Second International Conference Buildings
and the Environment, June 1997, Paris

A PROPOSITION OF INTERIOR AIR FLOW ASSESSMENT METHOD FOR HUMID TROPICAL ARCHITECTURE

J. I. KINDANGEN*° and G. KRAUSS*
***CETHIL/ Thermique du Bâtiment, INSA de Lyon, 20, av. A. Einstein, Bât. 307, 69621 Villeurbanne Cedex, FRANCE and °Sam Ratulangi University, Faculty of Engineering, Architectural Department, Manado 95115, INDONESIA.**

ABSTRACT

Increasing demands for energy saving and a higher degree of comfort in rooms compels designers or architects to use more sophisticated analysis methods. The measurement *in situ*, numerical simulation (CFD), and wind tunnel investigations are three of methods which are always utilised to analyse or to assess air flow in rooms and their environment. However, these methods remain generally very difficult for the majority of the designers or the architects. With the help of these two latter methods we attempt to establish a database concerning the influence of architectural design elements on interior air flow; with which we afford the training phase of our artificial neural nets.

This paper presents an assessment method of interior air flow using artificial neural nets. The air flow distribution inside a building depends not only on the external wind velocity, but also largely on the building design overall especially for humid tropical architecture. Due the difficulty to evaluate the interior air flow if we have to take into account a number of architectural design elements, we proposed to use this approach. The utilisation of the neural networks as a universal predictor is an interesting field of investigation. They provide reliable results in the cases where many parameters have to be taken into account simultaneously.

1. INTRODUCTION

The main factors which affect physical comfort are: the temperature, humidity and air movement, the latter being the only one which can be controlled to a significant extent without substantial energy expenditure. Air movement behaviour can be analysed by some methods such as measurement in real site [1], experimentation in wind tunnel [2-4] and numerical simulation such as Computational Fluid Dynamics [5-7].

In humid tropical climate, when a sufficient air flow exists to permit a good ventilation we can obtain the comfort in buildings without using any systems of active air-conditioning. The air motion acts on the thermal comfort, decreasing the risks of local overheating and permitting sufficient interior air velocity to improve the evaporation of sweat.

The main reason of utilising Artificial Neural Networks (ANN) is that due the difficulty to evaluate interior velocity coefficient if we take into account a number of architectural elements. Neural networks can estimate input-output functions, unlike statistical estimators, they estimate a function without a mathematical model of how outputs depend on inputs. They are *model-free* estimators and *learn from experience* with numerical sample data.

We can program or train neural networks to store, recognise, and associatively retrieve patterns or database entries; to solve combinatorial optimisation problems; to filter noise from measurement data; to control ill-defined problems; in summary to estimate sampled function when we do not know the form of the functions [8].

2. METHODOLOGY

2. 1. Parameter Used

In this setting, a non-dimensional parameter named coefficient of velocity is used. It is defined as the ratio of the mean air velocity in the zone studied to the air velocity in a point of reference in the exterior flow windward of the building.

$$C_v = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{V_i}{V_r}\right)$$

Where: C_v is interior velocity coefficient, V_i is mean velocity at interior location i (ms^{-1}), V_r is mean outdoor reference free-stream velocity at the reference height (ms^{-1}) and n is number of points measured.

Several authors [3,4] proposed thus the correlation in order to evaluate the interior velocity coefficients. Generally the proposed correlation carries on some simple profiles and does take into account one or two influential elements. To take into account multiple parameter (> 3) in the establishment of the correlation is very delicate. This led us to try using neural networks.

2. 2. Artificial Neural Networks

ANN was invented in the spirit of a biological metaphor. The biological metaphor for neural networks is the human brain. Like the brain, this computing model consists of many small units that are interconnected. These units (or nodes) have very simple abilities. Hence, the power of the model derives from the interplay of these units. It depends on the structure of their connections.

In 1943, MacCulloch and Pitts proposed a mathematical pattern of a biological neuron [9]. This type of neuron possesses a very simple dynamic. The input signal is valued like the weighted sum of excitations coming from the outside. From a mathematical point of view, a feed-forward neural network is a function. It takes an input and produces an output. The input and output are represented by real number. A simple neural network may be illustrated like in Figure 1, for a network with one hidden layer.

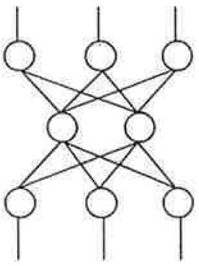

Figure 1. A neural network.

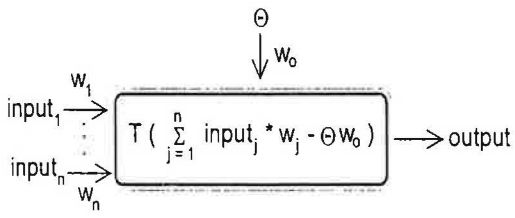

Figure 2. Information processing in a neural network unit.

The constellation of neurons and connections is called the architecture of the network, which is also called the topology. The computational neuron receives *n* inputs (from the environment or from other connecting neurons) and produces one output (to the environment or to other neurons). A neuron may have an additional input, Θ, that acts as a threshold or as an outside environmental input. If the input is from the environment, then such input should be received without modification by the neuron. If the input is from a connecting neuron, then the input may need to be attenuated or strengthened. For this purpose, the input may be multiplied by a weight, *w*, as shown in Figure 2.

These weights can be modified as the neuron is trained so that the neuron produces the desired output. The weights also are characterised as long term memory because changes to them occur in very small increments; therefore, they do not *forget* quickly. The neuron, itself, corresponds to short term memory because it replaces its value every time new input is presented.

To produce its output, the neuron may use a threshold (activation or signal) function to map the input to a specified range. Usually this range is very small ([0,1]) in order to keep neurons from having extremely large values. Using a threshold function is often necessary to cause an entire network of neurons to converge or *settle down* during training and recall.

Typical threshold functions include:

- linear -- $f(x) = x$

- non linear ramp -- $f(x) = \begin{cases} +\alpha & \text{if } x \geq +\alpha \\ x & \text{if } |x| < \alpha \\ -\alpha & \text{if } x \leq -\alpha \end{cases}$

- step -- $f(x) = \begin{cases} +\alpha & \text{if } x > 0 \\ -\alpha & \text{otherwise} \end{cases}$

- sigmoid -- $S(x) = (1 + e^{-x})^{-1}$ or $S(x) = \tanh x$

The process of training the network is straightforward. Initially, the weights in the network are randomised, except those weights from the outside world to the input layer, which are set as +1. The network is then presented with repeated sets of inputs targets together with their correct output targets. The feed-forward operation calculates an output for each input and then compares it with the desired or correct output. The difference between the desired and computed output is the error, and this is then propagated backward through the network, using the gradient descent rule to update the weights on the connections as it goes, so that the same error will not occur again. When the total error of the network reaches a minimum then the network is trained. The weights are the values that are modified by the training algorithm. The training of all patterns of a training set is called an epoch.

3. IMPLEMENTED NEURAL NETWORKS

A neural network constitutes a universal predictor. It can learn a pattern set and output as it would dictate on unknown patterns through generalisation over the pattern set. Thus, it is possible to use it for prediction of interior velocity coefficients (C_v) as a function of several architectural elements.
The constitution of a training set and a generalisation set, the choice of the precise architecture of the network and the choice of adequate parameters of training should be considered at the time of the network design.

There are no reliable guidelines to decide what the number of neurons in a hidden layer should be, nor how many hidden layers to use [10-12]. Stevenson [13] showed that two hidden layers are sufficient to produce better results. Networks with more than two hidden layers are rare, mainly due to the difficulty of training them.

The constitution of the training set consists of getting the input and the output parameters. In our case the inputs are the directions of the wind and some architectural elements of a building and the outputs are the interior velocity coefficients. The larger the space of training set and the wider it covers the inputs spectrum; the better the results of the training.

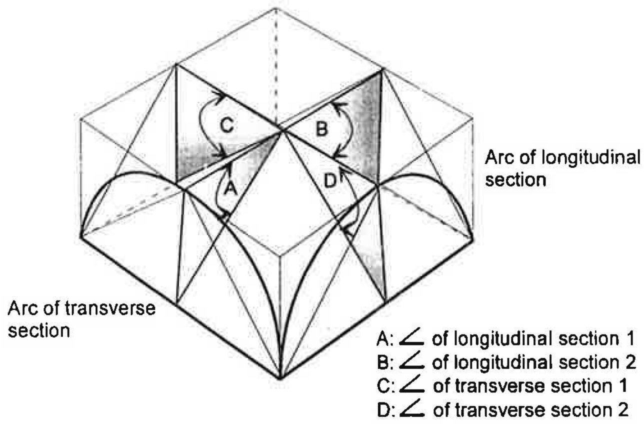

Figure 3. Roof geometry elements.

The input data contains the value associated of the C_v and the architectural elements treated as follows:

- *Direction of wind (building orientation).*
- *Roof shape having 6 elements, as shown in Figure 3.*
- *Whether building is stilted -- pierced floor or not.*
- *Windward side (5 elements):*
  - *Number of openings (maximum for 2 openings)*
  - *Percentage of wall porosity*
  - *Presence of wind wing (cheeks)*
  - *Overhangs on this side*
  - *Openings with wire net or not*
- *Leeward side (5 elements -- the same as the windward side).*
- *Right side (4 elements):*
  - *Number of openings*
  - *Percentage of wall porosity*
  - *Overhangs on this side*
  - *Openings with wire net or not*
- *Left side (4 elements -- the same as the right side).*

Two types of training sets were used. The first used the real values for the inputs-outputs. For the second the whole numeric value was normalised by:

$$x_i = \frac{(x - x_{min})}{(x_{max} - x_{min})}$$

Where: x_i is normalised value, x is original value, x_{min} is minimal original value and x_{max} is maximal original value.

- The normalisation of the training inputs generally improves the quality of the training. The basis of training was constituted in using some obtained results by numerical simulations (*STAR CD* and *FLUENT*) [5-7] and by some results of measurements achieved in a numerical wind tunnel [2].

In order to evaluate the performance of a network, we used the distribution of the inputs which belong to the numerical simulations (CFD) of 30 models, corresponding to 97 cases and by some tests in wind tunnel of 15 models, corresponding to 119 cases. The available data are divided into two parts, one for training and the other for generalisation or validation. The basis of training includes 166 cases and the basis of the generalisation includes 50 cases.

The architecture of the network, we retained the following profiles: network with 4 layers: 1 input layer -- 2 hidden layers -- 1 output layer. The following activation functions are used: logistic sigmoid (logsig), tangent hyperbolic (tansig) and linear functions (purelin).

The number of neurons used for each layer could vary between 5 and 50 neurons. The types of training used are back-propagation, back-propagation with the momentum and Levenberg-Marquardt optimisation. The algorithms of training have been implemented in the software package *MATLAB* [14].

4. RESULTS

A number of networks with different numbers of hidden neurons were run (112 networks with 2 hidden layers), and the network with the minimum validation error (the minimum mean error in generalisation) is used as the optimum configuration. Initially, the error on both the training set and the generalisation set should fall; however, once the generalisation error starts to rise, it is clear that there are unmodelled dynamics in the net that it is not due to modelling. Further training is counterproductive. The best obtained results and the network configurations used are shown in the Table 1.

The best obtained results	
	ANN with 2 hidden layers
Mean error in training	0.46%
Mean error in generalisation	3.11%
No.of neurons, activation function	20, tansig (in the 1st hidden layer)
No.of neurons, activation function	20, tansig (in the 2nd hidden layer)
No.of neurons, activation function	1, logsig (in the output layer)
Number of epochs	40 epochs
Back-propagation method	Levenberg-Marquardt
Error goal	0.01
Type of training set	normalised

Table 1. The best obtained results.

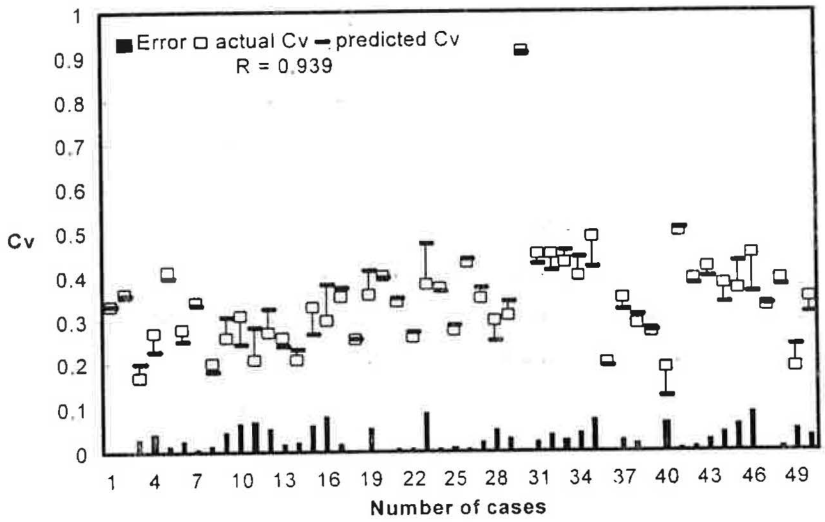

Figure 4. Actual C_v vs. predicted C_v in generalisation for the best obtained results.

The results of the Figure 4 were obtained with the Levenberg-Marquardt back-propagation having 2 hidden layers with 20 neurons and transfer function of tangent hyperbolic function for the first and the second 20 neurons and the same one with the first. This networks provided reliable results, 3.11% of mean error in generalisation.

5. CONCLUSIONS AND PERSPECTIVES

The utilisation of neural networks as a universal predictor is an interesting field of investigation as an interior air flow assessment tool. They provide reliable results in cases where many parameters have to be taken into account simultaneously.

The neural networks allow thus to preserve the descended results under a usable form of some applied measurements or of numerical simulations.

The enrichment of the training set requires only a re-actualisation of the obtained weights by a new phase of training but not a complete redefinition of the network.

A certain number of inconveniences must be mentioned: the knowledge contained in the network is not legible, in the case that we are interested in the constitution of training set is a difficult task which requires some implementation of other means (experimental or numerical) and the rules of network design are very empirical and the clarification of network requires a large number of tests.

6. REFERENCES

1. C. Celestine, *Ventilation Naturelle et Confort Thermique dans l'Habitat en Climat Tropical Humide - Approche Expérimentale en Vraie Grandeur de la Ventilation Naturelle sur un Site de la Guadeloupe - Etude Comparative en Soufflerie*, Thèse Docteur Ingénieur, INSA de Lyon (1985)
2. G. Gouin, *Contribution Aérodynamique à l'Etude de la Ventilation Naturelle de l'Habitat en Climat Tropical Humide*, Thèse de Docteur, Université de Nantes (1984)
3. D. R. Ernest, F. S. Bauman, E. A. Arens, The prediction of indoor air motion for occupant cooling in naturally ventilated buildings, *ASHRAE Trans*, **97**(1) 539-552 (1991)
4. B. Givoni, *L'Homme, L'Architecture et Le Climat*, Editions du Moniteur, Paris (1978)
5. J. I. Kindangen and G. Krauss, Investigation of natural ventilation with Computational Fluid Dynamics. A comparison study with wind tunnel results, *Architectural Science Review*, **39**(2) 113-120 (1996)
6. J. I. Kindangen, G. Krauss and P. Depecker, Effects of roof shapes on wind-induced air motion inside buildings, *Building and Environment*, **32**(1) 1-11 (1997)
7. J. I. Kindangen, G. Krauss, G. Rusaouën, Influence of architectural parameters on air flow distribution inside a naturally ventilated single-unit house, In *Proc. of The 13th International Conference on Passive and Low Energy Architecture, Louvain-la-Neuve, Belgium*, 171-176 (1996)
8. B. Kosko, *Neural Networks and Fuzzy Systems: A Dynamical Systems Approach to Machine Intelligence*, Prentice-Hall International Editions, Englewood Cliffs, NJ (1992)
9. J. F. Jodouin, *Les Réseaux Neurones, Principes et Définitions*, Hermès, Paris (1994)
10. J. F. Jodouin, *Les Réseaux Neuromimétiques*, Hermès, Paris (1994)
11. H. Abdi, *Les Réseaux de Neurones*, PUG, Grenoble (1994)
12. S. A. Mengel, *Using Neural Networks to Predict Student Behavior in Intelligent Tutoring Systems*, Ph.D. Dissertation, Texas A&M University, 1990
13. J. Stevenson, Using artificial neural nets to predict building energy parameters, *ASHRAE Trans.* **100**(2) 1081-1087 (1994)
14. Demuth, M. Beale, *Neural Network Toolbox, for use with MATLAB*, The Math Works Inc. (1994)