Progress In Electromagnetics Research, PIER 72, 279–306, 2007

RESONANT FREQUENCY CALCULATION FOR CIRCULAR MICROSTRIP ANTENNAS WITH A DIELECTRIC COVER USING ADAPTIVE NETWORK-BASED FUZZY INFERENCE SYSTEM OPTIMIZED BY VARIOUS ALGORITHMS

K. Guney

Department of Electrical and Electronics Engineering
Faculty of Engineering
Erciyes University
38039 Kayseri, Turkey

N. Sarikaya

Department of Aircraft Electrical and Electronics
Civil Aviation School
Erciyes University
38039 Kayseri, Turkey

Abstract—This paper presents a method based on adaptive-network-based fuzzy inference system (ANFIS) to calculate the resonant frequency of a circular microstrip antenna (MSA) with a dielectric cover. The ANFIS is a class of adaptive networks which are functionally equivalent to fuzzy inference systems (FISs). Six optimization algorithms, hybrid learning, least-squares, nelder-mead, genetic, differential evolution and particle swarm, are used to determine optimally the design parameters of the ANFIS. The resonant frequency results predicted by ANFIS are in very good agreement with the results reported elsewhere. When the performances of ANFIS models are compared with each other, the best result is obtained from the ANFIS model optimized by the LSQ algorithm.

1. INTRODUCTION

Microstrip antennas (MSAs) have many attractive features such as light weight, low production cost, low profile, integrability into arrays, conformability to curved surfaces, simplicity of fabrication, and ease of integration with microwave integrated circuit (MIC) or monolithic microwave integrated circuit (MMIC) components [1–17]. These attractive features have recently increased the application of MSAs and stimulated greater effort to investigate their performance.

A dielectric cover or superstrate is usually employed in practice to protect the MSAs from environmental hazards, or may be naturally formed (e.g., ice) during flight or severe weather conditions. Due to the limited bandwidth of MSAs, it is important to determine the effect of a dielectric cover on the resonant frequency of MSAs in order to introduce appropriate corrections in the design of MSAs.

Several methods [18–24], varying in accuracy and computational effort, have been proposed and used to calculate the resonant frequency of a circular MSA with a dielectric cover. The Hankel transform analysis [18], the modified Wolff model [19], Galerkin's method in the Hankel transform domain [22], a formula [23] based on an improved cavity model [25] and Bernhard's work [26] were presented for computing the resonant frequency of a circular MSA with a dielectric cover. The spectral-domain moment-method proposed by Fan and Lee [20] can also be used to calculate the resonant frequency. Lee and Fan [21] presented a Fortran program of CAD (Computer-Aided Design) formula for the resonant frequency of a circular MSA with a dielectric cover. This CAD formula was obtained by using a database built by moment-method. Closed-form Hankel transforms for circular disk basis modes involving Chebyshev polynomials and edge condition closed-form expressions for two kinds of Hankel transform integrals, which are encountered in the spectral moment method solution of a circular patch, were obtained in [24]. These expressions were applied in the formulation of the the full-wave moment method solution for the resonance frequency of a circular microstrip patch antenna with a dielectric cover. It is clear that the methods used in calculating the resonant frequency can be broadly classified into two categories: analytical and numerical methods. The analytical methods, based on some fundamental simplifying physical assumptions regarding the radiation mechanism of antennas, are the most useful for practical design as well as providing a good intuitive explanation of the operation of MSAs. However, these methods are not suitable for many structures, in particular, if the thickness of the substrate is not very thin. The numerical techniques provide accurate results but usually require

considerable computational time and costs.

In this paper, a method for computing the resonant frequency of the circular MSA with a dielectric cover, based on the adaptive-network-based fuzzy inference system (ANFIS) [27, 28], is presented. First, the antenna parameters related to the resonant frequency are determined, and then the resonant frequency depending on these parameters is calculated by using the ANFIS. The ANFIS is a class of adaptive networks which are functionally equivalent to fuzzy inference systems (FISs). The FIS is a popular computing framework based on the concepts of fuzzy set theory, fuzzy if-then rules, and fuzzy reasoning. The ANFIS can simulate and analysis the mapping relation between the input and output data through a learning to determine optimal parameters of a given FIS. It can be trained with no need for the expert knowledge usually required for the standard fuzzy logic design. A prominent advantage of the ANFIS is that, after proper training, ANFIS completely bypasses the repeated use of complex iterative processes for new cases presented to it. Even if training takes a few minutes, the test process takes only a few microseconds. Because of these attractive features, the ANFIS in this paper is used to calculate the resonant frequency of a circular MSA with a dielectric cover.

In previous works [29–38], we successfully utilized ANFIS for computing accurately the various parameters of the rectangular, triangular, and circular MSAs, and for tracking multiple targets and estimating the phase inductance of the switched reluctance motors. In reference [32], the resonant frequency of circular MSA *without a dielectric cover* has been computed by using ANFIS. In this paper, ANFIS is employed to compute the resonant frequency of circular MSA *with a dielectric cover*. In previous works [29–38], only the hybrid learning (HL) algorithm [27, 28] was used to determine the optimum design parameters of the ANFIS. However, in this study, six different optimization algorithms, HL algorithm, least-squares (LSQ) algorithm [39–41], nelder-mead (NM) algorithm [42, 43], genetic algorithm (GA) [44, 45], differential evolution algorithm (DEA) [46–48], and particle swarm optimization (PSO) [49, 50], are used to determine optimally the design parameters of the ANFIS. These optimization algorithms are employed to obtain better performance and faster convergence with simpler structure.

In this paper, the next section briefly describes the spectral-domain moment-method [20, 21] for computing the resonant frequency of the circular MSA with a dielectric cover. The application of the ANFIS to the resonant frequency computation is given in the following section. The results are then presented and conclusion is made.

2. RESONANT FREQUENCY OF A CIRCULAR MICROSTRIP ANTENNA WITH A DIELECTRIC COVER

The basic geometry of a circular MSA covered by a dielectric layer and the co-ordinate system under consideration is depicted in Figure 1. The patch is circular with radius a and is fed through coaxial probe.

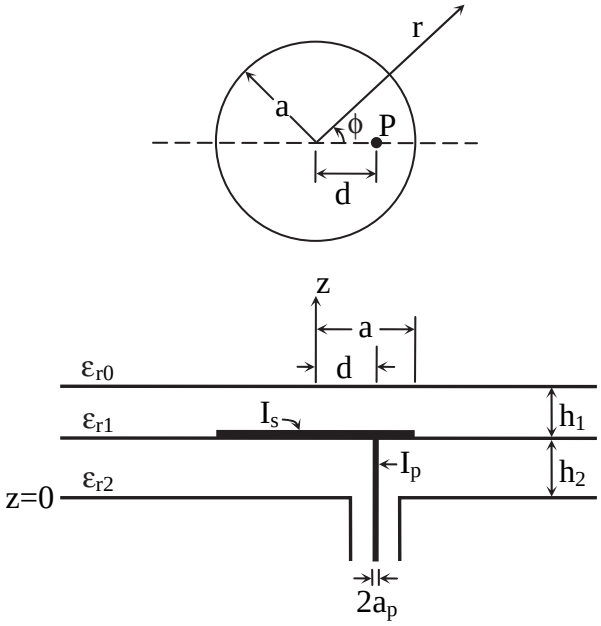

Figure 1. Geometry of the circular microstrip antenna with a dielectric cover.

The coaxial probe is modeled by a uniform cylindrical current sheet of radius a_p. The substrate-dielectric cover thicknesses and relative dielectric constants are h_2, h_1 and, ε_{r2}, ε_{r1}, respectively. The resonant frequency of the circular MSA with a dielectric cover can be obtained by using the spectral-domain moment-method proposed by [20, 21]. In this method, the problem is formulated in the Hankel transform domain. First, the integral equations are obtained for the surface current density on the patch. Galerkin's method is then used to solve for the surface current density, and the resonant frequency is calculated by using the input impedance formula. The surface current density $\mathbf{I}_p$ is given by the following equation for 1A total current in the coaxial

probe [20]:

$$\mathbf{I}_p = \hat{z} I_p(r,\phi) = \hat{z}\frac{1}{2\pi}\frac{\delta\left[|\mathbf{r}-\hat{x}d|-a_p\right]}{|\mathbf{r}-\hat{x}d|} \tag{1}$$

where d is the feed location. The problem can be solved either with vector Hankel transforms or Hankel transforms with the r and ϕ components in quadrature [51]. The following vector $\mathbf{F}$ is defined by

$$F_\pm = F_r \pm jF_\phi \tag{2a}$$

The Hankel transform quantities $\tilde{F}_\pm$ are given by

$$\tilde{F}_\pm(\alpha) = \int_0^\infty F_\pm(r)\, J_{n\pm1}(\alpha r)\, r dr \tag{2b}$$

The following integral equations are obtained for the unknown patch current densities [20].

$$\int_0^\infty \left[G_{11}\tilde{I}_{s+} + G_{12}\tilde{I}_{s-} + H_1\right] J_{n+1}(\alpha r)\,\alpha\, d\alpha = 0, \text{ on the patch} \tag{3a}$$

$$\int_0^\infty \left[G_{21}\tilde{I}_{s+} + G_{22}\tilde{I}_{s-} + H_2\right] J_{n-1}(\alpha r)\,\alpha\, d\alpha = 0, \text{ on the patch} \tag{3b}$$

with

$$G_{11} = G_{22} = \frac{\zeta_0}{2k_0}\left[\frac{k_0^2 S_2}{Q} + \frac{R_2}{D}\right] \tag{4a}$$

$$G_{12} = G_{21} = \frac{\zeta_0}{2k_0}\left[\frac{k_0^2 S_2}{Q} - \frac{R_2}{D}\right] \tag{4b}$$

$$H_1 = -H_2 = \frac{\zeta_0 R_2}{2k_0 D}\frac{2\alpha\tilde{I}_p}{\gamma_2^2} \tag{4c}$$

$$S_2 = j\gamma_0 \tan\gamma_1 h_1 + \gamma_1 \tag{4d}$$

$$R_2 = \gamma_1\gamma_2\left(j\gamma_1 \tan\gamma_1 h_1 + \varepsilon_{r1}\gamma_0\right) \tag{4e}$$

$$\zeta_0 = \sqrt{\frac{\mu_0}{\varepsilon_0}}, \qquad k_0^2 = w^2\mu_0\varepsilon_0 \tag{4f}$$

$$\begin{aligned} D = {} & \varepsilon_{r1}\gamma_1\left(-\gamma_2 + j\varepsilon_{r2}\gamma_0\cot\gamma_2 h_2\right) \\ & - \left(j\varepsilon_{r1}^2\gamma_0\gamma_2 + \varepsilon_{r2}\gamma_1^2\cot\gamma_2 h_2\right)\tan\gamma_1 h_1 \end{aligned} \tag{4g}$$

$$Q = \gamma_1 \left(-\gamma_0 + j\gamma_2 \cot \gamma_2 h_2\right) - \left(j\gamma_1^2 + \gamma_0\gamma_2 \cot \gamma_2 h_2\right) \tan \gamma_1 h_1 \quad (4h)$$

$$\tilde{I}_p = \frac{1}{2\pi} J_n(\alpha d)\, J_0(\alpha a_p) \quad (4i)$$

$$\gamma_i = \sqrt{\varepsilon_{ri} k_0^2 - \alpha^2}, \quad i = 0,\, 1,\, 2. \quad (4j)$$

Galerkin's method can be used to solve Eq. (3). The surface patch current density $I_{s\pm}$ is expanded in terms of basis functions:

$$I_{s\pm}(r,\phi) = \sum_{n=-\infty}^{\infty} e^{jn\phi} \sum_{m=1}^{M} C_{m\pm}(n) I_{sm\pm}(r,n) \quad (5)$$

The surface patch current density $I_{s\pm}$ in the Hankel transform domain can be written as

$$\tilde{I}_{s\pm}(\alpha,n) = \sum_{m=1}^{M} C_{m\pm}(n) \tilde{I}_{sm\pm}(\alpha,n) \quad (6)$$

The following equations are obtained by substituting Eq. (6) into Eq. (3) and multiplying the resulting equation by $rI_{si\pm}(r,n)$ and integrating from $r = 0$ to ∞,

$$\sum_{m=1}^{M} \left(Z_{im}^{11} C_{m+} + Z_{im}^{12} C_{m-}\right) = V_i^1, \quad i = 1, \ldots, M \quad (7a)$$

$$\sum_{m=1}^{M} \left(Z_{im}^{21} C_{m+} + Z_{im}^{22} C_{m-}\right) = V_i^2, \quad i = 1, \ldots, M \quad (7b)$$

with

$$Z_{im}^{11}(n) = \int_0^{\infty} \tilde{I}_{si+} G_{11} \tilde{I}_{sm+} \alpha d\alpha = Z_{mi}^{11}(n) \quad (8a)$$

$$Z_{im}^{12}(n) = \int_0^{\infty} \tilde{I}_{si+} G_{12} \tilde{I}_{sm-} \alpha d\alpha = Z_{mi}^{21}(n) \quad (8b)$$

$$Z_{im}^{22}(n) = \int_0^{\infty} \tilde{I}_{si-} G_{22} \tilde{I}_{sm-} \alpha d\alpha = Z_{mi}^{22}(n) \quad (8c)$$

$$V_i^1(n) = -\int_0^\infty \tilde{J}_{si+} H_1 \alpha d\alpha \tag{9a}$$

$$V_i^2(n) = -\int_0^\infty \tilde{J}_{si-} H_2 \alpha d\alpha \tag{9b}$$

The input impedance can be expressed as [20, 21, 52]

$$Z_{in} = \pi \sum_{n=-\infty}^{\infty} \left[\sum_{m=1}^{M} C_{m+} V_m^1 + \sum_{m=1}^{M} C_{m-} V_m^2 \right] + j x_p \tag{10a}$$

with

$$x_p = \frac{\zeta_0}{\sqrt{\varepsilon_{r2}}} \tan(k_0 h_2) \tag{10b}$$

The resonant frequency is the frequency at which the input resistance is maximum.

The space-domain basis functions can be chosen as [20]

$$I_{sm\pm}(r, n) = J_{n\pm1}(k_{mn} r), \quad r < a \tag{11}$$

where k_{mn} is the mth root of $J'_n(k_{mn}a) = 0$. The prime sign denotes the derivative of Bessel function $J_n(x)$ with respect to the argument x. The Hankel transform is given by

$$\begin{aligned} \tilde{I}_{sm\pm}(\alpha, n) = {} & a \left[\alpha J_{n\pm1}(k_{mn}a) J'_{n\pm1}(\alpha a) \right. \\ & \left. - k_{mn} J_{n\pm1}(\alpha a) J'_{n\pm1}(k_{mn}a) \right] / \left(k_{mn}^2 - \alpha^2 \right) \end{aligned} \tag{12}$$

From the formulations given above and in the literature [18–24] we see that only five parameters, a, h_1, ε_{r1}, h_2, and ε_{r2}, are needed to describe the resonant frequency of the circular MSA with a dielectric cover. It is clear that the method described above is mathematically complex and requires high performance large-scale computer resources and a very large number of computations. For this reason, in this work, the resonant frequency of the circular MSA with a dielectric cover is computed by using a method based on ANFIS. Only five parameters, a, h_1, ε_{r1}, h_2, and ε_{r2}, are used in calculating the resonant frequency.

3. APPLICATION OF ANFIS TO THE COMPUTATION OF RESONANT FREQUENCY

The ANFIS [27, 28] is a class of adaptive networks which are functionally equivalent to FISs. It is a very powerful approach for

building complex and nonlinear relationship between a set of input and output data. The ANFIS used in this paper implements a first-order Sugeno fuzzy model [27, 28]. Among many FIS models, the Sugeno fuzzy model is the most widely applied one for its high interpretability and computational efficiency, and built-in optimal and adaptive techniques.

The ANFIS architecture used in this work for the resonant frequency calculation of circular MSAs with a dielectric cover is illustrated in Figure 2, in which a circle indicates a fixed note, whereas a rectangular indicates an adaptive note. For the ANFIS, the inputs are h_1/a, h_2/a, ε_{r1}, and ε_{r2}, and the output is the normalized resonant frequency f_r/f_{no}. f_r and f_{no} represent, respectively, the resonant frequencies of the TM_{11} mode (dominant mode) of the circular MSA with and without dielectric cover.

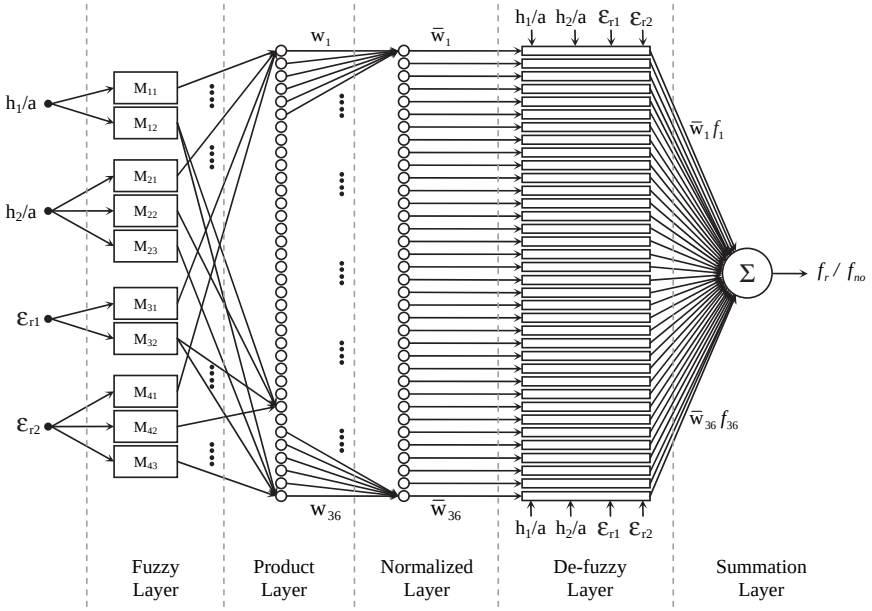

Figure 2. Architecture of ANFIS.

The accuracy of a properly trained ANFIS depends on the accuracy and the effective representation of the data used for its training. A good collection of the training data, i.e., data which is well-distributed, sufficient, and accurately simulated, is the basic requirement to obtain an accurate model. There are two types of data generators for antenna applications. These data generators are

the measurement and simulation. The selection of a data generator depends on the application and the availability of the data generator. The training and test data sets used in this paper have been obtained from the Fortran program [21] developed by using the results of spectral-domain moment method described in Section 2. 1000 data sets are used to train the ANFIS. Training data sets are in the range of $0 < h_1/a < 0.3$, $0 < h_2/a < 0.2$, $1 < \varepsilon_{r1} < 11$, and $1 < \varepsilon_{r2} < 11$. 177 data sets, which are completely different from training data sets, are used to test the ANFIS. The input and output data sets are scaled between 0 and 1 before training.

In the design of ANFISs, it is very important to determine the types and the parameters of membership functions (MFs). The number of the MFs for the input variables h_1/a, h_2/a, ε_{r1}, and ε_{r2} are 2, 3, 2 and 3, respectively. The number of rules is then 36 ($2 \times 3 \times 2 \times 3 = 36$). The types of the input MFs for the input variables h_1/a, h_2/a, ε_{r1}, and ε_{r2} are the Gaussian, Generalized bell, Gaussian, and Generalized bell, respectively. The Gaussian and the Generalized bell MFs are specified by two and three parameters, respectively. Therefore, the ANFIS used in this work contains a total of 206 fitting parameters, of which 26 ($2 \times 2 + 3 \times 3 + 2 \times 2 + 3 \times 3 = 26$) are the premise parameters and 180 ($5 \times 36 = 180$) are the consequent parameters.

Each possible combination of inputs and their associated MFs is represented by a rule in the rule base of the ANFIS in the following equation:

1. if (h_1/a is M_{11}) and (h_2/a is M_{21}) and (ε_{r1} is M_{31}) and (ε_{r2} is M_{41}) then $R_1 = f_1\ (h_1/a,\ h_2/a,\ \varepsilon_{r1},\ \varepsilon_{r2})$
2. if (h_1/a is M_{11}) and (h_2/a is M_{21}) and (ε_{r1} is M_{31}) and (ε_{r2} is M_{42}) then $R_2 = f_2\ (h_1/a,\ h_2/a,\ \varepsilon_{r1},\ \varepsilon_{r2})$
3. if (h_1/a is M_{11}) and (h_2/a is M_{21}) and (ε_{r1} is M_{31}) and (ε_{r2} is M_{43}) then $R_3 = f_3\ (h_1/a,\ h_2/a,\ \varepsilon_{r1},\ \varepsilon_{r2})$
4. if (h_1/a is M_{11}) and (h_2/a is M_{21}) and (ε_{r1} is M_{32}) and (ε_{r2} is M_{41}) then $R_4 = f_4\ (h_1/a,\ h_2/a,\ \varepsilon_{r1},\ \varepsilon_{r2})$

$$\begin{array}{ccc} \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots \end{array} \tag{13}$$

35. if (h_1/a is M_{12}) and (h_2/a is M_{23}) and (ε_{r1} is M_{32}) and (ε_{r2} is M_{42}) then $R_{35} = f_{35}\ (h_1/a,\ h_2/a,\ \varepsilon_{r1},\ \varepsilon_{r2})$
36. if (h_1/a is M_{12}) and (h_2/a is M_{23}) and (ε_{r1} is M_{32}) and (ε_{r2} is M_{43}) then $R_{36} = f_{36}\ (h_1/a,\ h_2/a,\ \varepsilon_{r1},\ \varepsilon_{r2})$

where M_{ij}, R_k, and f_k represent the jth MF of the ith input, the output of the kth rule, and the kth output MF, respectively.

In this work, the input MFs are Gaussian and Generalized bell types and the output MFs are a linear type as given bellow:

$$\begin{aligned} M_{ij}(x) &= Gauss\,(x; c_{ij}, \sigma_{ij}) = e^{-\frac{1}{2}\left(\frac{x-c_{ij}}{\sigma_{ij}}\right)^2} \\ & \quad i = (1,3),\ j = (1,2),\ (x = h_1/a \text{ or } \varepsilon_{r1}) \\ M_{ij}(x) &= Gbell\,(x;\ a_{ij},\ b_{ij},\ c_{ij}) = \frac{1}{1 + \left|\frac{x-c_{ij}}{a_{ij}}\right|^{2b_{ij}}} \\ & \quad i = (2,4),\ j = (1,2,3),\ (x = h_2/a \text{ or } \varepsilon_{r2}) \\ R_k &= f_k\,(h_1/a,\ h_2/a,\ \varepsilon_{r1},\ \varepsilon_{r2}) \\ &= d_{k1}h_1/a + d_{k2}h_2/a + d_{k3}\varepsilon_{r1} + d_{k4}\varepsilon_{r2} + d_{k5} \quad k = (1, \ldots, 36) \end{aligned} \tag{14}$$

Here a_{ij}, b_{ij}, c_{ij}, σ_{ij} and d_k are the parameters that characterize the shapes of the MFs. The optimal values of these parameters are determined by training.

The output of the ANFIS is the weighted average of the individual rule outputs. The weighting factor of each rule, which is expressed as w_k, is calculated by evaluating the membership expressions in the antecedent of the rule. This is accomplished by first converting the input values to fuzzy membership values by utilizing the input MFs and then applying the "*and*" operator to these membership values. The "*and*" operator corresponds to the multiplication of input membership values. Hence, the weighting factors of the rules are calculated as follows:

$$\begin{aligned} w_1 &= M_{11}\,(h_1/a) \cdot M_{21}\,(h_2/a) \cdot M_{31}\,(\varepsilon_{r1}) \cdot M_{41}\,(\varepsilon_{r2}) \\ w_2 &= M_{11}\,(h_1/a) \cdot M_{21}\,(h_2/a) \cdot M_{31}\,(\varepsilon_{r1}) \cdot M_{42}\,(\varepsilon_{r2}) \\ w_3 &= M_{11}\,(h_1/a) \cdot M_{21}\,(h_2/a) \cdot M_{31}\,(\varepsilon_{r1}) \cdot M_{43}\,(\varepsilon_{r2}) \\ w_4 &= M_{11}\,(h_1/a) \cdot M_{21}\,(h_2/a) \cdot M_{32}\,(\varepsilon_{r1}) \cdot M_{41}\,(\varepsilon_{r2}) \\ &\vdots \\ &\vdots \\ w_{35} &= M_{12}\,(h_1/a) \cdot M_{23}\,(h_2/a) \cdot M_{32}\,(\varepsilon_{r1}) \cdot M_{42}\,(\varepsilon_{r2}) \\ w_{36} &= M_{12}\,(h_1/a) \cdot M_{23}\,(h_2/a) \cdot M_{32}\,(\varepsilon_{r1}) \cdot M_{43}\,(\varepsilon_{r2}) \end{aligned} \tag{15}$$

Once the weighting factors are obtained, the output of the ANFIS can be found by calculating the weighted average of the individual rule

outputs:

$$f_r/f_{no} = \frac{\sum_{k=1}^{36} w_k R_k}{\sum_{k=1}^{36} w_k} = \sum_{k=1}^{36} \bar{w}_k f_k \tag{16}$$

It is clear that the ANFIS architecture consists of five layers: fuzzy layer, product layer, normalized layer, de-fuzzy layer, and summation layer [27, 28]. In the fuzzy layer, crisp input values are converted to fuzzy values by the MFs. After, in product layer, "*and*" operation is performed between the fuzzy values by using production so as to calculate the firing strength of each rule. Then, the normalized firing strengths are calculated in the normalized layer. In the de-fuzzy layer, the output rules are constructed. Finally, each rule is weighted by own normalized firing strength and the output of the ANFIS is calculated by summing of all rule outputs in the summation layer.

The ANFIS makes use of an optimization algorithm to optimize the fuzzy system parameters of the first-order Sugeno fuzzy model. The parameter optimization is done in a way such that the error measure between the target and the actual output is minimized. It can be observed that there are two adaptive layers in the ANFIS architecture, namely the fuzzy layer and the de-fuzzy layer. During the learning process of the ANFIS, the premise parameters in the fuzzy layer and the consequent parameters in the de-fuzzy layer are tuned until the desired response of the FIS is achieved.

In this paper, six different optimization algorithms, which are described briefly below, are used to determine the optimum values of the design parameters and adapt the FIS.

3.1. Optimization Algorithms

3.1.1. Hybrid Learning (HL) Algorithm

The HL algorithm [27, 28], which combines the least-squares (LSQ) method and the gradient descent (GD) algorithm, is commonly used to train and adapt the FIS. This algorithm converges much faster since it reduces the dimension of the search space of the GD algorithm.

From the architecture of the ANFIS, it is clear that the output can be expressed as:

$$O = \Gamma(\mathbf{u}, s) \tag{17}$$

where $\mathbf{u}$ is the vector of input variables, s is the total parameter set which combines premise parameters s_p and consequent parameters s_c, and Γ is the overall function implemented by the FIS. There will exist a function Υ such that the composite of $\Upsilon \circ \Gamma$ will be linear in s_c as follows:

$$\Upsilon(O) = \Upsilon \circ \Gamma(\mathbf{u}, s) \tag{18}$$

where Υ is the identity function. Hence, substituting values of the premise parameters s_p and the training data set P into Eq. (18), a matrix equation can be obtained as:

$$\mathbf{A}\mathbf{x} = \mathbf{y} \tag{19}$$

where $\mathbf{x}$ is the unknown parameter in s_c. Let $|s_c| = n$ and $|P| = m$. Then the dimensions of $\mathbf{A}$, $\mathbf{x}$ and $\mathbf{y}$ are $m \times n$, $n \times 1$, and $m \times 1$, respectively. As the number of training data m is usually greater than the number of linear parameters n, a LSQ method is used to find $\mathbf{x}$. On the other hand, the error measure for the jth $(1 \leq j \leq m)$ training data can be defined as the sum of squared errors:

$$E_j = \sum_{i=1} (T_{i,j} - O_{i,j})^2 \tag{20}$$

where $T_{i,j}$ is the ith component of the jth target output vector and $O_{i,j}$ is the ith component of the actual output vector produced by the jth input vector. So, the overall error measure is equal to:

$$E = \sum_{j=1}^{m} E_j \tag{21}$$

and the derivative of the overall error measure E with respect to the any premise parameter β $(\beta \in s_p)$ will be calculated as:

$$\frac{\partial E}{\partial \beta} = \sum_{j=1}^{m} \frac{\partial E_j}{\partial \beta} \tag{22}$$

The update formula for the any premise parameter β is accordingly to be:

$$\Delta\beta = -\eta \frac{\partial E}{\partial \beta} \tag{23}$$

with

$$\eta = \frac{\tau}{\sqrt{\sum_{\beta} \left(\frac{\partial E}{\partial \beta}\right)^2}} \tag{24}$$

where τ is a predetermined positive number and η is the step size, which can be changed to vary the speed of convergence. Now, the combination of the GD and the LSQ methods to update the parameters of the ANFIS is possible. As each epoch of the HL algorithm involves a forward pass and a backward pass in the ANFIS, the output of the whole system will be a linear combination of the consequent parameters. Thus, the output is written as in Eq. (16). Based on Eq. (16), the node outputs go forward till de-fuzzy layer and consequent parameters can be identified by the LSQ method, according to the calculation of Eq. (19) in the forward pass. On the other hand, the error rates of each node output propagate from the output end toward fuzzy layer, and the premise parameters are updated by the GD method using Eq. (23) in the backward pass. The merit of HL algorithm is that it can efficiently obtain the optimal premise parameters and consequent parameters values in the optimization process.

3.1.2. Least-Squares (LSQ) Optimization Algorithm

The LSQ optimization algorithm [39–41] is a mathematical optimization technique which, when given a series of observation data, attempts to find a function which closely approximates the data. It attempts to minimize the sum of the squares of the ordinate differences (called residuals) between the points generated by the function and the corresponding points in the data.

Although the LSQ methods for linear models are the most widely used techniques for fitting a set of observation data, occasionally it is appropriate to assume that the data are related through a model with nonlinear parameters. Nonlinear models can be divided into two types, which are referred to as intrinsically linear and intrinsically nonlinear models. Through appropriate transformations of its input-output variables and fitting parameters, an intrinsically linear model can be expressed in the standard form of a linear model. Thus standard LSQ method is applied to approximate the optimal parameters effectively. In the least-squares problem, the output of a system is generally given by the linearly parameterized expression:

$$y = f_1(\mathbf{u})\, x_1 + f_2(\mathbf{u})\, x_2 + \cdots + f_n(\mathbf{u})\, x_n \tag{25}$$

where $\mathbf{u} = [u_1, u_2, \cdots, u_p]^T$ is the input vector of the system, $f_1, f_2, \cdots, f_n$ are known functions of $\mathbf{u}$, and $x_1, x_2, \cdots, x_n$ are unknown parameters to be estimated.

For estimating the unknown parameters x_i, some experiments should be performed to obtain a training data set includes data pairs as $\{(\mathbf{u}_i; y_i), i = 1, \cdots, m\}$; they represent the desired input-output pairs

of the target system. Using Eq. (25), a set of m linear equations is expressed as:

$$\begin{cases} f_1(\mathbf{u}_1)\,x_1 + f_2(\mathbf{u}_1)\,x_2 + \cdots + f_n(\mathbf{u}_1)\,x_n &= y_1 \\ f_1(\mathbf{u}_2)\,x_1 + f_2(\mathbf{u}_2)\,x_2 + \cdots + f_n(\mathbf{u}_2)\,x_n &= y_2 \\ \vdots & \vdots \quad \vdots \\ f_1(\mathbf{u}_m)\,x_1 + f_2(\mathbf{u}_m)\,x_2 + \cdots + f_n(\mathbf{u}_m)\,x_n &= y_m \end{cases} \tag{26}$$

Using matrix notation, Eq. (26) can be rewritten as:

$$\mathbf{A}\mathbf{x} = \mathbf{y} \tag{27}$$

with

$$\mathbf{A} = \begin{bmatrix} f_1(\mathbf{u}_1) & \cdots & f_n(\mathbf{u}_1) \\ \vdots & \vdots & \vdots \\ f_1(\mathbf{u}_m) & \cdots & f_n(\mathbf{u}_m) \end{bmatrix}, \quad \mathbf{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}, \quad \mathbf{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_m \end{bmatrix} \tag{28}$$

where $\mathbf{A}$ is an $m \times n$ design matrix, $\mathbf{x}$ is an $n \times 1$ unknown parameter vector, and $\mathbf{y}$ is an $m \times 1$ output vector.

For determining as uniquely the unknown vector $\mathbf{x}$, it is necessary that $m \geq n$. If $\mathbf{A}$ is square ($m = n$) and nonsingular, it can be rewritten as:

$$\mathbf{x} = \mathbf{A}^{-1}\mathbf{y} \tag{29}$$

Generally, m is greater than n (there are more data pairs than parameters) and an exact solution satisfying all the m equations is not always possible since the data might be contaminated by noise. Thus, Eq. (27) should be modified by incorporating an error vector $\mathbf{e}$ (represents random noise or modeling error):

$$\mathbf{A}\mathbf{x} + \mathbf{e} = \mathbf{y} \tag{30}$$

Instead of finding the exact solution, a solution which minimizes the sum of squared error is searched:

$$E(\mathbf{x}) = \sum_{i=1}^{m} \mathbf{e}^2 = \mathbf{e}^T\mathbf{e} = (\mathbf{y} - \mathbf{A}\mathbf{x})^T (\mathbf{y} - \mathbf{A}\mathbf{x}) \tag{31}$$

where $\mathbf{e} = (\mathbf{y} - \mathbf{A}\mathbf{x})$ is the error vector for specific $\mathbf{x}$.

In this work, *Levenberg-Marquardt method*, which combines the best features of Gauss-Newton technique and the steepest-descent

method, is used for the calculation of the next step ($\mathbf{x}_{n+1}$) in the least-squares optimization:

$$\mathbf{x}_{n+1} = \mathbf{x}_n - \frac{1}{2}\left(\mathbf{J}^T\mathbf{J} + \vartheta\mathbf{I}\right)^{-1}\mathbf{g} \tag{32}$$

where $\mathbf{J}^T\mathbf{J} = \mathbf{H}$ is the Hessian matrix which consists of the second partial derivatives of $E(\mathbf{x})$, $\mathbf{g}$ is gradient of $E(\mathbf{x})$, ϑ is some nonnegative value, and $\mathbf{I}$ is the identity matrix. Depending on ϑ, the algorithm transits smoothly between Gauss-Newton method and steepest-descent method.

3.1.3. Nelder-Mead (NM) Optimization Algorithm

The NM "simplex" optimization algorithm [42, 43] is an enormously popular direct search method for multidimensional unconstrained minimization. The method uses the concept of a simplex, which is a geometrical figure consisting, in n dimension, $(n+1)$ vertices $(x_0, x_1, \ldots, x_n)$, and approximately finds a locally optimal solution to a problem with n variables when the cost (objective) function varies smoothly. The algorithm is applied several times with a different simplex for the beginning in order to explore new regions of the study domain and to find the global minimum of the cost function.

If x_0 is accepted as initial starting point, then other n points x_i are generated according to the relation $x_i = x_0 + \xi\ell_i$, where ℓ_i are n unit vectors, and ξ is a constant which is typically equal to 1. The initial simplex is moved, expanded, contracted and shrunk through a sequence of elementary geometric transformations. These transformations are called as reflection, expansion, contraction, and shrinkage (or multi-contraction), respectively. Through these transformations, the simplex can improve itself and come closer and closer to a local optimum point sequentially. After each transformation, the current worst vertex is replaced by a better one. Trial moves are generated according to the following basic operations:

At the beginning of the algorithm, one moves only the point of the simplex, where the objective function is worst, and one generates another point image of the worst point. This operation is the reflection. The reflection point x_r is calculated by:

$$x_r = (1+\rho)\,\bar{x} - \rho x_n \tag{33}$$

where $\bar{x} = \sum_{i=0}^{n-1} \frac{x_i}{n}$ is the centroid of the n best points (all vertices except for x_n) and ρ is the reflection coefficient.

If the reflected point is better than all other points, the method expands the simplex in this direction. This operation is the expansion. The expansion point x_e is calculated by

$$x_e = (1 - \chi)\,\bar{x} + \chi x_r \tag{34}$$

where χ is the expansion coefficient. If x_e is not better than all other points and at least better then the worst, the algorithm performs again the reflection with the new worst point.

When the worst point is at least as good as the reflected point, the contraction is implemented as:

$$x_c = (1 - \varphi)\,\bar{x} + \varphi x_n \tag{35}$$

where φ is the contraction coefficient. In this operation, the simplex adapts itself to the function landscape and finally surrounds the optimum.

If the worst point is better than the contracted point, the shrinkage is performed as:

$$v_i = x_0 + \psi\,(x_i - x_0)\,, \qquad i = 1, \ldots, n \tag{36}$$

where ψ is the shrinkage coefficient. Here, the (unordered) vertices of the simplex at the next iteration consist of $x_0, v_1, \ldots, v_n$.

At each step it is checked that the generated point is not outside the allowed reduced solution space. The algorithm stops when the difference between the best point and the worst point of the simplex is smaller than a certain value. Another criterion for the end of the algorithm is the number of function evaluations in order to avoid excessive calculation times.

3.1.4. Genetic Algorithm (GA)

The GA [44, 45] is a derivative-free stochastic optimization method based loosely on the concepts of natural selection and evolutionary processes. It is also is a parallel, robust, and probabilistic search technique that is simple and easily implemented without gradient calculation, compared with the conventional gradient-based search procedure.

A basic GA consists of six components. These are a random number generator, a fitness evaluation unit and genetic operators for reproduction, crossover, mutation, and selection operations.

The initial population required at the start of the algorithm is a set of number strings generated by the random number generator. After the initialization, the fitness value of each individual in the population

is calculated. The fitness of each individual in the population is rated correspondingly by its efficiency to finding the required solution.

Later, the termination criterion is tested. This step is the main for a loop which continues testing successive generations until the termination criterion is satisfied.

Then, a subpopulation is chosen for reproduction to closest adherence to the desired solution and the crossover is performed. The genetic chromosomes of the selected subpopulation are paired and their genetic material is combined to produce a third chromosome. Two parents possessing a desirable genetic sequence have a high likelihood of producing offspring with similar traits. The number of crossover operations is governed by a crossover rate.

The next operator is the mutation. In mutation, the resulting offspring is subjected to random mutation. The success of genetic algorithms stems largely from this crucial step which allows rapid adaptation to changing conditions in the population. The number of mutation operations is determined by a mutation rate.

After the mutation, best solutions are kept for the next generation by means of the selection. In the selection process, the solutions in the population are evaluated and a fitness value of each solution is calculated that how closely the solution approximates the desired solution.

Finally, fitness level of the new population is measured to determine whether to terminate the algorithm or to continue with successive generations until the termination criterion is satisfied.

3.1.5. Differential Evolution Algorithm (DEA)

The DEA [46–48] is a simple, population based, and direct-search algorithm for optimizing globally the multi-modal functions. Like the GA, it employs the crossover and mutation operators, and selection mechanism. An important difference between the GA and the DEA is that the GA relies on the crossover operator which provides the exchange of information among the solutions to construct the better solutions, while the DEA relies on the mutation operation as the main operator. The DEA also employs a non uniform crossover which takes child vector parameters from one parent more often than from others. The non uniform crossover operator efficiently shuffles information about successful combinations. This enables the search to focus on the most promising area of the solution space. The DEA introduces a novel mutation operation which is not only simple, but also significantly effective. The mutation operation is based on the differences of randomly sampled pairs of solutions in the population. Apart from being simple and able globally optimizing multi-modal

search spaces, some other features of the DEA are being: fast, easy to use, very easily adaptable for integer and discrete optimization, and quite effective in nonlinear constraint optimization including penalty functions.

The version of DEA used in this work is known as DE/best/1/exp or "DE1" [46–48]. Classic DEA begins by initializing a population of Np and D-dimensional vectors with parameter values which are distributed with random uniformity between the pre-specified lower initial parameter bound $x_{j,low}$ and the upper initial parameter bound $x_{j,high}$:

$$\begin{aligned} x_{j,i,t} &= x_{j,low} + rand\,(0,1)\,(x_{j,high} - x_{j,low})\,, \\ & \quad j = (1,\,2,\,\ldots\,,\,D), \quad i = (1,\,2,\,\ldots\,,\,Np)\,, \quad t = 0 \end{aligned} \tag{37}$$

The subscript t is the generation index, while j and i are the parameter and population indices, respectively. Hence, $x_{j,i,t}$ is the jth parameter of the ith population vector in generation t.

To generate a trial solution, DEA first mutates a best solution vector from the current population by adding to the scaled difference of two other vectors from the current population:

$$\mathbf{v}_{i,t} = \mathbf{x}_{best,t} + Sc\,(\mathbf{x}_{r_1,t} - \mathbf{x}_{r_2,t})\,, \qquad r_1,\,r_2 \in \{1,\,2,\,\ldots\,,\,Np\} \tag{38}$$

where $\mathbf{v}_{i,t}$ is the mutant vector. Vector indices r_1 and r_2 are randomly selected except that all are distinct and different from the population index i (i.e., $r_1 \neq r_2 \neq i$). The mutation scale factor Sc is a positive real number that is typically less than 1.0.

Next, one or more parameter values of this mutant vector $\mathbf{v}_{i,t}$ are exponentially crossed with those belonging to the ith population vector $\mathbf{x}_{i,t}$ (the target vector). The result is the trial vector $\mathbf{u}_{i,t}$:

$$u_{j,i,t} = \begin{cases} v_{j,i,t} & \text{if } rand\ (0,1) \leq Cr \text{ or } j = j_{rand} \\ x_{j,i,t} & \text{otherwise} \end{cases} \;;\quad j_{rand} \in \{1,2,\ldots,D\} \tag{39}$$

The crossover constant ($0.0 \leq Cr \leq 1.0$) controls the fraction of parameters that the mutant vector contributes to the trial vector. In addition, the trial vector always inherits the mutant vector parameter with the randomly chosen index j_{rand} to ensure that the trial vector differs by at least one parameter from the vector with which it will be compared (i.e., the target vector, $\mathbf{x}_{i,t}$).

As the final operation, the selection is used to create better solutions. If the trial vector's function value is less than or equal to that of the target vector, the trial vector replaces the target vector

in the next generation. Otherwise, the target vector remains in the population for at least one more generation:

$$\mathbf{x}_{i,t+1} = \begin{cases} \mathbf{u}_{i,t} & \text{if } f(\mathbf{u}_{i,t}) \leq f(\mathbf{x}_{i,t}) \\ \mathbf{x}_{i,t} & \text{otherwise} \end{cases} \tag{40}$$

These processes are repeated until a termination criterion is attained or a predetermined generation number is reached.

3.1.6. Particle Swarm Optimization (PSO) Algorithm

The PSO algorithm [49, 50] is an evolutionary computation technique inspired by social behavior of bird flocking or fish schooling. Similar to GA, PSO is an optimization tool based on population, and the system is initialized with a population of random solutions and can search for optima by the updating of generations. In the algorithm each member of the population is called a "particle", and each particle "flies" around in the multidimensional search space with a velocity, which is constantly updated by the particle's own experience and the experience of the particle's neighbors or the experience of the whole swarm. Compared to other evolutionary techniques, the advantages of PSO are that it is easy to implement and there are only few parameters to adjust.

Let x and v denote a particle positions and its corresponding velocity in a multidimensional search space, respectively. The best previous position of a particle is recorded and denoted by p^b. The best particle among all the particles in the group is denoted by p^g. At last, the modified velocity and position of each particle can be calculated as:

$$v_{t+1} = \varsigma \left(v_t + c_1 r_1 \left(p_t^b - x_t \right) + c_2 r_2 \left(p_t^g - x_t \right) \right), \tag{41a}$$

$$x_{t+1} = x_t + v_{t+1} \tag{41b}$$

where ς is the constriction factor, c_1 and c_2 are the positive constant coefficients, r_1 and r_2 are the uniformly distributed random numbers in [0, 1], v_t is the current velocity of the particle at iteration t, x_t is the current position of the particle at iteration t, v_{t+1} is the modified velocity, and x_{t+1} is the position of the particle at iteration $t+1$. The constriction factor ς which can prevent explosion is calculated by [50]:

$$\varsigma = \frac{2}{\left| 2 - (c_1 + c_2) - \sqrt{(c_1 + c_2)^2 - 4(c_1 + c_2)} \right|} \tag{42}$$

The algorithm is terminated if the minimum error criterion is attained or a predetermined number of iterations is reached. In this paper, Clerc's Constricted PSO [50] was used.

3.2. Parameter Values of Optimization Algorithms

Selection of parameters of the optimization algorithm mostly depends on experience besides the type of problem at hand. The parameter values of the optimization algorithms are: for HL, the initial step size = 0.06, the step size decrease rate = 0.9, and the step size increase rate = 1.1; for LSQ, the seed number = 0; for NM, the reflection coefficient = 1, the expansion coefficient = 2, the contraction coefficient = 0.5, and the shrinkage coefficient = 0.5; for GA, the population size = 350, the number of elite individuals = 20, the crossover rate = 0.8, the mutation rate = 0.05, and the migration rate = 0.2; for DEA, the population size = 2060, the mutation scale factor = 0.001, the crossover rate = 0.5, and the strategy = 1; for PSO algorithm, the particles size = 30, the maximum velocity divisor = 4, the acceleration constants # 1 and # 2 = 2.1, and the constriction coefficient = 0.64174.

4. RESULTS AND CONCLUSIONS

The resonant frequency of the circular MSA with a dielectric cover is successfully computed by using ANFIS models. The optimum design parameters of the ANFIS are determined with the use of HL, LSQ, NM, GA, DEA, and PSO. The training and test RMS errors of six ANFIS models are given in Table 1. These RMS error values clearly show that the ANFIS models can be used in calculating the resonant frequency of the circular MSA with a dielectric cover. When the performances of ANFIS models are compared with each other, the best results for training and testing are obtained from the ANFIS model optimized by the LSQ optimization algorithm, as shown in Table 1. The final shapes of the MFs are illustrated in Figure 3 for the ANFIS model optimized by the LSQ. For brevity, the final shapes of the MFs of other ANFIS models are not given.

The test results of ANFIS optimized by the LSQ are compared with the results of the moment-method full-wave analysis [20] and the CAD formula [21] in Figures 4–6. It can be clearly seen from Figures 4–6 that the results of ANFIS are in very good agreement with the results of moment-method and the CAD formula.

In order to make a further validation, the results of ANFIS optimized by the LSQ are compared with the results of Galerkin's

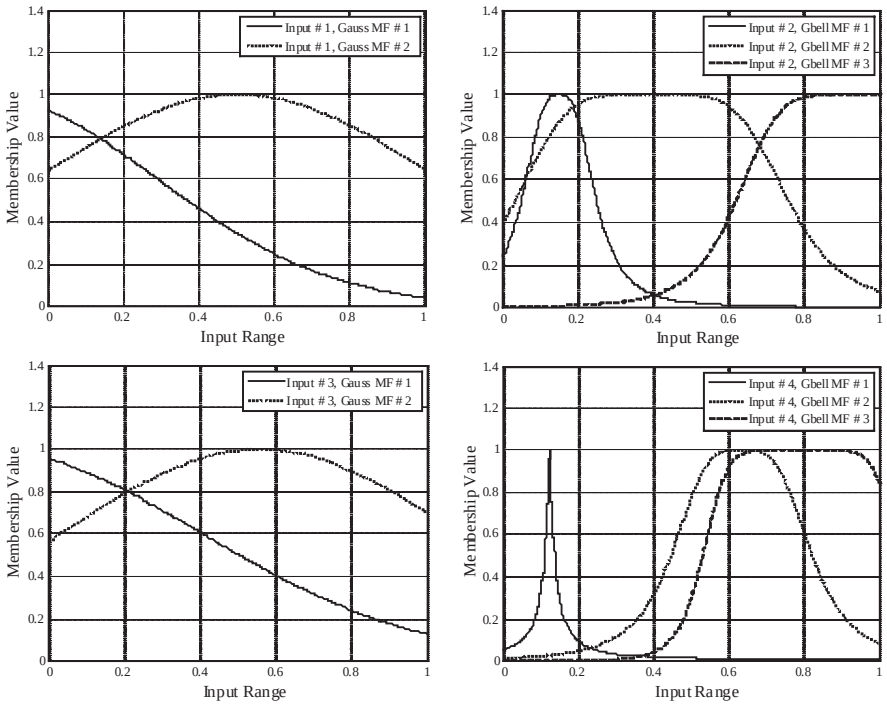

Figure 3. Shapes of the MFs of input variables (Input #1 = h_1/a, Input # 2 = h_2/a, Input # 3 = ε_{r1}, and Input # 4 = ε_{r2}) for the ANFIS model optimized by the LSQ.

method in the Hankel transform domain [22] in Figure 7. It is seen from Figure 7 that the results of ANFIS are in good agreement with the results of Galerkin's method in the Hankel transform domain.

The resonant frequency test result of ANFIS optimized by the LSQ is also compared with the measured result [22] in Table 2. For comparison, the results obtained by using the other theories [21–24] are also given in Table 2. It is evident that the result of ANFIS agrees better with the measured result than the results of [21–24].

In this paper, only the resonant frequency of the TM_{11} mode is calculated by using the ANFIS because this circular microstrip patch mode is widely used in MSA applications. However, the ANFIS can be easily adapted to compute the resonant frequencies of higher-order modes of practical interest if the data sets for these modes are available. It must also be emphasized that the proposed method is not limited to the resonant frequency calculation of circular MSA

Table 1. Training and test errors of ANFIS models optimized by different algorithms.

Optimization Algorithms	RMS Training Errors	RMS Test Errors
HL	0.0004860	0.0013710
LSQ	0.0003898	0.0002597
NM	0.0004809	0.0005790
GA	0.0004868	0.0009843
DEA	0.0004867	0.0009824
PSO	0.0004899	0.0009589

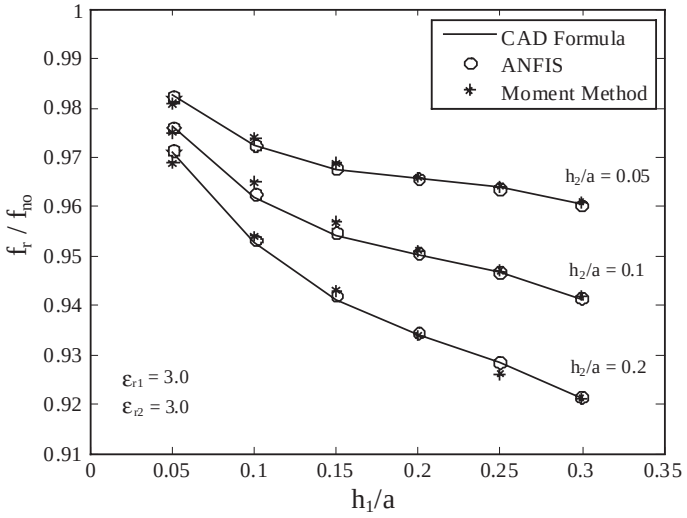

Figure 4. Comparison of the resonant frequency results of the ANFIS optimized by the LSQ, the moment-method [20], and the CAD formula [21] for $\varepsilon_{r1} = 3.0$ and $\varepsilon_{r2} = 3.0$.

with a dielectric cover. This method can easily be applied to other antenna and microwave circuit problems. Accurate, fast, and reliable ANFIS models can be developed from measured/simulated antenna data. Once developed, these ANFIS models can be used in place of computationally intensive numerical models to speed up antenna design.

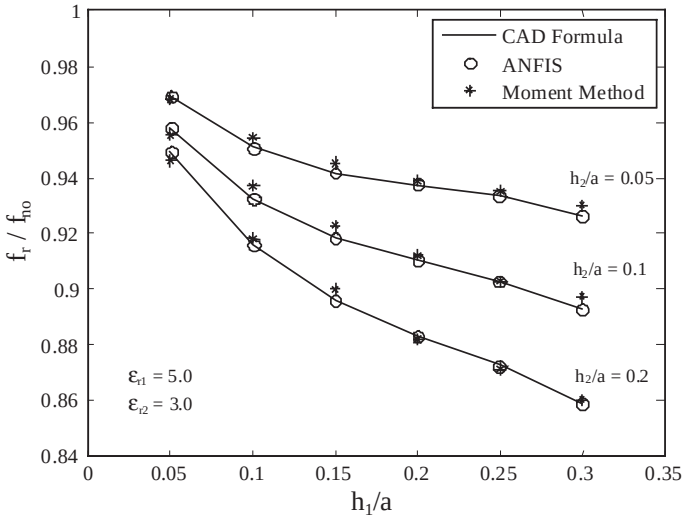

Figure 5. Comparison of the resonant frequency results of the ANFIS optimized by the LSQ, the moment-method [20], and the CAD formula [21] for $\varepsilon_{r1} = 5.0$ and $\varepsilon_{r2} = 3.0$.

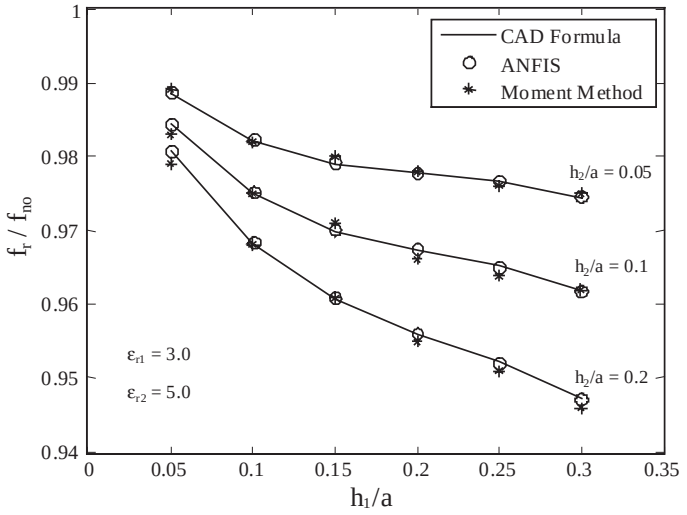

Figure 6. Comparison of the resonant frequency results of the ANFIS optimized by the LSQ, the moment-method [20], and the CAD formula [21] for $\varepsilon_{r1} = 3.0$ and $\varepsilon_{r2} = 5.0$.

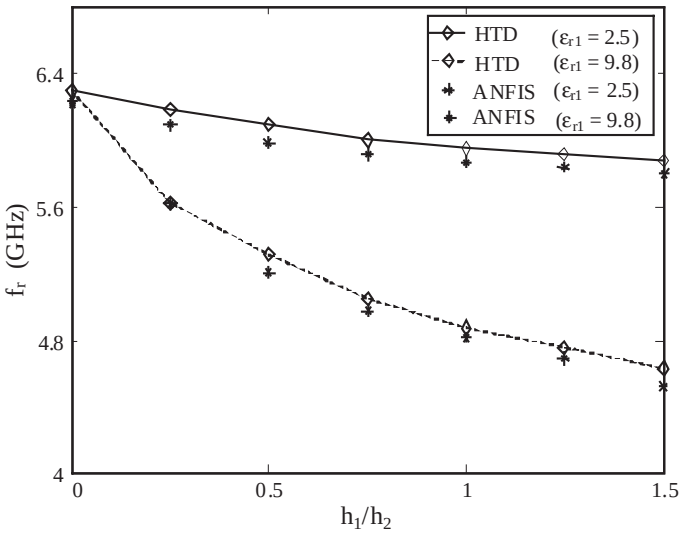

Figure 7. Comparison of the resonant frequency results of the ANFIS optimized by the LSQ and Galerkin's method in the Hankel transform domain (HTD) [22] for $h_2 = 1.5875\,\text{mm}$, $a/h_2 = 5$, and $\varepsilon_{r2} = 2.5$.

Table 2. Comparison of the computed resonant frequency results of the ANFIS optimized by the LSQ, Ensemble [22], and [21–24] with the measured result for $a = 5.89\,\text{mm}$, $h_2 = 0.49\,\text{mm}$, $h_1/h_2 = 3.11$, $\varepsilon_{r1} = 2.5515$, and $\varepsilon_{r2} = 2.43$.

Resonant Frequency (GHz)						
	Calculated					
Measured [22]	Ensemble [22]	Losada et al. [22]	Bouttout et al [24]	Guha and Siddiqui [23]	Lee and Fan [21]	Present ANFIS Model
8.57	8.79	8.71	8.80	8.60	8.61	8.58

As a consequence, an alternative method based on ANFIS is used to compute accurately the resonant frequency of the circular MSA with a dielectric cover. The HL, LSQ, NM, GA, DEA and PSO are used to optimize the parameters of ANFIS. The best result is obtained from the ANFIS optimized by LSQ algorithm. It was shown that the ANFIS results are in very good agreement with the results available in the literature. The ANFIS has the advantages of easy implementation and good learning ability. The high-speed real-time computation feature of the ANFIS recommends its use in antenna CAD programs.

REFERENCES

1. Bahl, I. J. and P. Bhartia, *Microstrip Antennas*, Artech House, Dedham, MA, 1980.
2. James, J. R., P. S. Hall, and C. Wood, *Microstrip Antennas-Theory and Design*, Peter Peregrisnus Ltd., London, 1981.
3. Gupta, K. C. and A. Benalla (eds.), *Microstrip Antenna Design*, Artech House, Canton, MA, 1988.
4. James, J. R. and P. S. Hall, *Handbook of Microstrip Antennas*, IEE Electromagnetic Wave Series, Vols. 1 and 2, No. 28, Peter Peregrinus Ltd., London, 1989.
5. Bhartia, P., K. V. S. Rao, and R. S. Tomar (eds.), *Millimeter-Wave Microstrip and Printed Circuit Antennas*, Artech House, Canton, MA, 1991.
6. Hirasawa, K. and M. Haneishi, *Analysis, Design, and Measurement of Small and Low-Profile Antennas*, Artech House, Canton, MA, 1992.
7. Pozar, D. M. and D. H. Schaubert (eds.), *Microstrip Antennas—The Analysis and Design of Microstrip Antennas and Arrays*, IEEE Press, New York, 1995.
8. Zurcher, J. F. and F. E. Gardiol, *Broadband Patch Antennas*, Artech House, Norwood, MA, 1995.
9. Sainati, R. A., *CAD of Microstrip Antennas for Wireless Applications*, Artech House, Norwood, MA, 1996.
10. Lee, K. F. and W. Chen, *Advances in Microstrip and Printed Antennas*, John Wiley and Sons, 1997.
11. Garg, R., P. Bhartia, I. Bahl, and A. Ittipiboon, *Microstrip Antenna Design Handbook*, Artech House, Canton, MA, 2001.
12. Cooray, F. R. and J. S. Kot, "Analysis of radiation from a cylindrical-rectangular microstrip patch antenna loaded with a superstrate and an air gap, using the electric surface current model," *Progress In Electromagnetics Research*, PIER 67, 135–152, 2007.
13. Sim, C. Y. D., J. S. Row, and Y. Y. Liou, "Experimental studies of a shorted triangular microstrip antenna embedded with dual V-shaped slots," *J. Electromagnetic Waves and Applications*, Vol. 21, No. 1, 15–24, 2007.
14. Pirhadi, A., M. Hakkak, and F. Keshmiri, "Using electromagnetic bandgap superstrate to enhance the bandwidth of probe-fed microstrip antenna," *Progress In Electromagnetics Research*, PIER 61, 215–230, 2006.

15. Bao, X. L. and M. J. Ammann, "Comparison of several novel annular-ring microstrip patch antennas for circular polarization," *J. Electromagnetic Waves and Applications*, Vol. 20, No. 11, 1427–1438, 2006.
16. Qian, Z. H., R. S. Chen, K. W. Leung, and H. W. Yang, "FDTD analysis of microstrip patch antenna covered by plasma sheath," *Progress In Electromagnetics Research*, PIER 52, 173–183, 2005.
17. Liu, S. F., S. D. Liu, and B. R. Guan, "A novel wideband high-temperature superconducting microstrip antenna," *J. Electromagnetic Waves and Applications*, Vol. 19, No. 15, 2073–2079, 2005.
18. Luk, K. M., W. Y. Tam, and C. L. Yip, "Analysis of circular microstrip antennas with superstrate," *IEE Proc. Microwave, Antennas Propagat.*, Vol. 136, No. 3, 261–262, 1989.
19. Verma, A. K. and Z. Rostamy, "Modified Wolff model for determination of resonance frequency of dielectric covered circular microstrip patch antenna," *Electronics Lett.*, Vol. 27, No. 24, 2234–2236, 1991.
20. Fan, Z. and K. F. Lee, "Input impedance of circular-disk microstrip antennas with a dielectric cover," *Microwave Opt. Technol. Lett.*, Vol. 5, No. 13, 701–704, 1992.
21. Lee, K. F. and Z. Fan, "CAD formulas for resonant frequencies of TM_{11} mode of circular patch antenna with or without superstrate," *Microwave Opt. Technol. Lett.*, Vol. 7, No. 12, 270–273, 1994.
22. Losada, V., R. R. Boix, and M. Horno, "Resonant modes of circular microstrip patches in multilayered substrates," *IEEE Trans. Microwave Theory Tech.*, Vol. 47, No. 4, 488–497, 1999.
23. Guha, D. and J. Y. Siddiqui, "Resonant frequency of circular microstrip antenna covered with dielectric superstrate," *IEEE Trans. Antennas Propagat.*, Vol. 51, No. 7, 1649–1652, 2003.
24. Bouttout, F., F. Benabdelaziz, and A. Khellaf, "Closed-form Hankel transforms for circular disk basis modes involving Chebyshev polynomials and edge condition," *Electronics Lett.*, Vol. 36, No. 10, 866–867, 2000.
25. Guha, D., "Resonant frequency of circular microstrip antennas with and without air gaps," *IEEE Trans. Antennas Propagat.*, Vol. 49, 55–59, 2001.
26. Bernhard, J. T. and C. J. Tousignant, "Resonant frequencies of rectangular microstrip antennas with flush and spaced dielectric superstrates," *IEEE Trans. Antennas Propagat.*, Vol. 47, 302–308, 1999.

27. Jang, J.-S. R., "ANFIS: Adaptive-network-based fuzzy inference system," *IEEE Trans. Systems, Man, and Cybernetics*, Vol. 23, 665–685, 1993.
28. Jang, J.-S. R., C. T. Sun, and E. Mizutani, *Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence*, Prentice-Hall, Upper Saddle River, NJ, 1997.
29. Guney, K. and N. Sarikaya, "Adaptive neuro-fuzzy inference system for the input resistance computation of rectangular microstrip antennas with thin and thick substrates," *J. Electromagnetic Waves and Applications*, Vol. 18, No. 1, 23–39, 2004.
30. Guney, K. and N. Sarikaya, "Computation of resonant frequency for equilateral triangular microstrip antennas using the adaptive neuro-fuzzy inference system," *Int. J. RF and Microwave Computer-Aided Eng.*, Vol. 14, 134–143, 2004.
31. Guney, K. and N. Sarikaya, "Input resistance calculation for circular microstrip antennas using adaptive neuro-fuzzy inference system," *Int. J. Infrared and Millimeter Waves*, Vol. 25, 703–716, 2004.
32. Guney, K. and N. Sarikaya, "Adaptive neuro-fuzzy inference system for computing the resonant frequency of circular microstrip antenna," *The Applied Computational Electromagnetics Society (ACES) J.*, Vol. 19, No. 3, 188–197, 2004.
33. Guney, K. and N. Sarikaya, "Adaptive neuro-fuzzy inference system for the computation of the bandwidth of electrically thin and thick rectangular microstrip antennas," *Electrical Eng.*, Vol. 88, 201–210, 2006.
34. Guney, K. and N. Sarikaya, "Adaptive neuro-fuzzy inference system for computing the physical dimensions of electrically thin and thick rectangular microstrip antennas," *Int. J. Infrared and Millimeter Waves*, Vol. 27, No. 2, 219–233, 2006.
35. Guney, K. and N. Sarikaya, "A hybrid method based on combining artificial neural network and fuzzy inference system for simultaneous computation of resonant frequencies of rectangular, circular, and triangular microstrip antennas," *IEEE Trans. Antennas Propagat.*, Vol. 55, No. 3, 659–668, 2007.
36. Turkmen, I. and K. Guney, "Cheap joint probabilistic data association with adaptive neuro-fuzzy inference system state filter for tracking multiple targets in cluttered environment," *AEU — Int. J. Electronics and Commun.*, Vol. 58, No. 5, 349–357, 2004.
37. Turkmen, I. and K. Guney, "Tabu search tracker with adaptive neuro-fuzzy inference system for multiple target tracking,"

Progress In Electromagnetics Research, PIER 65, 169–185, 2006.

38. Daldaban, F., N. Ustkoyuncu, and K. Guney, "Phase inductance estimation for switched reluctance motor using adaptive neuro-fuzzy inference system," *Energy Conversion and Management*, Vol. 47, No. 5, 485–493, 2006.
39. Levenberg, K., "A method for the solution of certain nonlinear problems in least-squares," *Quart. Appl. Math. II*, 164–168, 1944.
40. Marquardt, D. W., "An algorithm for least-squares estimation of nonlinear parameters," *SIAM J. Appl. Math.*, Vol. 11, 431–441, 1963.
41. Dennis, J. E., "Nonlinear least-squares," *State of the Art in Numerical Analysis*, 269–312, Academic Press, 1977.
42. Spendley, W., G. R. Hext, and F. R. Himsworth, "Sequential application of simplex designs in optimization and evolutionary operation," *Technometrics*, Vol. 4, 441–461, 1962.
43. Nelder, J. A. and R. Mead, "A simplex method for function minimization," *Computer J.*, Vol. 7, 308–313, 1965.
44. Holland, J., *Adaptation in Natural and Artificial Systems*, University of Michigan Press, MI, 1975.
45. Goldberg, D. E., *Genetic Algorithms in Search, Optimization and Machine Learning*, Reading, Addison-Wesley, MA, 1989.
46. Price, K. V., "Differential evolution: a fast and simple numerical optimizer," *IEEE North American Fuzzy Info. Process Conf.*, 524–527, Berkeley, CA, 1996.
47. Storn, R. M. and K. V. Price, "Differential evolution: a simple and efficient heuristic for global optimization over continuous spaces," *J. Global Opt.*, Vol. 11, 341–359, 1997.
48. Price, K. V., R. M. Storn, and J. Lampinen, *Differential Evolution: A Practical Approach to Global Optimization*, Springer, Berlin, 2005.
49. Kennedy, J. and R. Eberhart, "Particle swarm optimization," *Int. Conf. Neural Networks*, Perth, Australia, 1942–1948, 1995.
50. Clerc, M. and J. Kennedy, "The particle swarm: explosion, stability, and convergence in a multi-dimensional complex space," *IEEE Trans. Evol. Comput.*, Vol. 6, 58–73, 2002.
51. Araki, K. and T. Itoh, "Hankel transform domain analysis of open circuit microstrip radiating structure," *IEEE Trans. Antennas Propagat.*, Vol. 29, 84–89, 1981.
52. Pozar, D. M., "Input impedance and mutual coupling of rectangular microstrip antennas," *IEEE Trans. Antennas Propagat.*, Vol. 30, 1191–1196, 1982.