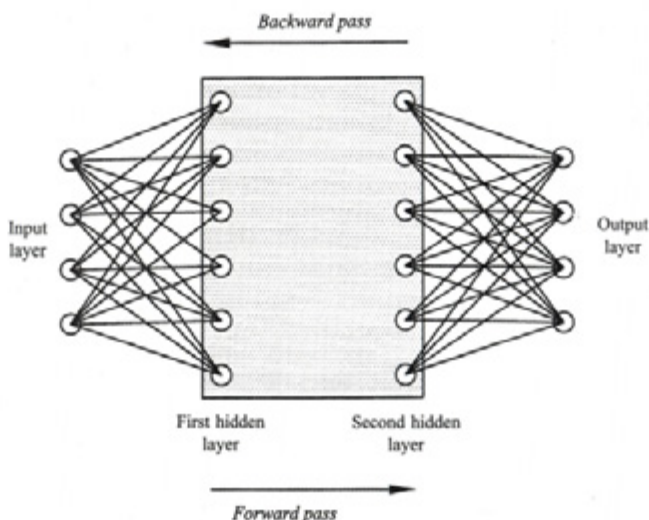


Fig. 3. The architecture of multilayer neural network

## 2.2 The Backpropagation Algorithm

Feedforward neural networks in this study use the backpropagation algorithm, as it is widely used and very popular. It provides predictive information and a systematic method for training multilayer neural networks that has been applied to the control of elevator call allocation [1] and [19]. It does not have feedback connections, but errors are backpropagated during training. Thus, data are processed from the input to output layer, whereas knowledge is performed by the algorithm of minimizing the square error by backward movement [20] and [21].

Backpropagation neural network is an example of supervised learning and applicable for the multilayer neural network for it considers weight parameter in all layers. There are two layers of neural cells between input layer and output layer as shown in Figure 3. The Sigmoid non-linear nodal function is used when the backpropagation learning algorithm is used to train the feedforward network which is intended for the next stopping floor prediction.

The backpropagation algorithm is also based on the generalized delta rule which calculates the gradient to find the deepest descent direction, searching for a minimum in the error surface. It calculates first the error in the output layer, then evaluating the contribution of each hidden neuron in that error. Thus, the corresponding weights are modified in order to reduce the mean squared error between the actual network output and the desired

one. The learning algorithm is an iterative algorithm, intending to reduce the error at the output. The sequence of steps involved is as follows.

Step 1: Initialize all of the weights to small random values. This breaks the network symmetry and ensures that the network does not become saturated by large values of weights.

Step 2: Apply an input vector to the input layer and specify the desired network outputs. The inputs can be new on each trial or samples from a training set can be presented cyclically until weights stabilize.

Step 3: Calculate the network outputs by performing a forward through the layers, using Eqs. (1) and (2) to calculate the outputs of individual neurons. The outputs of the input layer neurons are made equal to the corresponding component of the input vector.

Step 4: Adjust the weights of the network in a way that minimizes the error using recursive algorithm at the output nodes by:

$$w_{ij}(t+1) = w_{ij}(t) + g y_j E_j + e \Delta w_{ij}(t+1) \quad (3),$$

where  $w_{ij}$  is the weight from  $i$ -th node to  $j$ -th node,  $g$  is a learning rate which is normally sets less than 1 to prevent to oscillation of weight values,  $e$  is a momentum coefficient and is normally set to around 0.9 and where:

$$\Delta w_{ij}(t+1) = w_{ij}(t+1) - w_{ij}(t) \quad (4),$$

and  $E$  is the error at  $j$ -th node:

$$E_j = -y_j(1 - y_j)(y_j - d_j) \quad (5),$$

for an output layer, and:

$$E_j = -y_j(1 - y_j) \sum_k E_k w_{jk} \quad (6),$$

for an intermediate layer neuron.

Step 5: Repeat by going to Step 2.

The major disadvantages of this algorithm are its slow convergence, the possibility of being stuck at a local minimum instead of a global and the memorization instead of generalization. Various methods have been suggested to speed up convergence time. Two of these involve the inclusion of a momentum term and a learning rate term in the weight adjustment equations. Using this method, the network tends to follow the bottom of narrow gullies in the error surface rather than oscillating from side to side.

There are also problems if the number of training pattern is small because this patterns using a big network in order to avoid the over fitting of the training data. Some modifications to the original algorithm tend to overcome these difficulties. Increasing the number of hidden layers can also improve the generalized capacity.

### 2.3 Modification Parameters

Modification parameters can be made to the backpropagation algorithm which may improve its

performance and scale the adjustments to weights. They involve change stop the weight update procedure, a variation to improve computational power and an explicit statement of the backpropagation algorithm for two hidden layers. A simple change to training law that sometimes results in much faster training is the addition of a momentum term. It is an attempt to try to keep the weight change in process and maintains the same direction the weight variation, ensuring stability in the learning. For constant learning rate, the number of epochs with respect to momentum can be seen in Figure 4. A momentum term is added to speed up the learning stage and takes value of 0.9 in this study.

Learning rate as another modification parameter can be used to control the amount which weights are changed during the training and be used as one of convergence. This convergence is drawn according to epoch with constant momentum parameter as shown in Figure 4. Therefore, the learning rate is set at 0.5 for training the network. As can be seen from Figure 4, care has to be taken in choosing the learning rate and the momentum, in order to ensure convergence. Higher learning rate values make convergence faster, but learning rate values higher than 0.5 must be avoided, as well as momentum values higher than 0.9.

### 2.4 Training the Network

The process of learning is known as training. There are three phases of training. In first phase,

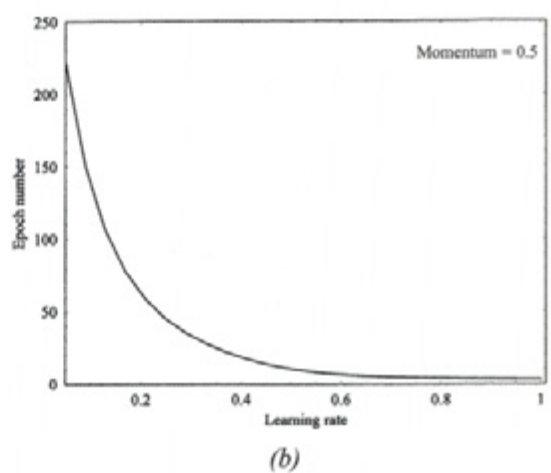
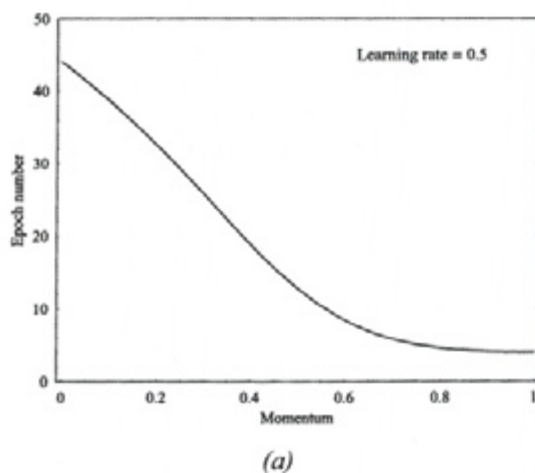


Fig. 4. Modification parameters: (a) Momentum with constant learning rate (b) Learning rate with constant momentum



Table 1. Parameters of neural networks training

Number of floors	$n$		Neural model	backpropagation
Number of layers	4		Topology	Feedforward
Number of neurons	Input layer	$3n + 3$	Supervision	Supervised
	1 <sup>st</sup> Hidden	$2(3n + 3)$	Learning rule	Generalized delta rule
	2 <sup>nd</sup> Hidden	$2(3n + 3)$	I/O information	Binary
Transfer function	Output layer	$n$	Momentum	0.9
	Input layer	Linear	Learning rate	0.5
	1 <sup>st</sup> Hidden	Sigmoid	Sigmoid temperature	0.5
	2 <sup>nd</sup> Hidden	Sigmoid		
	Output layer	Linear		

supervised learning is used to train the network to predict the next stopping floor. The training set is first presented to the network, until it learns the internal representation of training pairs and then the error at the output node is reduced along the steepest descent direction. The initial weights and the thresholds are randomly generated at the beginning. At the end of the first phase of training, weights from the input layer to the hidden layer of the network are fixed. In phase two, the output layer of the network is retrained to emulate the existing controller. In phase three, single weights in the output layer of the network are perturbed, and the resulting performance is measured. The weights are then modified in the direction of improved performance. Neural networks to be trained wherein the parameters specified in Table 1 were employed.

Because of the size of the training pattern, it is kept a separate file produced by the training pattern program. A training pattern consists of training pairs of inputs and outputs. The inputs are

car calls, up and landing calls and car direction. All the inputs were introduced for several epochs. The outputs are the next stopping floors. A schematic diagram of an elevator system of a group of cars is illustrated in Figure 5. A group of cars has to handle a group of passengers in a building with a given number of floors. In this figure, the elevator cars are represented as boxes in the diagram. In addition, a black circle indicates a car call, a black triangle indicates a landing call assigned to an elevator, and a white triangle indicates an unassigned landing calls.

When considering even small number of stops, the size of the training pattern becomes extremely large to be calculated manually. Thus it decodes the direction and the position linearly, and the calls in binary [12]. The module is used to classify the input vectors and to convert them in binary information as shown in Figure 5, by a processing module active code/binary code. Thus, the neural network receives only binary data that represent a favorable situation for applications in

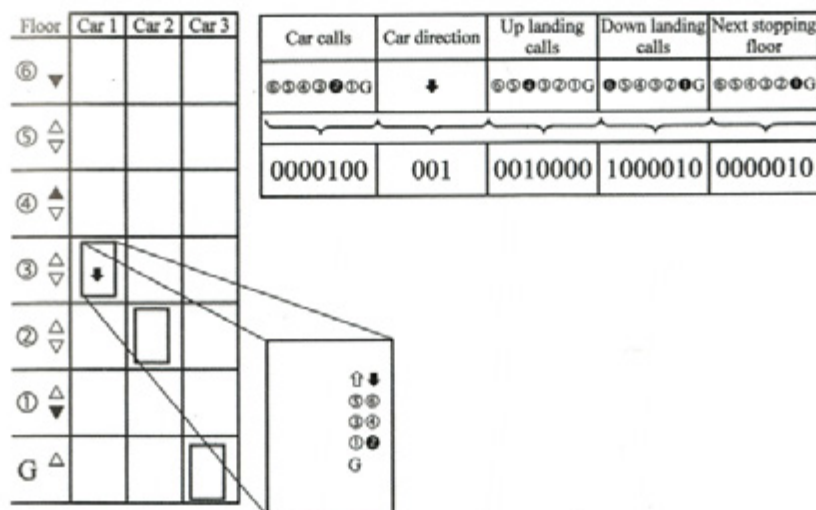


Fig. 5. Elevator system schematic diagram

large systems. This represents a reduction on the computational effort necessary to realize the training and an improvement on the quality of the analysis. This form permits to represent the contingences by a binary code and reduce the quantity that is an adequate way of providing a faster training and the analysis more reliable [9] and [12].

In order to train the neural network to be able to calculate the next stopping floors, the set of possible patterns has to be evaluated in advance, and stored in a file to be used later by the training program. While calculating the situations, it finds the next stopping floor corresponding to each simulation, and then writes all these variables to a file. In the sample file record given in Fig. 5, "0" represents no call and "1" represents a call to be registered. The length of record lines depends on the number of floors. The first 7 digits represent the car calls, the next 3 represents the direction (001 for down and 100 for up), the next 7 represents the up landing calls (each digit represents one call), and the next 7 represents the down landing calls and the last 7 digits represents the calculated next stopping floors. This format is used to give flexibility in order to changing the number of floors during input stage.

Each training pattern was randomly selected to be fed into the network, continuing until all the patterns were done with. This constitutes one epoch of learning. The network was trained for 1000 epochs, until the mean squared error dipped below 0.01 then the program stops [9]. After 1000 iterations the minimum reached which corresponds to the model that minimizes the fit for the validation data. Further iterations give an increasing criterion and the model becomes worse. Once the training stage is successfully finished and the trained network is tested, the network is employed in the prediction of the next stopping floors by considering what has been learnt in normal operation mode.

## 2.5 Overtraining the Network

In neural networks, one of the major pitfalls is overtraining, analogous to curve fitting for rule based trading systems. Overtraining refers to the reduction in generalization ability that can occur as networks are trained, and may be avoided by early stopping the training stage in time, before an

adaptation to the noise starts. It is difficult to detect an optional stopping moment without the use of a test set, pruning, reducing the size of a network after it is trained [12], [15] and [17].

Overtraining occurs when a network has learned not only the basic mapping associated with input and output data, but also the subtle nuances and even the error specific to the training set. Overtraining can be detected during the training stage by the use of a test set. Once the data set has been selected, it is separated into training and testing subsets. In the testing mode, the network is fed new inputs and utilizes the representation it had previously learned to generate associated outputs without changing its weights.

Overtraining is reduced by limiting the number of neuron to minimum necessary, increasing the training set size, using cross-validation training to identify and stop training before over fitting occurs. The cross-validation set was used to monitor generalization during training.

## 3 THE ELEVATOR CONTROL SIMULATION

In this section the elevator control simulation example is considered. A widely used method for elevator system performance prediction is the use of simulation. This can usually produce a set of curves, which describe the behavior of a certain variable against the change in system loading. One of the most useful curves which are used to study elevator system performance is the average waiting time curve versus interfloor demand. Interfloor demand as percentage can be found by dividing the passenger arrival rate by the handling capacity of the system [2]. Simulations are run for several levels of interfloor demand for the improved control system. At the end of each simulation, the average waiting time is recorded and the performance figure calculated by dividing the average waiting time by the interval to normalize it.

### 3.1 The Structure of Software

The simulation program is executed on a pseudo building with various numbers of elevator cars. The simulator was written by the author. A modular construction has been used as shown in Fig.6 as this allows ease of programming, extension, modification and defined interfaces



between modules and databases. Conversational programming techniques have been used to take advantage of an interactive digital computer. All inputs are checked for validity and where an error is detected the user is prompted as to the correct reply. The principal features of the simulator are that it can simulate up to 3 cars and 10 floors.

The lift configuration module is an interactive conversational program which allows the setting up of all the data defining the system to be simulated. Three cars with various contract capacities are used to service the whole building. In this program interfloor distance, contract speed, time for opening and closing the door, passenger transfer time are chosen by users. After that the training module is executed to be able to calculate the next stopping floors, the set of possible patterns has to be evaluated in advance, and stored in a file to be used in neural networks module. In this program, feedforward neural network is used to allocate the calls to the best cars during day period by means of weight matrix. The software uses the backpropagation learning algorithm to training the network.

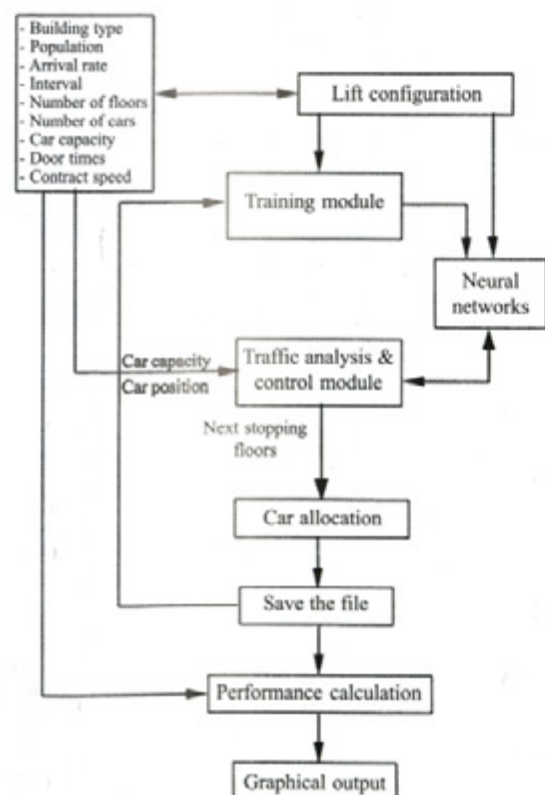


Fig. 6. The flow chart of simulation program

The traffic analysis and control module which is embedded in the simulation program assigns the landing calls to the convenient cars. This module decides which car to be dispatched to which floor. In the traffic analysis and control module, the executive is the simplest part in the module and is used to control which sub-module is to run. After each sub-module has been used, the executive is re-entered to determine the next phase in a simulation. The executive also controls the access to the traffic analysis and control module. It generates landing and car calls. The car capacity is selected by users. The goal of the module is to obtain the next stopping floor, which is important in the allocation of the cars. In the first step, the landing calls and car calls, next arrival floors and next destination floors are determined. The number of passengers in the car, which one of the factors in selecting the next arrival floor is determined by means of Inverse-Stop Passenger method [12]. After two steps the next stopping floors can be determined using feedforward neural networks.

The output of simulation is the distribution of calls to the most suitable cars and car direction. The results are kept a separate file and saved in a file to be used later by the training program. Performance calculations have been arranged to present graphs to the user. They include round trip time, average waiting time and the normalized system performance figure which are used for a performance comparison of elevator control systems.

The flow chart of simulation program which employs both conventional Duplex/Triplex control and the neural network algorithm is depicted in Figure 6 [9].

### 3.2 Performance Criteria and Calculations

The performance objectives of an elevator system can be defined in many ways. One possible objective is to minimize the average waiting time, which is the time between a passenger's arrival and entry into a car. The passenger waiting time would be the best indicator of the quality of service that an installed elevator system could provide. To calculate the average waiting time according to car loads, one can find:

$$AWT = \frac{2Ht_s + (S+1)t_d + 2Pt_p}{L} \left[ 0.4 + \left( \frac{1.8P}{CC} - 0.77^2 \right) \right] \quad (7)$$

where  $P$  is the number of passengers enter the car at the main terminal,  $CC$  is the contract or rated capacity of car,  $H$  is the highest reversal floor,  $S$  is the expected number of stops,  $t_v$  is the single floor transit time,  $t_s$  is stopping time and  $t_p$  passenger transfer time and  $L$  is the number of cars within a elevator group. The highest reversal floor and the expected number of stops are determined by means of simulation program. The number of passengers in the car is calculated by means of Inverse-Stop Passenger method.

The normalized system performance figure is defined as the percentage of the simulated average passenger waiting time divided by calculated up-peak interval. The interfloor demand is defined as the interfloor arrival rate divided by calculated up peak handling capacity.

#### 4 SIMULATION RESULTS

To investigate the improvement obtained by using the group control where neural network are embedded into the Duplex/Triplex control algorithm, simulations are run for several levels of interfloor traffic demand level for the improved Duplex/Triplex control system (ICS). The simulation program is executed for a building which population 1000 with 10 floors, three cars with contract capacity equal to each is used to service the whole building. Interfloor distance is fixed at 3.3 m, contract speed 1.0 m/s. Door operating time is 20 seconds while closing time 2.6 secs and passenger transfer time is 1.2 secs. Simulations output file keeps the record of the round trip time, the interval and the average passenger waiting time. The program runs on a 486PC with 8 MB RAM, 256MB hard disk configuration. The normalized performance

figures, with respect to the interfloor demand are compared with other traffic control algorithms which are the fixed sectoring priority timed system (FSPTS) and the fixed sectoring common sector system (FSCSS) control algorithm. For a performance comparison, the results of each simulation performed under the same conditions have been tabled. Table 2 gives the comparison of the performance figure of the traffic control algorithms in a normalized form. Existing control algorithms as standards are used for comparison [9] and [10].

The simulation results showed the proposed method could provide a considerable improvement of the call assignment, if the method is implemented to an elevator control system for low-rise buildings in up peak traffic conditions. The ICS shows an improvement over the fixed sectoring priority timed system (FSPTS) and the fixed sectoring common sector system (FSCSS) control algorithm, when the interfloor demand exceeds 55%.

#### 5 EXPANSIONS IN MATERIALS HANDLING SYSTEMS

Automated Guided Vehicle (AGV), one of the materials handling systems, is widely used in flexible manufacturing systems. Due to the increasing complexity of manufacturing environments coupled with the need for increased flexibility in AGV systems. The design and evaluation of AGV systems are highly complex because of the randomness and the large number of variables involved. Typically the design of material handling systems using AGVs includes determination of the vehicle guideway layout, the traffic flow pattern, the number of vehicles required [22].

Table 2. Comparison of the system performance

Interfloor demand [%]	Performance figure [%]		
	FSPTS	FSCSS	ICS
10	25.7	28.5	42.5
20	31.4	37.1	51.3
30	45.6	51.3	64.1
40	64.6	66.9	78.8
50	86.6	82.8	93.1
60	114.0	106.8	112.4
70	143.9	133.4	134.1
80	174.9	163.8	151.6
90	209.5	195.2	168.2
100	238.8	226.6	190.8



The problem of AGV consists of the operational control strategies of dispatching and routing of a set of AGVs and can be solved by using a lot of heuristic algorithms as Petri nets, fuzzy logic and neural networks. Hao and Lai [23] have described a neural network approach for the AGV problem and proposed neural network models to perform dispatching and routing tasks for the AGV under conditions of single or multiple vehicles which are based on Kohonen's self-organizing feature maps. Bostel *et.al.* have proposed a new navigation technique for AGV's based on neural networks [24].

The desired path along which the vehicle should move may be a fixed route, semi fixed route or an arbitrary route. Fixed route implies the installation of a fixed active guide like an energized cable or a passive one like a reflecting strip painted on the plant floor. It is essentially AGV programmed to convey materials or work in progress through a predefined route in horizontal environment. It is similar with elevator which carries the passengers and loads through a predefined route in vertical environment.

Vehicle travel time of AGV is a fundamental parameter for solving various flexible manufacturing system design problems. The feedforward neural networks embedded system presented in this paper possesses generality for transportation systems. It is mostly used in elevator engineering, but it could be used for single vehicle or multi vehicle AGV's control up to 3 vehicles in flexible manufacturing systems, if it is adapted and generalized for fixed route path problem using average waiting time and average travel time concepts as described in elevator traffic calculation. This control system could be applied the static production environment where the product mix or machine routings are assumed to be stable over time. It can also be used to determine and estimate the vehicle waiting times of AGV. It is believed that the models can be applied to a number of call-a-ride cases, such as AGV travel.

## 6 CONCLUSIONS

In this paper the outline of a program preparing feedforward neural networks has been introduced. It has been seen that the average waiting time which defines elevator performance goes down satisfactorily thanks to the neural network applied to elevator control systems and that it gives

better results compared to conventional methods. It could effectively support the conventional group control system and shorten the average waiting time by forecasting car position and using call distribution laws. The system results are shown in Table 2 compared to those obtained from conventional control algorithms. A 15 to 20% improvement in performance has been achieved by using this technique. It can be concluded that better results are reached when neural networks are combined with conventional techniques. On the other hand, the main drawback of the method is that it mainly applies to up-peak traffic. More research is also needed for down-peak, as it depends on building occupancy. This control system may be expanded to the other materials handling systems such as AGV's control problem with a few necessary arrangements.

## Acknowledgement

The constructive comments of the referees are gratefully acknowledged.

## 7 REFERENCES

- [1] Miravete, A. *New materials and new technologies applied to elevators*. New York: Elevator World Inc, 2002.
- [2] Barney, G.C. *Elevator traffic handbook*. London: Spon Press, 2003.
- [3] Ho, M., Robertson, B. Elevator group supervisory control using fuzzy logic. *Canadian Conference on Elevator and Computer Engineering* 2, 11.4.4., 1994.
- [4] Prowse, R.W., Thomson, T., Howells, D. *Design and control of lift systems using expert systems and traffic sensing*. Elevator technology 4 (Ed. G.C.Barney). London: IAEE Publ, 1992, p. 219-226.
- [5] Qun, Z., Ding, S., Yu, C., Xiaofeng, L. Elevator group control system modeling based on object oriented Petri net. *Elevator World*, 49(2001)8, p. 99 - 105.
- [6] Chan, W.L., So, A.T.P. Dynamic zoning for intelligent supervisory control. *Int. to Elevator Engineering*, 1(1996), p. 1 - 10.
- [7] Siikonen, M-L. *On traffic planning methodology*. Elevator technology 10 (Ed. A. Lusting). London: IAEE Publ, 2000, p. 267-274.

- [8] Closs, G.D. *The computer control of passenger traffic in large lift systems*. PhD Thesis, University of Manchester, Institute of Science and Technology, Manchester, UK, 1970.
- [9] Imrak, C.E. *Traffic analysis, design and simulation of elevator systems*. PhD Thesis, Istanbul Technical University, Istanbul, Turkey, 1996.
- [10] Imrak, C.E., Barney, G.C. *Application of neural networks on traffic control*. Elevator technology 9 (Ed. G.C.Barney), IAEE Publ, London, 1998, p. 140-148.
- [11] Imrak, C.E., Ozkirim, M. The modeling and simulation of elevator group control systems for public service buildings. *Preprints the 3<sup>rd</sup> IFAC Workshop DECOM-TT 2003*, Istanbul, Turkey, June 2003, p. 159-164.
- [12] Al-Sharif, L.R. *Predictive methods in lift traffic analysis*. PhD Thesis, University of Manchester Institute of Science and Technology, Manchester, UK, 1992.
- [13] Rumelhart, D.E., McClelland, J.L. *Parallel distributed processing*. Vol. 1 and 2, Cambridge: MIT Pres, 1986.
- [14] So, A.T.P., Beebe, J.R., Chan, W.L., Liu, S.K. *Elevator traffic pattern recognition by artificial neural network*. Elevator technology 6 (Ed. G.C.Barney). London: IAEE Publ, 1995, p. 122-131.
- [15] Mukherjee, A., Deshpande, J.M. Application of artificial neural networks in structural design expert systems. *Computer & Structures*, 54(1995)3, p. 367-375.
- [16] Korn, A.G. *Neural networks and fuzzy-logic control on personal computers and workstations*. London: MIT Press, 1995.
- [17] Lisboa, R.G. *Neural network current application*. New York: Chapman and Hall Pub, 1992.
- [18] Alexandris, N., Chrissikopoulos, V., Vassilacopoulos, G. *Lift - an expert system for lift system design*. Elevator technology 2 (Ed. G.C.Barney). London: IAEE Publ, 1988, p. 1-9.
- [19] Rumelhart, D.E., Hinton, G.E., Williams, R.J. Learning representations by backpropagation errors. *Nature*, 323 (1986), p. 533-536.
- [20] Jovanovic, J., Krivokapic, Z. The application of an atypical neural networks when quantifying the modeling of environmental aspects. *Strojnski vestnik - Journal of Mechanical Engineering*, 52(2006)6, p. 392-403.
- [21] Krivokapic, Z., Zogovic, V., Spaic, O. Using neural networks to follow the wear of a S390 twist drill. *Strojnski vestnik - Journal of Mechanical Engineering*, 52(2006)7-8, p. 437-442.
- [22] Koo, P-H., Jang, J. Vehicle travel time models for AGV systems under various dispatching rules. *The International Journal of Flexible Manufacturing Systems*, 14(2002), p. 249-261.
- [23] Hao, G., Lai, K.K. Solving the AGV problem via a self-organizing neural network. *The Journal of the Operational Research Society*, 47(1996), p. 1477-1493.
- [24] Bostel, A.J., Gan, W.W., Sagar, V.K., See, C.H. Generation of optimal routes in a neural network based AGV controller. *Second International Conference on 'Intelligent Systems Engineering'*, Hamburg, Germany, September 1994, p. 165 -170.