ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

*(An ISO 3297: 2007 Certified Organization)*

Vol. 3, Issue 3, March 2014

# Comparison of Soft Computing Techniques for the Design of Microstrip Patch Antenna: A Review Paper

Vishal Davara[1], A. B. Upadhyay[2]

M.E., Department of Electronics & Communication, L.D.College of Engineering, Gujarat, India[1]
Assistance Professor, Department of Electronics & Communication, L.D.College of Engineering, Gujarat, India[2]

**ABSTRACT**: This paper describes different computing techniques for the design of Microstrip patch antenna. The performance of antenna is greatly influenced by the choice of technique used to design antenna. In this paper three different soft computing techniques are presented and compared. The most recently used techniques Artificial Neural Networks (ANNs), Fuzzy Logic (FL) and Adaptive Neuro-Fuzzy Inference System (ANFIS) are described. Antenna design is simulated for all three techniques and result is compared with each other. ANFIS is the most efficient technique to use. As far as number of training data set is concern ANFIS requires small data set and hence ANFIS provides optimization too.

**KEYWORDS**: ANN, ANFIS, FL,MSA (Microstrip Patch Antenna).

## I. INTRODUCTION

In the last decade, the use of various soft computing techniques has increased for design and optimization of various antennas. Soft computing technique differs from hard computing in that, unlike hard computing, it is tolerant of imprecision, uncertainty, approximation and on the basis of partial truth. The principle of soft computing is: Exploit the tolerance of imprecision, uncertainty, approximation and partial truth to achieve the tractability, robustness and low solution cost.

The principal constituents of Soft Computing (SC) are:
- Artificial Neural Networks (ANNs)
- Fuzzy Logic (FL)
- Evolutionary Computation (EC)
- Machine Learning (ML)
- Probabilistic Reasoning (PR)
- Genetic Algorithm (GA)
- Adaptive Neuro-Fuzzy Inference System (ANFIS).

The principal constituent methodologies in Soft Computing (SC) are complementary rather than competitive. Soft computing may be viewed as a foundation component of conceptual intelligence. In many cases a problem can be solved most effectively by using FL, ANNs, GA and PR in combination rather than competitive. A good example of a particular effective combination is what has come to be known as “Neuro-Fuzzy systems.” Such a system is widely used in consumer products ranging from air conditioners and washing machine to photocopiers. This paper presents most widely used three techniques.1) Artificial Neural Networks, 2) Fuzzy Inference System and 3) Adaptive Neuro Fuzzy Inference System.

## II. ARTIFICIAL NEURAL NETWORKS

### *A. Neural computing*

Artificial Neural Networks (Anns) is an important processing paradigm that is inspired from the biological nervous system, such as brain. It is composed highly inter connected processing elements called as neurons. Artificial

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

*(An ISO 3297: 2007 Certified Organization)*

**Vol. 3, Issue 3, March 2014**

Neural Networks (Anns), like people learn from example. A neural network is an artificial representation of the human brain. It tries to simulate its learning process. The term artificial means that neural networks are implemented in computer programs that are able to handle the large number of necessary calculations during the learning process. An ANN is configured for a specific application, such as pattern recognition or data classification, through a learning process. Learning in biological systems involves adjustments to the synaptic connections that exist between the neurons. This is also true for anns [3].

*B. ANNs Architecture*

The basic architecture consists of three types of neuron layers: input, hidden, and output. ANN structure has two basic components, (1) the processing elements and (2) the interconnection between them. The processing elements are called neurons and the connections between the neurons are known as links or synapses, as shown in fig-1. Every link has the corresponding weight associated with it. Each neuron receives stimulus from other neurons connected to it, process the information and produce an output. Neurons that receive the stimuli or input from external environment are known as input neurons, while neurons whose outputs are given to the external environment are known as output neurons. Neurons that receive the stimuli or input from the other neurons and whose outputs are stimuli for other neurons in the networks are known as hidden neurons [2].

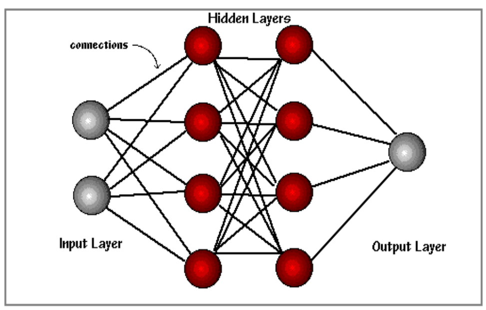


Fig-1 Neural Network Architecture [2].

*C. Ann versus Conventional Modelling*

Digital Computers:
Deductive Reasoning. We apply known rules to input data to produce output.
Computation is centralized, synchronous, and serial.
Memory is packeted, literally stored, and location addressable.
Not fault tolerant. One transistor goes and it no longer works.
Exact.
Static connectivity.
Applicable if well-defined rules with precise input data.

Neural Networks:
Inductive Reasoning. Given input and output data (training examples), we construct the rules.
Computation is collective, asynchronous, and parallel.
Memory is distributed, internalized, short term and content addressable.
Fault tolerant, redundancy, and sharing of responsibilities.
Inexact.
Dynamic connectivity.
Applicable if rules are unknown or complicated, or if data are noisy or partial.

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 3, March 2014

*D. Network Size and Layer*

The number of hidden neurons depends on the degree of non-linearity of function and dimensionality of inputs and outputs. Highly nonlinear component needs more neurons and smoother needs fewer neurons. However, we do not specify the size of the networks. User can employ either experience or a trial and error process to judge the number of hidden neurons. Generally, one or two hidden layer is commonly used for antenna applications. But there is currently no theoretical reason to use neural networks more than two hidden layer. With no hidden layer is capable of representing linear separable functions or decisions [3].

Rule-of-thumb methods:

• The number of hidden neurons should be 2/3 the size of the input layer, plus the size of the output layer.

• The number of hidden neurons should be less than twice the size of the input layer.

• For a three layer network with n input and m output neurons, the hidden layer would have (n $\times$ m) neurons.

## III. FUZZY INFERENCE SYSTEM

Fuzzy inference is the process of formulating the mapping from a given input to an output using fuzzy logic. The mapping then provides a basis from which decisions can be made, or patterns discerned. The process of fuzzy inference involves membership functions, logical operations, and if-then Rules. You can implement two types of fuzzy inference systems in the toolbox: Mamdani-type and Sugeno-type.

Fuzzy inference systems have been successfully applied in fields such as automatic control, data classification, decision analysis, expert systems, and computer vision. Fuzzy inference systems are associated with a number of names, such as fuzzy-rule-based systems, fuzzy expert systems, fuzzy modelling, fuzzy associative memory, fuzzy logic controllers, and simply (and ambiguously) fuzzy systems [2].

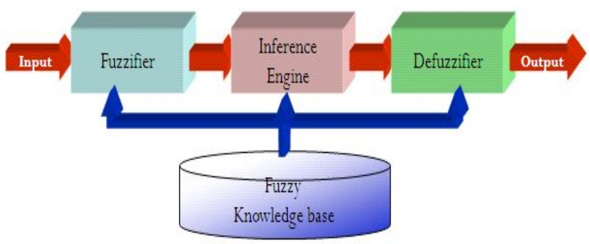


Fig-2 Fuzzy Inference System [2].

Step1. Fuzzify Inputs

The first step is to take the inputs and determine the degree to which they belong to each of the appropriate fuzzy sets via membership functions. In Fuzzy Logic Toolbox software, the input is always a crisp numerical value limited to the universe of discourse of the input variable (in this case the interval between 0 and 10) and the output is a fuzzy degree of membership in the qualifying linguistic set (always the interval between 0 and 1). Fuzzification of the input amounts to either a table lookup or a function evaluation.

Step2. Apply Fuzzy Operator

After the inputs are fuzzified, the designer should know the degree to which each part of the antecedent is satisfied for each rule. If the antecedent of a given rule has more than one part, the fuzzy operator is applied to obtain one number that represents the result of the antecedent for that rule. This number is then applied to the output function. The input to the fuzzy operator is two or more membership values from fuzzified input variables. The output is a single truth value.

Step3. Apply Implication Method

Before applying the implication method, the designer must determine the rule's weight. Every rule has a weight (a number between 0 and 1), which is applied to the number given by the antecedent. Generally, this weight is 1

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

*(An ISO 3297: 2007 Certified Organization)*

Vol. 3, Issue 3, March 2014

(as it is for this example) and thus has no effect at all on the implication process. From time to time you may want to weight one rule relative to the others by changing its weight value to something other than 1.

After proper weighting has been assigned to each rule, the implication method is implemented. A consequent is a fuzzy set represented by a membership function, which weights appropriately the linguistic characteristics that are attributed to it. The consequent is reshaped using a function associated with the antecedent (a single number). The input for the implication process is a single number given by the antecedent, and the output is a fuzzy set. Implication is implemented for each rule. Two built-in methods are supported, and they are the same functions that are used by the AND method: *min* (minimum), which truncates the output fuzzy set; and *prod* (product), which scales the output fuzzy set.

Step4. Aggregate All Outputs

Because decisions are based on the testing of all of the rules in a FIS, the rules must be combined in some manner in order to make a decision. Aggregation is the process by which the fuzzy sets that represent the outputs of each rule are combined into a single fuzzy set. Aggregation only occurs once for each output variable, just prior to the fifth and final step, defuzzification. The input of the aggregation process is the list of truncated output functions returned by the implication process for each rule. The output of the aggregation process is one fuzzy set for each output variable. As long as the aggregation method is commutative (which it always should be), then the order in which the rules are executed is unimportant.

Step5. Defuzzify

The input for the defuzzification process is a fuzzy set (the aggregate output fuzzy set) and the output is a single number. As much as fuzziness helps the rule evaluation during the intermediate steps, the final desired output for each variable is generally a single number. However, the aggregate of a fuzzy set encompasses a range of output values, and so must be defuzzified in order to resolve a single output value from the set.

Perhaps the most popular defuzzification method is the centroid calculation, which returns the centre of area under the curve. There are five built-in methods supported: centroid, bisector, middle of maximum (the average of the maximum value of the output set), largest of maximum, and smallest of maximum.

The steps of fuzzy reasoning (inference operations upon fuzzy if-then rules) performed by fuzzy inference systems are:
Compare the input variables with the membership functions on the premise part to obtain the membership values (or compatibility measures) of each linguistic label. (This step is often called Fuzzification).
Combine (through a specific T-norm operator, usually multiplication or min.) the membership values on the premise part to get firing strength (weight)of each rule.
Generate the qualified consequent (either fuzzy or crisp) of each rule depending on the firing strength.

## IV. ADAPTIVE NEURO-FUZZY INFERENCE SYSTEMS

The basic idea behind these Neuro-Adaptive learning techniques is very simple. These techniques provide a method for the fuzzy modelling procedure to learn information about a data set, in order to compute the membership function parameters that best allow the associated fuzzy inference system to track the given input/output data. This learning method works similarly to that of neural networks.

The word ANFIS derives its name from adaptive Neuro-Fuzzy inference system. Using a given input/output data set, the toolbox function ANFIS constructs a fuzzy inference system (FIS) whose membership function parameters are tuned (adjusted) using either a back propagation algorithm alone, or in combination with a least squares type of method. This allows your fuzzy systems to learn from the data they are modelling. Fuzzy systems are more favourable in that their behaviour can be explained based on fuzzy rules and thus their performance can be adjusted by tuning the rules. But since, in general, fuzzy systems are restricted to the fields where the expert knowledge is available and the number of input variable is small. To overcome the problem of knowledge acquisition, neural network are extended to automatically extract fuzzy rules from numerical data. ANFIS use the neural network to optimize certain parameters of an ordinary fuzzy system, or to pre-process data and extract fuzzy rules from the data [7].

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 3, March 2014

### *A. ANFIS Objectives*

It integrates the best features of Fuzzy Systems and Neural Networks:

From Fuzzy System: Representation of prior knowledge into a network topology to reduce the optimization
From NN: Adaptation of back propagation to structured network to provide the automatic fuzzy logic.
Therefore, the ANFIS provide the smoothness due to the fuzzy logic and adaptive learning due to neural network back propagation. However ANFIS has the strong computational complexity [2].

### *B. Importance of ANFIS compared to ANN & FS*

Advantages of the ANFIS as compared to the neural network:

Faster convergence rate as compared to the feed-forward neural networks.
Small training data set gives the more accurate results as compared to the neural network.
It can be possible by varying the number of the membership functions.

Advantages of the ANFIS as compared to the fuzzy system:

Fuzzy systems are more complex by increasing the number of input variables or/and by increasing the size of data set.
ANFIS creates its own representation or organization of the given information during the training time.
It can generate its own fuzzy if-then rules depending on the data set given at the training time [1].

### *C. Constraint of ANFIS*

ANFIS is much more complex than the fuzzy inference systems discussed so far, and is not available for all of the fuzzy inference system options. Specifically, ANFIS only supports Sugeno-type systems, and these must have the following properties:

• Be first or zeroth order Sugeno-type systems.
• It has a single output, obtained using weighted average defuzzification. All output membership functions must be the same type and either is linear or constant.
• Have no rule sharing. Different rules cannot share the same output membership function, namely the number of output membership functions must be equal to the number of rules.
• Have unity weight for each rule.

An error occurs if the FIS structure does not comply with these constraints. Moreover, ANFIS cannot accept all the customization options that basic fuzzy inference allows. That is, you cannot make your own membership functions and defuzzification functions; you must use the ones provided.

## V. SIMULATION

During training and testing processes different methodsseemed to provide different grades of accuracy. Table 1shows an account of the efficiencyobserved for each ofthese method whiletraining and table 2 shows an account ofthe efficiency observed for each of these method while testing. The testing and training phase was performed usingthe bestavailable unconventional tools used in the patch antenna design arena.

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 3, March 2014

| Sr. No. | *Training method* | **Methods and training efficiency** | |
|---|---|---|---|
| | | ***Efficiency Circle*** | ***Efficiency Square*** |
| 1 | Back propagation network | 99.80 | 99.867 |
| 2 | Conjugate gradient | 99.91 | 99.892 |
| 3 | Quasi Newton | 99.996 | 99.994 |
| 4 | ANFIS | 99.987 | 99.985 |

Table 1 Accuracy of various methods during training phase

| Sr. No. | *Training method* | **Methods and testing efficiency** | |
|---|---|---|---|
| | | ***Efficiency Circle*** | ***Efficiency square*** |
| 1 | Back propagation network | 97.03 | 98.607 |
| 2 | Conjugate gradient | 98.249 | 98.467 |
| 3 | Quasi Newton | 99.95 | 99.969 |
| 4 | ANFIS | 99.982 | 99.984 |

Table 2 Accuracy of various methods during testing phase

Here Table 1 and 2 shows the efficiency achieved while modelling the square and circular antenna using different soft computing techniques during training and testing phase respectively. Table 3 and 4 shows difference between frequency calculated for the different values of antenna parameters and frequency obtained from the models developed using different techniques like Back propagation network, Conjugate gradient, Quasi Newton and Adaptive neuro fuzzy inference system.

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 3, March 2014

| Sr. No. | | | | | | | Frequency calculated | Frequency calculated using neural network methods | | | Frequency calculated using neurofuzzy method |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | ***fdesign*** | **s/λ** | ***h*** | ***εr*** | ***W*** | ***L*** | ***fcalc*** | ***fBPN*** | ***fconjugate gradient*** | ***fquasi newton*** | ***fANFIS*** |
| 1 | 3017 | 0.025 | 0.24 | 2 | 0.3 | 0.2727 | 3017 | 3142 | 2995 | 3014 | 3017 |
| 2 | 2955 | 0.035 | 0.26 | 4 | 0.4 | 0.6076 | 2955 | 2937 | 2921 | 2960 | 2956 |
| 3 | 2613 | 0.04 | 0.27 | 5 | 0.45 | 0.3214 | 2613 | 2542 | 2592 | 2614 | 2614 |
| 4 | 4675 | 0.07 | 0.3 | 2.5 | 0.65 | 0.3823 | 4675 | 4807 | 4513 | 4677 | 465 |
| 5 | 4065 | 0.075 | 0.32 | 3.5 | 0.123 | 0.0683 | 4065 | 4109 | 3943 | 4066 | 4064 |
| 6 | 4274 | 0.08 | 0.29 | 4.5 | 0.234 | 0.1462 | 4274 | 4351 | 4150 | 4276 | 4273 |
| 7 | 6551 | 0.07 | 0.24 | 2.5 | 0.65 | 0.5909 | 6551 | 6910 | 6559 | 6549 | 6552 |

Table 3 Frequency values for Circular case

| Sr. No. | | | | | | | Frequency calculated | Frequency calculated using neural network methods | | | Frequency calculated using neurofuzzy method |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | ***fdesign*** | **s/λ** | ***h*** | ***εr*** | ***W*** | ***L*** | ***fcalc*** | ***fBPN*** | ***fconjugate gradient*** | ***fquasi newton*** | ***fANFIS*** |
| 1 | 3017 | 0.025 | 0.24 | 2 | 0.3 | 0.2727 | 2686 | 2693 | 2705 | 2688 | 2686 |
| 2 | 2955 | 0.035 | 0.26 | 4 | 0.4 | 0.6076 | 2657 | 2669 | 2644 | 2659 | 2657 |
| 3 | 2613 | 0.04 | 0.27 | 5 | 0.45 | 0.3214 | 2653 | 2653 | 2333 | 2357 | 2654 |
| 4 | 4675 | 0.07 | 0.3 | 2.5 | 0.65 | 0.3823 | 4165 | 4165 | 4087 | 4166 | 4165 |
| 5 | 4065 | 0.075 | 0.32 | 3.5 | 0.123 | 0.0683 | 3680 | 3680 | 3809 | 3679 | 3679 |
| 6 | 4274 | 0.08 | 0.29 | 4.5 | 0.234 | 0.1462 | 3867 | 3867 | 3959 | 3863 | 3868 |
| 7 | 6551 | 0.07 | 0.24 | 2.5 | 0.65 | 0.5909 | 5827 | 5827 | 5863 | 5830 | 5826 |

Table 4 Frequency values for Rectangular case

## VI. RESULTS AND CONCLUSION

In this paper neural network and neuro-fuzzy systems are used as unconventional tools for square and rectangular patch antenna design. Our work is intended to save the time of antenna designers as well as solving the design problems with very good accuracy. This tool requires no prior knowledge of the antennas for implementation. ANFIS and Quasi Newton methods can help in developing faster and accurate methods which can be used in computer aided microstrip antenna design. Thus ANFIS is the most efficient and effective method to use.

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 3, March 2014

## REFERENCES

[1] K. Guney, N. Sarikaya, Concurrent Neuro-fuzzy systems for resonant frequency computation of rectangular, circular, and triangular microstrip antennas, Progress In Electromagnetics Research, PIER 84, 253277, 2008.
[2] V.K. Lamba, Neuro-Fuzzy Systems, University Science Press, 2011.
[3] C. A. Balanis, Modern Antenna Handbook, John Wiley and Sons, Chapter 33, 2008.
[4] Abbas Ali Heidari, AbolfazlDadgarnia, Design and Optimization of a Circularly Polarized Microstrip Antenna for GPS Applications Using ANFIS and GA, IEEE Transactions, 2011.
[5] Jyh-Singh, Roger, Jang, Adaptive-Network-Based Fuzzy Inference System, IEEE Transactions on systems, man, and cybernetics, vol. 23, no. 3, May/June 1993.
[6] Zadeh, Lofti A., Fuzzy Logic, Neural Networks, and Soft Computing, Communication of the ACM, vol. 37, no. 3, March 1994.
[7]J. S. R. Jang, C. T. Sun, E. Mizutani, Neuro-Fuzzy and Soft Computing, Pearson Education, 2004.
[8] Ganesh Bharadwaj, PrasannaVadana. D, “Resonant Frequency Computation of E- Plane Mutually Coupled Rectangular and Circular Patch Antennas Using Unconventional Tools”, IEEE conference 2010.
[9] I. Ray, M. Khan, D. Mondal, and A.K. Bhattacharjee, “Effect on resonant frequency for E-plane mutually coupled microstrip antennas”, progress in electromagnetics research letters, vol. 3, pp. 133-140, 2008.