Frequency domain: After extracting the estimated line components from the wavenumber spectra (AOAs of direct and specular components) the residual frequency $S_D'(f)$ is estimated for the two datasets $d1$ and $d2$ referred to previously. A number of realisations is involved at different times and the resultant spectra are superimposed on one another. Theoretical considerations use a Gaussian model for the autocorrelation function. Indeed, the results presented in Fig. 1 seem to support this. Each spectrum shown in Fig. 1 is the average, computed from 32, concurrent, 127 sample sequences.

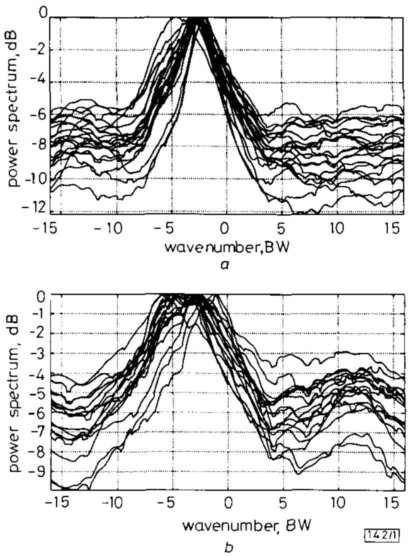

Fig. 1 *Frequency spectra of 20 realisations of dataset d1, and 17 realisations of dataset d2*

Original signal was about 40 dB above noise floor
a 20 realisations of dataset $d1$
b 17 realisations of dataset $d2$

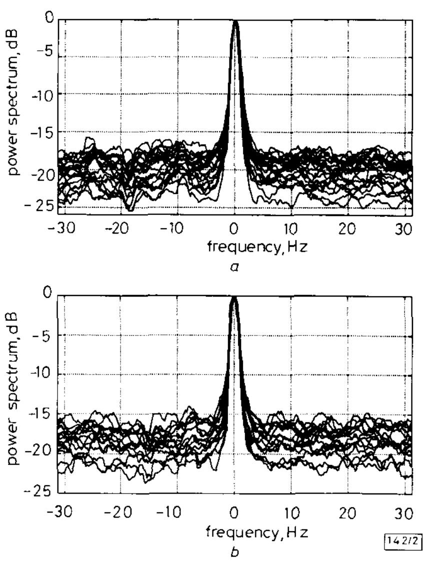

Fig. 2 *Wavenumber spectra of 20 realisations of dataset d1, and 17 realisations of dataset d2*

Projected wavenumber axis indicates AOAs of incoming waves
$1BW$ equivalent to physical aperture of 0·921° so that total, unambiguous, field of view is from −14·91° to 14·91°
Physical horizon is very close to $0BW$
a 20 realisations of dataset $d1$
b 17 realisations of dataset $d2$

Wavenumber domain: The wavenumber spectrum $S_D^s(\phi)$ of the residual data is next estimated, and the results obtained for the two datasets are shown in Fig. 2. Each spectrum is the average, computed from 127, concurrent, 32 sample sequences. Peaking of the diffuse component near the horizon is seen to occur for both datasets. For the second dataset the peak point is seen to be somewhat unstable due to the greater difficulty of accurately resolving the direct and specular components (~0·1–0·2BW separation).

Conclusion: Thomson's multiple window method provides an invaluable tool for investigating diffuse multipath at low grazing angles. The results presented herein represent the first step towards a better understanding of the important physical phenomena involved, and can help in building a better theoretical model for diffuse multipath.*

A. DROSOPOULOS
S. HAYKIN

6th February 1991

Communications Research Laboratory
McMaster University
Hamilton, Ontario L8S 4K1, Canada

* DROSOPOULOS, A., and HAYKIN, S.: 'Theoretical and experimental characterization of diffuse multipath in a low-angle tracking radar environment'. In preparation

References

1 THOMSON, D. J.: 'Spectrum estimation and harmonic analysis', *Proc. IEEE*, 1982, **70**, pp. 1055–1096
2 THOMSON, D. J.: 'Time series analysis of Holocene climate data', *Phil. Trans. R. Soc. Lond. A*, **330**, pp. 601–616
3 DROSOPOULOS, A.: 'Investigation of diffuse multipath in a low-angle tracking radar environment'. PhD dissertation, McMaster University, Hamilton, Ont., 1991 (in preparation)

PERFECT AUTO-ASSOCIATORS USING RAM-TYPE NODES

Indexing terms: Neural networks, Random-access memories

A RAM-based auto-associative neural network is described which has several desirable properties, including high storage capacity and absence of minima during recall. The system is implemented as a set of generalising RAM (GRAM) type nodes. The generalisation procedure is described, and comparisons with other types of autoassociator are drawn.

The structure of the proposed network is shown in Fig. 1. Each bit in the output is generated from a RAM whose address lines are connected to all the bits of the input field. One RAM at a time is selected at random to update its output bit according to the contents of the location addressed by the current input pattern. The whole output field is then copied back to the input field and another RAM is updated. Each location holds two bits to store three possible values; 0, 1 or u (for undefined). If the addressed location contains 0 or 1 then the output bit takes on these values, respectively. If the addressed location contains a u value, then the output bit is chosen to be either 0 or 1 with equal probability. This type of node is called a PLN (probabilistic logic node).[1]

For a bit pattern (or state) to be stable in the network, a bit pattern on the input field must generate the same bit pattern on the output field. To create a stable state, the bit pattern is copied to the input field and accesses the same location in all of the RAMs. The content of this location in each RAM is set to the same value that the RAM has to produce on the output field, i.e. 0 or 1. Because every pattern accesses a different location in the RAM it can be seen that any set of possible binary patterns can be made stable in the network. For example, in an eight bit system any number of patterns from 0 up to 256 can be made stable. There is no danger of erasing some states when training additional stable states into the network.

We have seen how stable states may be trained into the network. At the moment, however, input patterns that are not

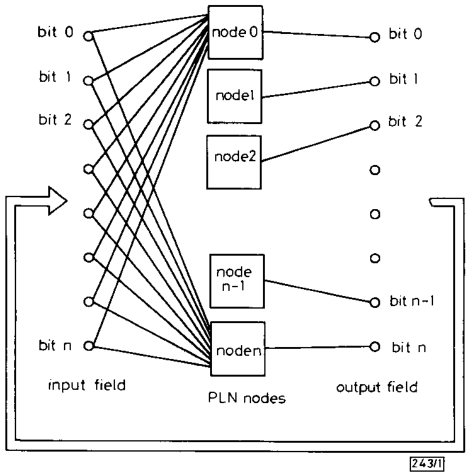

Fig. 1 *n-bit fully connected auto-associative network*
Output field copied to input field between node updates

stable states of the network are not recognised. An unknown pattern accesses a u filled location in every RAM, and the system will make random jumps one bit at a time as the individual RAMs update the output pattern. The output pattern wanders randomly until a stable state is found, which is not necessarily connected with the starting pattern in any way. By applying a 'generalising' or 'spreading' algorithm to the contents of the RAM it is possible to make the network move to the nearest stable state to the input pattern.

The spreading algorithm, first proposed by Aleksander[2] is a procedure for filling the contents of the u locations in each RAM according to the contents of locations that have similar addresses. A PLN that has been treated with the spreading procedure is known as a generalising RAM (GRAM). The spreading procedure consists of the following steps:

(1) starting from location 0, find the first location A that is filled with u

(2) examine all the locations whose addresses differ from the address of A by one bit

(3) if these locations hold a mixture of 1s and us then set the contents of A to 1; if these locations hold a mixture of 0s and us then set the contents of A to 0; if these locations hold a mixture of 1s and 0s and us then leave the contents of A as u

(4) go to the next location.

One complete pass through each RAM will typically spread the training data into neighbouring locations only. The process can be repeated up to n times, where n is the number of bits in the input or output field. For a fully spread network trained on one pattern, every pattern applied to the input will recall the single trained pattern.

Results: Experimental fully spread networks with between eight and 16 nodes were exhaustively tested with various numbers of stored patterns. In all cases, it was found that the system settled into the pattern from the training set that was nearest in terms of Hamming distance to the initial pattern presented to the network. This behaviour was observed regardless of the size of the network or the number of stored patterns. If two stored patterns were equally near to the starting pattern then the network would select either pattern with equal probability, averaged over a series of trials. At no time was a false minimum, i.e. a stable state not present in the training set, observed.

Discussion and conclusions: An auto-associator with very desirable properties has been described. The network always performs an optimal recall of the stored patterns, i.e. the nearest stored pattern in terms of Hamming distance is always retrieved, and there are never any problems with exceeding the capacity of the network because this is the maximum number of different patterns that can be applied to the input. The required stable state is automatically constructed at the output of the network. Other networks such as the Hamming net[3,4] perform an optimal comparison of the input pattern with the stored patterns, but this type of network only identifies the closest stored pattern without recalling it. Pattern recall is only obtained with extra circuitry. Tarassenko *et al.*[4] have demonstrated the superiority of the Hamming network over Hopfield-type nets for closest pattern recall. An important difference between these networks is that the Hamming nets use a localised weight arrangement, i.e. weight values for different patterns are not superimposed in the same place. This is also true for the GRAM nets, where different input patterns access completely distinct locations in memory.

GRAM auto-associators are simple to implement using conventional digital RAM technology. Their main disadvantage is the amount of RAM required to implement a large network. The full connectivity means that the number of locations required for a system with n bit input/output fields scales as an exponential function of n. Specifically

Number of locations required = $n2^n$

Number of bits required = $2n2^n$

Number of memory accesses required to fully spread the system after training = $n^2 2^n$.

These relations means that it would only be feasible to implement systems with field sizes smaller than 20. Research is currently being undertaken into larger networks composed of small fully connected GRAM nets.

J. LUCY 20th February 1991

Neural Systems Engineering Group
Department of Electrical Engineering
Imperial College
Exhibition Road, London SW7 2BT, United Kingdom

References

1 KAN, W., and ALEKSANDER, I.: 'A probabilistic logic neuron network for associative learning'. Proc. IEEE Int. Conf. on neural networks, San Diego, 1988, pp. 541–548
2 ALEKSANDER, I.: 'Ideal neurons for neural computers'. Proc. of Int. Conf. on Parallel Processing in Neural Systems and Computers (North Holland Elsevier Science, 1990), pp. 225–232
3 LIPPMANN, R. P.: 'An introduction to computing with neural nets', *IEEE ASSP Magazine*, April 1987, pp. 4–22
4 TARASSENKO, L., TOMBS, J. N., and REYNOLDS, J. H.: 'Neural network architectures for content addressable memory', *IEEE Proc. F*, 1991, **138**, (1), pp. 33–39

STABILISATION OF CLOSED LOOP SYSTEMS BY TIME DELAYS

Indexing terms: Closed loop systems, Stability

It is generally recognised that time delays will cause closed loop systems to become unstable. The adverse effect on stability caused by time delays is presented. Bounds on time delays for the stabilisation of unstable closed loop systems are also derived.

Introduction: The stability of closed loop systems with time delays has been investigated using various methods.[1–7] Most of them have shown the robust delay time for closed loop