REFERENCES

1. E. Ott, Chaos in dynamical systems, Cambridge University Press, Cambridge, England, 1992.
2. R.L. Devaney and L. Keens (Editors), Chaos and fractals: The mathematics behind the computer graphics, American Mathematical Society, Providence, RI, 1989.
3. L.M. Pecora and T.L. Carroll, Synchronization in chaotic systems, Phys Rev Lett 64 (1990), 821–824.
4. K.M. Cuomo and A.V. Oppenheim, Circuit implementation of synchronized chaos with applications to communications, Phys Rev Lett 71 (1993), 65–68.
5. S. Hayes, C. Grebogi, and E. Ott, Communicating with chaos, Phys Rev Lett 70 (1993), 3031–3034.
6. K.M. Cuomo, A.V. Oppenheim, and S.H. Strogatz, Synchronization of Lorentz-based chaotic circuits with applications to communications, IEEE Trans Circuits Syst II 40 (1993), 626–633.

CCC 0895-2477/99

A NEURAL-NETWORK-BASED LINEARLY CONSTRAINED MINIMUM VARIANCE BEAMFORMER

A. H. El Zooghby,[1] C. G. Christodoulou,[2] and M. Georgiopoulos[1]

[1] Department of Electrical and Computer Engineering
University of Central Florida
Orlando, Florida 32816

[2] Department of Electrical and Computer Engineering
University of New Mexico
Albuquerque, New Mexico 87131

Received 7 December 1998

ABSTRACT: *This paper presents a neural network approach for beamforming and interference cancellation. A three-layer radial basis function neural network is trained with input–output pairs. The results obtained from this network are in excellent agreement with the Wiener solution. It was found that networks implementing these functions are successful in tracking mobile users in real time as they move across the antenna's field of view.* © 1999 John Wiley & Sons, Inc. Microwave Opt Technol Lett 21: 451–455, 1999.

Key words: *adaptive array antennas; adaptive beamforming; neural networks; wireless communications; interference cancellation*

I. INTRODUCTION

New wireless systems such as cellular, personal communication systems (PCS), and personal communication networks (PCN) must satisfy an increasing demand for coverage, capacity, and service quality. For this purpose, the need for more powerful tools to improve different aspects of modern communications systems has become increasingly important. In recent years, it has become clear that the area of *smart antennas* will provide a key technological boom for the wireless communications industry [1–3]. This paper discusses the development of a *neural-network-based smart antenna* capable of performing adaptive beamforming and interference cancellation in real time. The base station array is able to track mobile users as they move within or between cells by allocating high-gain narrow beams in the directions of the desired users while simultaneously nulling unwanted sources of interference [4]. This space-division multiple access (SDMA) will improve the coverage as well as increase the system capacity of existing cellular and mobile communications systems [5–8].

Neural networks are gaining momentum in the field of signal processing [9–13] mainly because of their general-purpose nature, fast convergence rates, and new VLSI implementations. Motivated by these inherent advantages, this paper presents the development of a neural-network-based algorithm, which treats the problem of computing the weights of an adaptive array antenna as a mapping problem. The directions of the desired signals as well as the cochannel interference are estimated as described in [10], and this information is used to track the signals without the need for a reference signal. The organization of the paper is as follows. In Section II, a brief derivation of the optimum array weights for adaptive beamforming is presented. The RBFNN approach for the computation of the adaptive array weights is introduced in Section III. Finally, Section IV presents the simulation results, and Section V offers conclusive remarks.

II. ADAPTIVE BEAMFORMING AND INTERFERENCE NULLING

Consider a linear array composed of M elements. Let K $(K < M)$ be the number of narrowband plane waves, centered at frequency ω_0, impinging on the array from elevation angles $\{\theta_1, \theta_2, \ldots, \theta_K\}$. Using complex signal representation, the received signal at the ith element can be written as

$$x_i(t) = \sum_{m=1}^{K} S_m(t) e^{-j(i-1)2\pi\frac{d}{\lambda}\sin(\theta_m)} + n_i(t), \qquad i = 1, 2, \ldots, M \quad (1)$$

where $s_m(t)$ is the signal of the mth wave, $n_i(t)$ is the white Gaussian noise signal with zero mean and variance σ^2 received at the ith sensor, and d is the spacing between the elements of the array. Using vector notation, we can write the array output in matrix form:

$$\mathbf{X}(t) = \mathbf{A}\mathbf{S}(t) + \mathbf{N}(t) \quad (2)$$

where $\mathbf{X}(t)$, $\mathbf{S}(t)$, and $\mathbf{N}(t)$ are M-dimensional vectors, while $\mathbf{A}$ is the $M \times K$ steering matrix of the array toward the direction of the incoming signals defined as

$$\mathbf{A} = [a(\theta_1), a(\theta_2), \ldots, a(\theta_K)] \quad (3)$$

where $\mathbf{a}(\theta_m)$ is the steering vector toward the direction θ_m defined as

$$\mathbf{a}(\theta_m) = \left[1, e^{-j2\pi\frac{d}{\lambda}\sin(\theta_m)}, e^{-j4\pi\frac{d}{\lambda}\sin(\theta_m)}, \ldots, e^{-j(M-1)2\pi\frac{d}{\lambda}\sin(\theta_m)}\right]. \quad (4)$$

Steering vectors and matrices for planar and circular arrays can be found in [14] and [15], respectively. The weights of the array element outputs can be represented as an M-dimensional vector:

$$\mathbf{W} = [w_1, w_2, \ldots, w_M]^T. \quad (5)$$

Then the array output becomes

$$\mathbf{y}(t) = \sum_{i=1}^{M} w_i^* x_i(t) = \mathbf{W}^H \mathbf{X}(t). \quad (6)$$

The mean output power is thus given by

$$\mathbf{P} = E[\mathbf{y}(t)\mathbf{y}(t)^{*}] = \mathbf{W}^{H}\mathbf{R}\mathbf{W} \tag{7}$$

where $*$ denotes the conjugate and $\mathbf{R}$ is the spatial correlation matrix of the received signals defined as

$$\mathbf{R} = E[\mathbf{X}(t)\mathbf{X}(t)^{H}]. \tag{8}$$

In the above equation, H denotes the conjugate transpose, and $E[\ ]$ denotes the expectation value. To derive the optimal weight vector, the array output is minimized so that the desired signals are received with specific gain, while the contributions due to noise and interference are minimized. In other words,

$$\min \mathbf{W}^{H}\mathbf{R}\mathbf{W}, \qquad \text{subject to } \mathbf{W}^{H}\mathbf{S}_{d} = \mathbf{r}. \tag{9}$$

In the above equation, $\mathbf{r}$ is the $V \times 1$ constraint vector, where V is the number of desired signals, and $\mathbf{S}_d$ is the steering matrix associated with the look directions as defined in Eqs. (3) and (4). The method of Lagrange multipliers is used to solve the constrained minimization problem in (9). It can be shown that the optimum weight vector is given by the following equation:

$$\hat{\mathbf{W}}_{\text{opt}} = \mathbf{R}^{-1}\mathbf{S}_{d}[\mathbf{S}_{d}^{H}\mathbf{R}^{-1}\mathbf{S}_{d}]^{-1}\mathbf{r}. \tag{10}$$

Since the above equation is not practical for real-time implementation, an adaptive algorithm must be used to adapt the weights of the array in order to track the desired signal and to place nulls in the direction of the interfering signals.

III. NEURAL-NETWORK-BASED INTERFERENCE CANCELLATION

This section describes a new implementation for the problem of beamforming using neural networks. The optimum weight vector is a nonlinear function of the correlation matrix and the constraint matrix [see Eq. (10)]. Therefore, it can be approximated using a suitable architecture such as a radial basis function neural network [18]. Note that a radial basis function neural network can approximate an arbitrary function from an input space of arbitrary dimensionality to an output space of arbitrary dimensionality [16–18]. The array outputs are preprocessed, and then applied to the RBFNN as shown in Figure 1. The output of the network is the optimal weight vector. The preprocessing consists of evaluating the radiation pattern of the antenna. As can be seen from Figure 1, the RBFNN consists of three layers of nodes: the input layer, the output layer, and the hidden layer. The input layer is the layer where the inputs are applied; the output layer is the layer where the outputs are produced. The input vector to the network is the spatial correlation matrix $\mathbf{R}$. By exploiting the symmetry in correlation matrix $\mathbf{R}$, one need only consider either the upper or lower triangular part of the matrix. In our design, the upper triangular half of $\mathbf{R}$ is used. An $M \times M$ spatial correlation matrix $\mathbf{R}$ can be rearranged in an $M(M+1)$-dimensional vector of real and imaginary parts denoted $\mathbf{b}$. This procedure is illustrated in Table 1 where the lower triangular half of $\mathbf{R}$ is removed.

The output layer consists of $2M$ nodes (1 – D case) or $2MN$ nodes (2-D case) to accommodate the output vector (i.e., $\mathbf{W}_{\text{opt}}$). Like most neural networks, the RBFNN is designed to perform an input–output mapping, trained with

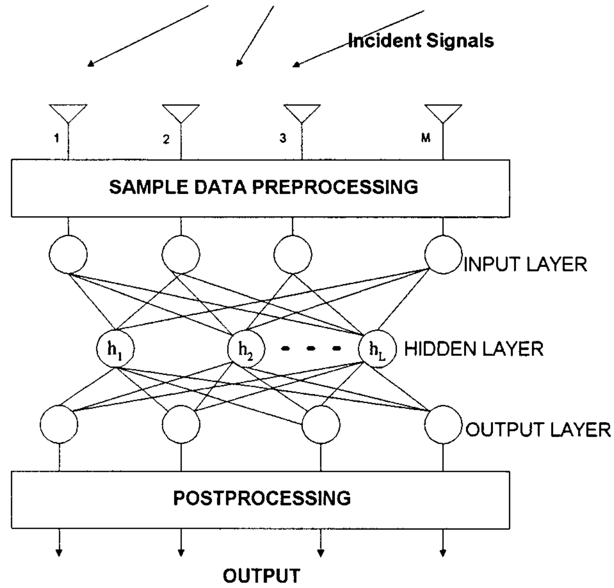

Figure 1 Architecture of a three-layer RBFNN for array processing applications

TABLE 1 Correlation Matrix Reduction

$\mathbf{R} = \begin{bmatrix} \mathbf{r}_{11} & \mathbf{r}_{12} & \mathbf{r}_{13} \\ r_{21} & \mathbf{r}_{22} & \mathbf{r}_{23} \\ r_{31} & r_{32} & \mathbf{r}_{33} \end{bmatrix}$	$\mathbf{b} = [\mathbf{r}_{11} \quad \mathbf{r}_{12} \quad \mathbf{r}_{13} \quad \mathbf{r}_{22} \quad \mathbf{r}_{23} \quad \mathbf{r}_{33}]$

examples of input–output pairs $(\mathbf{R}^{l}; \mathbf{W}_{\text{opt}}^{l})$; $l = 1, 2, \ldots, N_T$, where N_T stands for the number of examples contained in the training set. The purpose of the hidden layer in an RBFNN is to transform the input data $\mathbf{R}$ from an input space of dimensionality J to a space of higher dimensionality L (see Fig. 1). The rationale behind this transformation is based on Cover's theorem [19], which states that an input/output mapping problem cast in a high-dimensionality space nonlinearly is easier to solve. The nonlinear functions (the hs in Fig. 1) that perform this transformation are usually taken to be Gaussian functions of appropriately chosen means and variances. There are many learning strategies that have appeared in the literature to train an RBFNN. The one used in this paper was introduced in [18], where an unsupervised learning algorithm (such as the K-means [20]) is initially used to identify the centers of the Gaussian functions used in the hidden layer. Then, an ad hoc procedure is used to determine the widths (standard deviations) of these Gaussian functions. According to this procedure, the standard deviation of a Gaussian function of a certain mean is the average distance to the first few nearest neighbors of the means of the other Gaussian functions. The aforementioned unsupervised learning procedure allows one to identify the weights (means and standard deviations of the Gaussian functions) from the input layer to the hidden layer. The weights from the hidden layer to the output layer are identified by following a supervised learning procedure, applied to a single-layer network (the network from the hidden to the output layer). This supervised rule is referred to as the *delta rule*. The delta rule is essentially a gradient descent procedure applied to an appropriately defined optimization problem. For more details about the delta rule and how it is applied to single-layer networks, see [16]. Once training of the RBFNN is accomplished, the training phase is complete, and the trained neural network can operate in the performance mode (phase). In the *performance phase*, the neural network is expected to generalize, that is, respond to inputs (**R**s) that it has never seen before, but drawn from the same distribution as the inputs used in the training set. One way of explaining the generalization exhibited by the network during the performance phase is by remembering that, after the training phase is complete, the RBFNN has established an approximation of the desired input/output mapping. Hence, during the performance phase, the RBFNN produces outputs to previously unseen inputs by interpolating between the inputs used (seen) in the training phase. The step-by-step procedure to produce the training data $\{\mathbf{R}^l; \mathbf{W}^l_{\text{opt}}; l = 1, 2, \ldots, N_T\}$ for the RBFNN in this application is provided below. In practice, the directions of the multiple desired signals are estimated so that the matrix $\mathbf{S}_d$ is incorporated as *a priori* information in the training. This is accomplished by the algorithm described in [10].

A. Generation of Training Data

1. Generate the correlation matrix $\{\mathbf{R}^l;\ l = 1, 2, \ldots, N_T\}$ using Eq. (8).
2. Rearrange the upper triangular part of **R** into a vector **b**; then normalize it by its norm.
3. Produce the required training input/output pairs of the training set, that is, $\{(\mathbf{R}^l; \mathbf{W}^l_{\text{opt}});\ l = 1, 2, \ldots, N_T\}$. In this application, the training data were generated by assuming that multiple sources are located at elevation angles θ ranging from -90 to $+90°$ with increments of $\Delta\theta$ for the one-dimensional case. In other words, if we have four sources with $\Delta\theta = 5°$, then $\mathbf{R}^l$ would be based on sources at $\{-90°, -85°, 80°, 75°\}$, $\mathbf{R}^2$ based on sources at $\{-88°, -83°, -78°, -73°\}$, and so forth. In the two-dimensional array, in addition to angles θ, azimuth angles ϕ can be made to range from 0 to 360° in order to span the field of view of the antenna. As we have emphasized before, once the RBFNN is trained with a representative set of training input/output pairs, it is ready to function in the performance phase. In the performance phase, the RBFNN produces estimates of the optimum weights for the array outputs through a simple, computationally inexpensive, two-step process, described below.

B. Performance Phase of the RBFNN

1. Follow steps 1 and 2 in the training procedure.
2. Present the normalized array output vector at the input layer of the trained RBFNN. The output layer of the trained RBFNN will produce, as an output, the estimates of optimum weights for the array outputs (i.e., $\hat{\mathbf{W}}_{\text{opt}}$).

Unlike the least mean-square, recursive least squares, or the sample matrix inversion algorithms [4], where the optimization is carried out whenever the directions of the desired or interfering signals change, in our approach, the weights of the trained network can be used to produce the optimum weights needed to steer the narrow beams of the adaptive array to the direction of desired users. Knowing that the response time for neural networks (i.e., the time that it takes a trained neural network to produce an output if it is excited by an input) is very small, the proposed adaptive beamforming technique will track the mobile users as they move in real time.

IV. SIMULATION RESULTS

The pattern of an array of eight elements receiving one desired and three interfering signals is shown in Figure 2. The SNR of the sources is 10 dB with respect to the noise. The input to the network consisted of all of the elements of the correlation matrix **R**. Hence, the dimension of the input layer is 128 nodes. In Figure 3 an array of ten elements is simulated under the same conditions with 110 input nodes, where only the upper triangular part of **R** was used as the input. The results show that it is possible to reduce the dimension of the input layer significantly without affecting the interpolation capabilities of the network. Figure 4 illustrates an array of 16 elements receiving seven signals, four of which are interference. The SNRs of the desired signals were set to 10 dB, while those of the interfering signals were set to 20 dB and $\Delta\theta = 5°$. A 4×4 planar array is simulated in Figure 5, and is trained to track seven signals, three of which are desired with $\Delta\theta = 12°$ and $\phi = 60°$. Figure 6 shows the

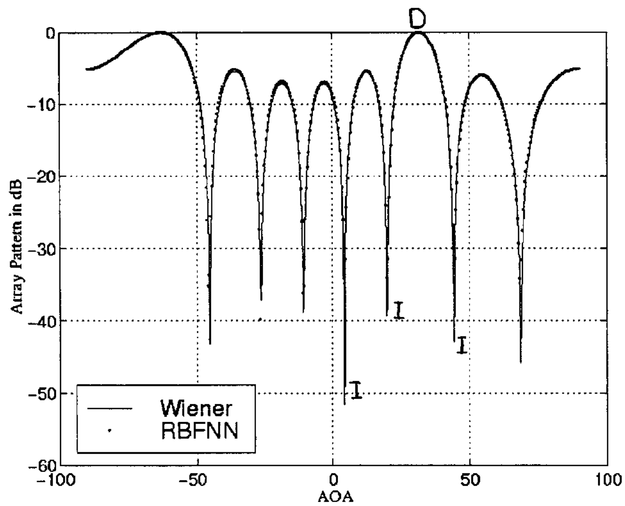

Figure 2 Adapted pattern of an eight-element array receiving one desired signal and three interfering. The SNR of the sources is 10 dB

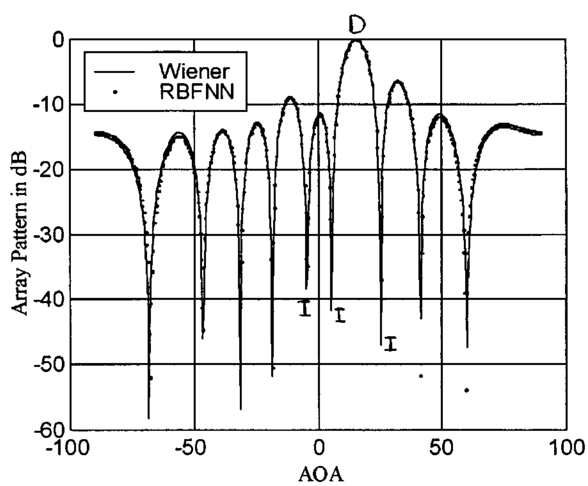

Figure 3 Adapted pattern of a ten-element array receiving one desired signal and three interfering. The SNR of the sources is 10 dB

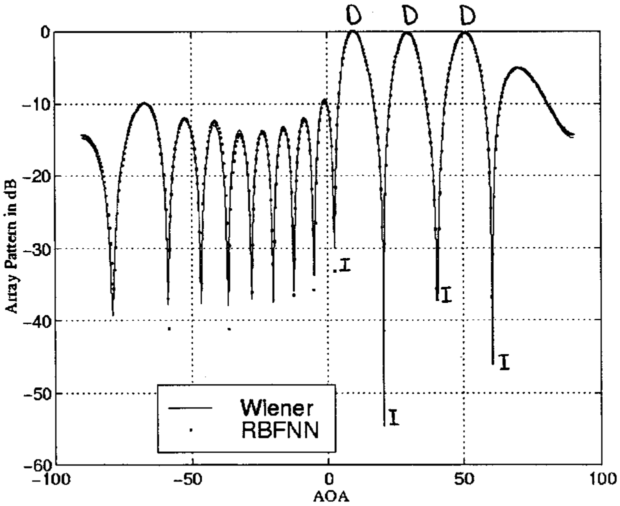

Figure 4 Adapted pattern of a 16-element array tracking seven signals, four of which are interference. The SNR of desired signals is 10 dB, the SNR of interfering signals is 20 dB, and $\Delta\theta = 5°$

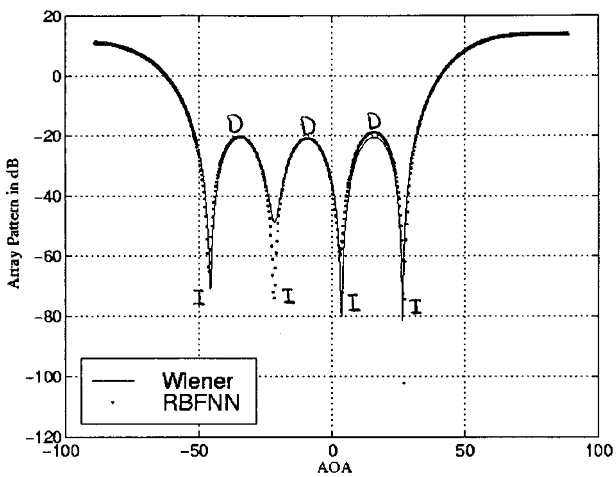

Figure 5 Gain of a 4×4 rectangular array shown tracking seven signals (three desired users, four cochannel signals) with $\Delta\theta = 12°$, $\phi = 60°$

tracking of two desired signals and one cochannel user with a ten-element (nine + one at center) circular array of 0.8λ radius. The signals are equipower (10 dB) with respect to the noise (0 dB). The angular separation is $\Delta\theta = 15°$ and $\phi = 40°$. The results obtained from the simulations demonstrate the accuracy of the proposed approach. The RBFNN was simulated on a 300 MHz Pentium II computer. The network response time to the various test data sets containing 100–150 input vectors was 1–2 s, indicating that the average response time to a single input is 0.01–0.02 s. Hardware implementation of the RBFNN is expected to achieve much higher speeds, which makes this approach suitable to use in real time.

V. CONCLUSION

A new approach to the problem of adaptive beamforming was introduced. The weights were computed using an RBFNN that approximates the Wiener solution. The network was successful in tracking multiple users, while simultaneously nulling interference caused by cochannel users. Both 1-D and

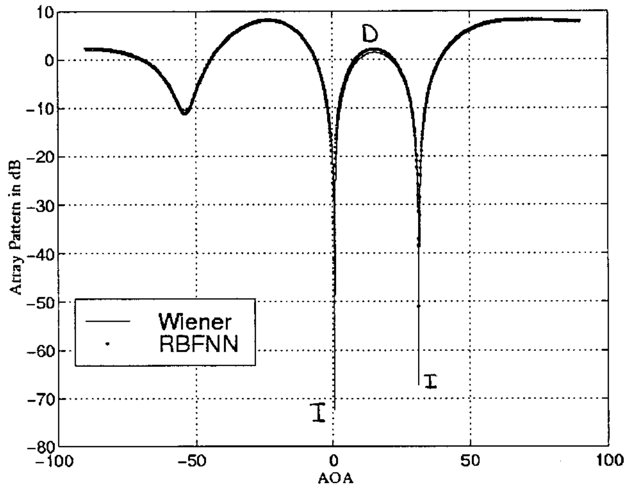

Figure 6 Tracking of one desired signal and two cochannel user with a ten-element (nine + one at center). Circular array of 0.8λ radius, equipower signals (10 dB), noise (0 dB). $\Delta\theta = 15°$, $\phi = 40°$

2-D arrays were simulated, and the results have been very good in every case. A comparison of the adapted pattern obtained by the RBFNN and the optimum solution demonstrates the high degree of accuracy of our approach. Some of the attractive features of this *novel* neural-network-based approach is that it can yield results in *real time*, hence outperforming other conventional techniques. This leads to the accurate estimation of the mobile location and the computation of the weights of the adaptive array antennas in a few characteristic time constants of the circuit, normally, on the order of hundreds of nanoseconds. Furthermore, digital implementations of neural networks make them ideal candidates for digital beamforming. Moreover, conventional beamformers require highly calibrated antennas with identical element properties. Performance degradation often occurs due to the fact that these algorithms adapt poorly to element failure or other sources of errors. Neural-network-based array antennas do not suffer from this shortcoming. The antenna behavior (uniform, nonuniform spacing, nonuniform elements, etc.) can be incorporated in the training of the neural network under different circumstances and scenarios. The network, being able to generalize, can then be used to predict the aperture behavior at all points. Future work will concentrate on: 1) the effects of element patterns, 2) finding the maximum number of simultaneous users that can be tracked by the antenna array using this new approach, and 3) improving the generalization capability of the network so that it can give satisfactory results for a different environment than it was trained for, i.e., for example, if the number of desired users and interference or the angular separations are not the same as in the training.

ACKNOWLEDGMENT

This work was supported in part by the Florida Space Consortium and NeuralWare Inc.

REFERENCES

1. T. Gebauer and H.G. Gockler, Channel-individual adaptive beamforming for mobile satellite communications, IEEE J Select Areas Commun 13 (1995), 439–448.
2. S. Swales, M. Beach, D. Edwards, and J. Mcgeehan, The performance enhancement of multibeam adaptive base-station anten-

nas for cellular land mobile radio systems, IEEE Trans Veh Technol 39 (1990), 56–67.
3. A.H. El Zooghby and C.G. Christodoulou, Optimum beamforming for co-channel interference nulling in mobile satellite communications, IEEE AP-S/URSI Int Symp, Baltimore, MD, July 1996, vol. 1, pp. 552–525.
4. Mozingo and Miller, Introduction to adaptive arrays, Wiley, New York, 1980.
5. A.F. Naguib, A. Paulraj, and T. Kailath, Capacity improvement with base-station antenna arrays in cellular CDMA, IEEE Trans Veh Technol 43 (1994), 691.
6. S. Nordholm, I. Claesson, and B. Bengtsson, Adaptive array noise suppression of hands free speaker input in cars, IEEE Trans Veh Technol 42 (1993), 514.
7. A.K. Djedid and M. Fujita, Adaptive array sensor processing applications for mobile telephone communications, IEEE Trans Veh Technol 45 (1996), 405.
8. T. Ohgane, T. Shimura, N. Matsuzawa, and H. Sasaoka, An implementation of a CMA adaptive array for high speed GMSK transmission in mobile communications, IEEE Trans Veh Technol 42 (1993), 282.
9. H.L. Southall, J.A. Simmers, and T.H. O'Donnell, Direction finding in phased arrays with a neural network beamformer, IEEE Trans Antennas Propagat 43 (1995), 1369.
10. A.H. El Zooghby, C.G. Christodoulou, and M. Georgiopoulos, Performance of radial basis function networks for direction of arrival estimation with antenna arrays, IEEE Trans Antennas Propagat 45 (1997), 1611–1617.
11. L. Long and L.Y. Da, Real-time computation of the noise subspace for the MUSIC algorithm, Proc ICASSP 1993, Apr. 1993, vol. 1, pp. 485–488.
12. D. Goryn and M. Kaveh, Neural networks for narrowband and wideband direction finding, Proc ICASSP, Apr. 1988, pp. 2164–2167.
13. P.R. Chang, W.H. Yang, and K.K. Chan, A neural network approach to MVDR beamforming problem, IEEE Trans Antennas Propagat 40 (1992), 313–322.
14. S.J. Yu and J.H. Lee, Design of two-dimensional rectangular array beamformers with partial adaptivity, IEEE Trans Antennas Propagat 45 (1997), 157–167.
15. A.H. El Zooghby, C.G. Christodoulou, and M. Georgiopoulos, Adaptive interference cancellation in circular arrays with radial basis function neural networks, IEEE AP/URSI Symp, Atlanta, GA, 1998.
16. S. Haykin, Neural networks, A comprehensive foundation, Macmillan College, Ontario, Canada, 1994.
17. B. Mulgrew, Applying radial basis functions, IEEE Signal Processing Mag 13 (1996), 50–65.
18. T.J. Moody and C.J. Darken, Fast learning in networks of locally tuned processing units, Neural Comput (1989), 289.
19. T.M. Cover, Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition, IEEE Trans Electron Comput EC-14 (1965), 326–344.
20. J.T. Tou and R.C. Gonzalez, Pattern recognition principles, Addison-Wesley, Reading, MA, 1976.

FLUID DIELECTRICS MEASUREMENT TECHNIQUE USING A WAVEGUIDE SLOT BRIDGE

Nataliya P. Yashina,[1,2] Sergey I. Tarapov,[1] Alex A. Vertiy,[1,3,4] Ertugrul Karacuha,[2] and Fatih Dikmen[2]
[1] Institute of Radiophysics and Electronics
National Academy of Science
Kharkov 310085, Ukraine
[2] Gebze Institute of Technology
41400 Gebze-Kocaeli, Turkey
[3] Fonon
Kiev, Ukraine
[4] Marmara Research Center
Turkey

Received 22 December 1998

ABSTRACT: *An experimental method is suggested that supposes the study of characteristics of slot bridge (SB) cells in the multimode operating regime. It is based on the preliminary numerical investigation of the resonant scattering characteristics of SB, i.e., the solution of the boundary value problem (BVP). The configuration of an experimental SB cell is computed according to the required frequency band and preliminary information about dielectric parameters of the materials chosen for tests. Etalon scattering characteristics of an SB cell are computed by software development on the basis of a rigorous solution of the BVP, providing numerical data with required accuracy.* © 1999 John Wiley & Sons, Inc. Microwave Opt Technol Lett 21: 455–458, 1999.

Key words: *rigorous mathematical model; microwave band; dielectric parameters; SB cell*

1. INTRODUCTION

An accurate mathematical model describing electromagnetic resonant scattering properties of a coaxial slot bridge (SB) has been developed in [1]. The developed numerical algorithm is a powerful tool for the investigation of diffraction and spectral characteristics of the cell. The preliminary experiments relying on this model proved that the coaxial SB cell manifests remarkable advantages in fluid dielectric parameters control, in industrial and technological applications.

The features that distinguish these structures from the formerly known ones are the following.

- Sufficiently good correspondence of real devices to their numerical model.
- Technological simplicity of manufacturing of a real SB cell that can be easily integrated into the pipelines with objects under testing. Materials used for SB cell construction provide ecological safety of the control technique.
- The possibility of real-time dielectric parameter control by fitting a programmable single-chip computer to a technological SB cell.

2. THEORETICAL BACKGROUND

The considered SB is depicted in Figure 1. The walls of the circular waveguides are assumed infinitely thin and perfectly conducting.

The diffraction problem for such an electromagnetic structure is formulated (see [1]) as a BVP problem of scattered field amplitude determination in SB.

The solving procedure is based on the semi-inversion method. Namely, the main singular part is extracted from the