J. Intell. Syst. **21** (2012), 121 – 142
DOI 10.1515/jisys-2012-0006

Recurrent Fuzzy-Neural MIMO Channel Modeling

Kandarpa Kumar Sarma and Abhijit Mitra

Abstract. Fuzzy systems and artificial neural networks (ANN), as important components of soft-computation, can be applied together to model uncertainty. A composite block of the fuzzy system and the ANN shares a mutually beneficial association resulting in enhanced performance with smaller networks. It makes them suitable for application with time-varying multi-input multi-output (MIMO) channel modeling enabling such a system to track minute variations in propagation conditions. Here we propose a fuzzy neural system (FNS) using a fuzzy time delay fully recurrent neural network (FTDFRNN) that has the capability to tackle time-varying inputs in fuzzified form and is used to model MIMO channels. The inference engine is constituted by novel FTDFRNN blocks which determine the decision boundaries and tracks in-phase and quadrature components of input signals encompassing stochastic behavior of the MIMO channel. The system shows significant improvement in performance compared to statistical and ANN approaches in terms of faster processing time, lower bit error rate (BER) margins and better precision while carrying out symbol recovery of transmitted data through severely faded MIMO channels.

Keywords. MIMO, Estimation, Artificial Neural Network, Recurrent Neural Network, Self Organizing Map, Optimization, Fuzzy, Fuzzy-Neural.

2010 Mathematics Subject Classification. 94A40.

1 Introduction

Although the multiple-input multiple-output (MIMO) wireless technology is one of the options likely to meet the demands of ever expanding mobile communication networks, its channel estimation continues to be a challenging area. A very common form of uncertainty and stochastic behavior is observed in MIMO wireless communication due to interference and correlation among channel coefficients. There are several statistical and other methods that have provided satisfactory performance while modeling the MIMO wireless channel [6, 8]. MIMO systems and related aspects including channel modeling are treated extensively in [3, 12]. Among soft-computational approaches, a few recent works have considered certain aspects beyond the training-testing realm. A work dealing with the application of artificial neural network (ANN) in form of multi layer perceptron (MLP) and its temporal variants for static and slowly varying MIMO channels has been reported in [15]. The performance constraints observed in this approach are overcome, to a large extent, by the use of a class of recurrent neural network (RNN) structures and have been appropriately configured to deal with time-varying MIMO channels [16]. These works lay stress on certain architectural challenges through which lower bit error rate (BER), better precision and faster processing time are obtained while modeling the stochastic nature of the MIMO channels. However, in certain practical cases like VOIP based using such wireless channels, the performance of RNN-based methods needs to be improved further especially in connection with the accuracy of the recovered content. In such situations, in order to enhance the system precision, the natural option that emerges as the suitable addition along with the RNN is either fuzzy system or the genetic algorithm (GA). The primary limitation associated with GA, however, is the underlying computational complexity and therefore the time constraint. Hence, the fuzzy based system integrated with RNN block turns out to be the only viable option to attain the desired precision performance. Further, fuzzy systems possess an integral ability to deal with uncertainty, discern minute variations, attach probabilistic linkages to finely varying changes [9, 13] and facilitate the formation of a mechanism for near error-free decision making using ANN. Also, the uniqueness associated with these soft-computational tools enables a smooth blending of fuzzy system with ANN in a supplementary-complementary arrangement which enhancement performances in terms of lower processing time and reduced system complexity, increase in precision, better adaptability and an unmatched capability to track and model random behavior of stochastic MIMO channels. The end result is a mechanism which shares a symbiotic association with each other that enhances respective and cumulative performances. The ANN provides process data extracted from non-parametric sources executing the task with model-free processing such that the fuzzy system uses the neural response to provide expert-level decision with the aim to achieve better overall performance. This is in addition to the stability of the set-up and the capability to process minutely varying contextual and relevant information due to the presence of feedback loops in the RNN [16] components encapsulated in the system with fuzzy-related modifications. In terms of implementation aspects, the advantage of such a composite architecture is a set-up which can make better discrimination between the significant and correlated coefficients of time-varying channels such that the correct class-wise grouping of the system improves and misclassification reduces with less processing time and lower design complexity. For its inherent capability to derive expert level knowledge and decision making [9, 13] fuzzy based systems have already been used in wireless communication in areas like equalization etc. A few important works are cited in [1, 4, 10, 11, 14, 17–19]. All of these works emphasize applications of fuzzy-based systems for wireless communication areas with nearly no stress on architectural challenges and spin-offs that can be derived using such systems with modifications. An attempt is made here to configure a fuzzy-based system using a time delay fully recurrent neural network (TDFRNN) with the ability to deal with time-varying inputs in fuzzified form to model MIMO channels. The fuzzified TDFRNN blocks are designed to generate a set of input conditioning norms and inference logic which captures finer variations in the signals transmitted by the MIMO set-up. The fuzzification process is automated using an ANN-aided approach while two distinct TDFRNN blocks generate the classification boundaries in terms of in-phase and quadrature components combined and optimized using a self organizing map (SOM) unit. The training process is accelerated by constantly changing fuzzy samples generated by dissimilar processing blocks. The results show significant improvement in processing time and accuracy as compared to ANN based approaches. The fuzzy systems, on an average, provide at least 5% improvement in accuracy as compared to the RNN-based estimation which has already been established to be a better alternative to statistical and ANN based approaches [16]. This is in addition to the processing time advantage that the fuzzy systems provide. The rest of the paper is organized as follows: Section 2 provides the details related to MIMO channel modeling using FNS. There are several subsections of Section 2 giving the related details of the system model proposed using a fuzzified time delay fully recurrent neural network (FTDFRNN) for capturing finite variations in time-varying signals propagating through a MIMO arrangement. The experimental details are included in Section 3. The content includes certain results derived from different subsections and stages. The description is concluded by Section 4.

2 The Proposal: Modeling MIMO Channels using Recurrent Fuzzy-Neural System

Fuzzy systems are rule driven tools that provide expert-level knowledge for decision making [9, 13]. ANNs, on the other hand, are adaptive, robust and non-parametric prediction techniques that demonstrate cognitive behavior [5]. Fuzzy systems alone, however, find it difficult to tackle uncertainty hence require support from other tools like ANN. Integration of a fuzzy system with ANN enables the former to acquire capabilities of implementing a framework of processing modalities like signal conditioning rules, inference norms etc., so that the composite systems acquire the ability of better learning, retentivity of knowledge and expert-level decision making with the ability to capture finite variations in the input. Hence, hybrid forms, like the fuzzy-neural system (FNS) or the neuro-fuzzy

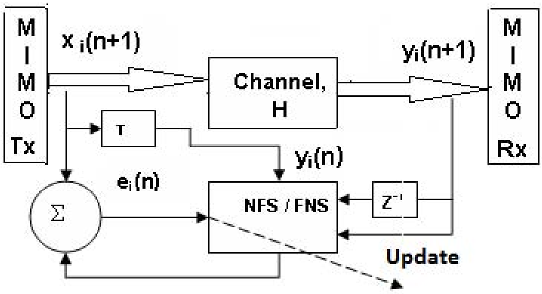


Figure 1. Fuzzy system based channel estimation.

system (NFS), are popular in a diverse range of applications. However, experimental results derived for VOIP based transmissions show that the FNS approach is better suited for time-varying MIMO channels and hence is adopted for modeling such a highly volatile propagation environment. The FNS is an implementation of a fuzzy system within the ANN architecture. In this approach, both numerical (measurement based) data and perception based information represented as fuzzy numbers are handled. Therefore, FNS captures more relevant content from an input, hence is better suited for real world situations [2]. The FNS allows automation of fuzzy rule generation and has the ability to perform combined learning of numerical data as well as expert-knowledge expressed as fuzzy if-then-else rules. Moreover, FNS has smaller networks and faster process times compared to ANNs and NFS [2], hence it is more suitable for applications like adaptive receiver design for high data rate mobile communication.

The FNS approach is explored here with appropriate modifications for making it compatible with MIMO channel modeling as depicted in Figure 1.

Using an orthogonal frequency division multiplexing (OFDM) signal for a 4×4 MIMO set-up, the input-output relation may be written as

$$x_i(k+1) = \big[s_i(k) + s_i(k-\tau)\big]\mathbf{H}(i,k) + v(k), \tag{1}$$

where $x_i(k+1)$ is the received signal at time $k+1$ with i taking values of 1 to 4 for a 4×4 channel set-up, $s_i(k)$ is the transmitted signal state at time k, $s_i(k-\tau)$ is the transmitted signal at time $k-\tau$, $v(k)$ is the background noise, τ is the delay associated with multipath fading and the channel coefficients for ith input and kth time constituting the channel matrix $\mathbf{H}(i,k)$. Expression (1) shows the contribution of direct and delayed versions of the signal components while propagating through the MIMO channel. In general, the received signal state at time $k+n$ can be expressed as

$$x_i(k+n) = F\big[s_i(k,\tau), s_i(k), \mathbf{H}(i,k)\big] + v(k). \tag{2}$$

The mapping function $F(\cdot)$ represents the transformation which the signal suffers during transmission through the MIMO channel. It relates present and delayed versions of the signal $\mathbf{s}$ along with the channel matrix $\mathbf{H}$ for formulating a generation mechanism of the received signal sequences which also includes considerable amount of co-channel interference (CCI). The proposed fuzzy-based system for modeling the MIMO channel is configured to track the time varying transformation which the signal undergoes while it propagates through the channel represented by a mapping process $F(\cdot)$. The mapping $F(\cdot)$ is dependent on $s_i(k,\tau)$, $s_i(k)$ and $\mathbf{H}(i,k)$ as represented by expression (2) which is different to that shown in equation (1). In another form, the expression of the received signal in a MIMO set-up may be expressed showing dependence of delayed outputs and other parameters as

$$\tilde{y}(n) = \mathrm{FM}_G\big[\tilde{y}(n-1), \ldots, \tilde{x}(n-1), \ldots, [\tilde{\mathbf{H}}], \ldots\big], \tag{3}$$

where FM_G is a fuzzy mapping process to be generated by an ANN-based process. The expression $\tilde{y}(n)$ becomes the target pattern which should be given to a soft-computational tool during training. The requirement is to design another ANN-based process such that its output is given by

$$\tilde{y_G}(n) = \mathrm{FM}_G\big[\tilde{y}(n-1), \ldots, \tilde{x}(n-1), [\tilde{\mathbf{H}}], [\tilde{W}], [\tilde{V}], \tilde{\theta}\big], \tag{4}$$

so that $\tilde{y}_G(n)$ tends to $\tilde{y}(n)$ in terms of a cost function. Here $[\tilde{W}]$, $[\tilde{V}]$ and $\tilde{\theta}$ are forward and backward connectionist weights and biases, respectively. Therefore, the training of a fuzzy process is related to the minimization of a cost function expressed as

$$\mathrm{CF} = \frac{1}{\mathrm{TN} \times \mathrm{VD}} \sum \sum d(y_{di}, y_{ai}), \tag{5}$$

where TN is the number of training samples, VD is the dimension of the samples, $d(\cdot)$ is a distance measure, y_{di} is the desired output and y_{ai} is the actual output. The channel matrix $\mathbf{H}$ represents the wireless medium through which the propagation takes place. This channel is considered to be a time variant system with an impulse response given by

$$h(t,\tau) = \sum_{g=0}^{N_p-1} \alpha_g(t,\tau) \exp\big(j(2\pi f_c \tau_g(t) + \phi(t,\tau))\big)\delta(\tau - \tau_g(t)), \tag{6}$$

where N_p is the number of multipath components, $\alpha_g(t,\tau)$ is the amplitude component and $\tau_g(t)$ is the excess delay component caused by the gth multipath component at time t and δ is the delta function. The inverse dynamics allows a definition

$$\big[\hat{\mathbf{H}}(i,k)\big] = G\big[x_i(k), x_i(k-\tau)\big] \tag{7}$$

and

$$\big[\hat{s}_i(k)\big] = G\big[x_i(k), x_i(k-\tau)\big]. \tag{8}$$

The received signal states $x_i(k)$ and $x_i(k-\tau)$ can be obtained from expressions (1) and (2). The mapping $G[\cdot]$ represents a trained soft-computational tool which can apply its learning to estimate the channel coefficients and recover the signal symbols. It signifies a modeling mechanism related to the MIMO channel from which the significant and CCI coefficients can be determined from the received and transmitted signal components. The inverse dynamics resembles a control process. Its outcome, denoted by $\mathbf{H}$, is regulated by factors like the received signal $x_i(k)$, the transmitted signal $s_i(k)$ and the pathdelay τ. It indicates that some a priori knowledge about the transmitted content is required which enables the system to learn the patterns and then use the learning for estimating the channel coefficients and recovering the symbol bits from the received content. The channel matrix $\mathbf{H}$ can be determined from the inverse dynamics $G[\cdot]$ obtained from the following sets of data:

$$\big[x_i(k), s_i(k), s_i(k-\tau)\big]. \tag{9}$$

The learning of the soft-computational network is oriented to achieve the following objectives:

$$\|e_1(i,k)\|^2 = \|\mathbf{H}(i,k) - \hat{\mathbf{H}}(i,k)\|, \tag{10}$$

$$\|e_2(i,k)\|^2 = \|s_i(k) - \hat{s}_i(k)\| \tag{11}$$

from which

$$\|e_1(i,k)\|^2 = \big\|\mathbf{H}(i,k) - G\big[x_i(k), x_i(k,\tau)\big]\big\|, \tag{12}$$

$$\|e_2(i,k)\|^2 = \big\|s_i(k) - G\big[x_i(k), x_i(k,\tau)\big]\big\|. \tag{13}$$

These considerations are followed while designing the MIMO channel modeling approaches described in the subsequent sections. The FNS structure deviates considerably from the traditional ANN and has the following attributes [13]: fuzzy inputs, weights, outputs and aggregation operation instead of summation as observed in ANNs. For the present work a FN MIMO channel estimator is formulated and is shown in Figure 2.

The MIMO channel matrix (MCM) is formed using a representation given as

$$\mathbf{H}_{i,j}(z) = \sum_{i,j}\{a_{i,j}z^{-j}\} \tag{14}$$

and the co-channel interference (CCI) responses are expressed as

$$C_{m,n}(z) = \sum_{m,n}\{c_{m,n}z^{-n}\}, \tag{15}$$

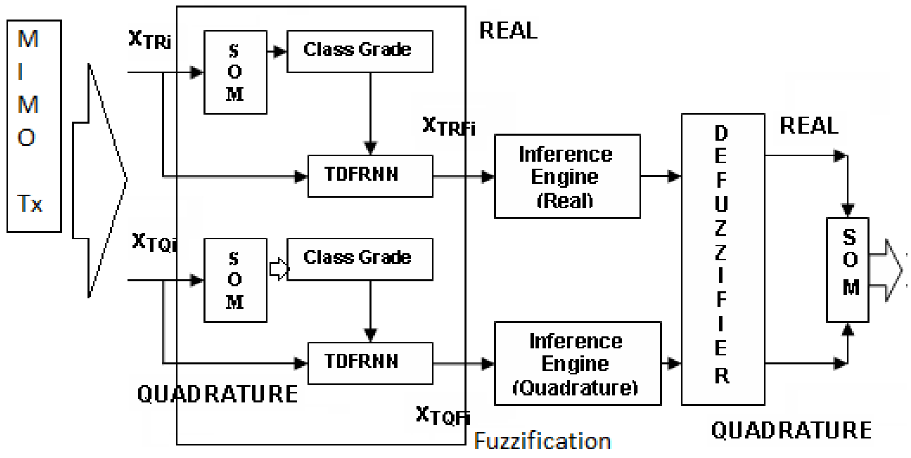


Figure 2. Fuzzy system based channel estimation.

so that the complete signal content at the receiver is expressed as

$$Y_R(z) = X_{i,j}\left[\sum_{i,j}\{a_{i,j}z^{-j}\} + \sum_{m,n}\{c_{m,n}z^{-n}\}\right] + N, \qquad (16)$$

where X is the OFDM matrix generated using the parameters given in Table 1. Before applying these symbols to the fuzzy system, they are all fuzzified.

A set of characteristics of the channel model developed using the Clarke–Gans formulation considers the parameters as given in Table 2.

2.1 Input Conditioning

The primary input to the system comes from a MIMO transmitter during training. The inputs in the in-phase and quadrature forms are fuzzified and membership grades assigned. The fuzzy inputs next go to the inference engine which decides upon the class decisions as per the assigned inference logic. The output is de-fuzzified and obtained from real and imaginary sections and combined. The outputs are trimmed using a self organizing map (SOM) optimizer.

The training samples are normalized and confined within a few linguistic steps (Table 3); see [13].

Sl. no.	Parameter	Specification
1	Baseband modulation	16-QAM, BPSK
2	FFT length	512
3	Number of carriers	128
4	Cyclic prefix	16

Table 1. Parameters used for generating the OFDM signal.

Sl. no.	Parameter	Value
1	Frequency, f_c	900 MHz
2	Mobile speed, V	3–100 kmph
3	Number of paths	16
4	Wavelength, λ	$(3 \times 10^8)/f_c$
5	Doppler shift, f_m	V/λ
6	Sampling freq., f_s	$10 \times f_m$
7	Number of samples, N	10^3 to 10^6
8	Number of retransmissions	10
9	Sampling Period, T_s	$1/f_s$
10	Antenna configuration	2×2, 4×4, 8×8

Table 2. Parameters used for simulating channel using Clarke–Gans model.

Sl. no.	State	Notation
1	Negative Large	NL
2	Negative Medium	NM
3	Negative Small	NS
4	Close to Zero	CZ
5	Positive Large	PL
6	Positive Medium	PM
7	Positive Small	PS

Table 3. Linguistic steps used to condition the inputs.

The inputs are also constrained by the following norms:

$$f_1(x) = \begin{cases} \text{NL}, & -0.99 \le x < -0.66, \\ \text{NM}, & -0.66 \le x < -0.33, \\ \text{NS}, & -0.33 \le x < 0, \\ \text{Z}, & x = 0, \\ \text{PL}, & 0.66 \le x < 0.99, \\ \text{PM}, & 0.33 \le x < 0.66, \\ \text{PS}, & 0 < x < 0.33. \end{cases} \tag{17}$$

2.2 Fuzzification

The inputs are divided into in-phase and quadrature components to allow the FNS to learn the individual signal segments separately. The fuzzy sets of the respective inputs are generated using a SOM to create clusters for each of the samples forming the input matrix. This clustering is used to train a MLP to act as an automated membership generator. The fuzzy sets of the respective inputs are generated following two ways. In the first method, a Bell-shaped membership function is used to generate the member-grades of each input. A one-to-one correspondence is established between input and fuzzy sets. This association is taken to train an MLP to generate the membership function. At the end of the training, the MLP becomes an automatic membership generator (Figure 3 (a)). The second method is to use a SOM to create clusters for each of the samples forming the input matrix. This clustering is used to train an MLP to act as an automated membership generator (Figure 3 (b)).

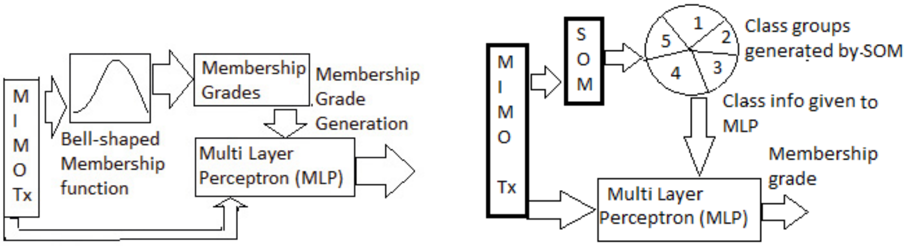
Let x_T be the input to the MLP (Figure 3) while y_{EMG} is the expected set of membership grades given by the Bell-shaped function. The output of the 3-layered MLP is given as

$$y_{\text{AMG}} = \sum_k f_k \left[\sum_i f_i \left[\sum_j f_j (x_{Tj} W_{jm} + b_j) W_{in} \right] \right], \tag{18}$$

where the $f(\cdot)$ are activation functions, $W[\cdot,\cdot]$ are connectionist weights and b are biases related to specific layers of the MLP. The instantaneous error is given as

$$E_p = \sum_k [y^p_{\text{EMG}_k} - y^p_{\text{AMG}_k}], \tag{19}$$

such that the weight adaptation can be continued till the goal is reached with



(a) Bell-shape function and trained MLP (b) SOM and trained MLP

Figure 3. Membership grade generation using.

gradient given as

$$\Delta W_{kj} = -\eta \frac{\delta E_p}{\delta W_{kj}} \tag{20}$$

for use in the training process. Here η is the learning rate.

In case of the SOM-MLP approach, the membership grades are generated as

$$y_{\mathrm{EMG}} = x_T[W_j], \tag{21}$$

where W_j are the random weights of the competitive layer of the SOM. The 'winners take all' competition starts [5] in this layer such that the winning neuron index, J, satisfies

$$y_J = \max_j \{y_{0j}^T, W_j\}. \tag{22}$$

For a 4×4 MIMO set-up, four training signals are transmitted with three different AWGN values viz. -3 dB, 1 dB and 3 dB for Gaussian, Rayleigh and Rician faded channels. The channel has h_{11}, h_{22}, h_{33} and h_{44} as primary direct channel impulse responses while $h_{12}, h_{21}, h_{31}, h_{41}, \ldots$ are cross-channel impulse responses between the specific transmitter and the receiver. The input training sequences x_1, x_2, x_3, x_4 are transmitted in data blocks with each block constituted by N_I information and N_T training symbols. For different time slots the training sequences provide

$$h_{i1} = \frac{Y_{Ti,1}}{x}, \quad h_{i2} = \frac{Y_{Ti,2}}{x}, \quad h_{i3} = \frac{Y_{Ti,3}}{x}, \quad h_{14} = \frac{Y_{Ti,4}}{x}, \tag{23}$$

where $i = 1, 2, 3, 4$. Let the above knowledge be considered to be a priori. So if during training $[\tilde{h}_{11}, \tilde{h}_{12}, \ldots, \tilde{h}_{21}, \tilde{h}_{22}, \ldots, \tilde{h}_{31}, \tilde{h}_{32}, \tilde{h}_{33}, \ldots, \tilde{h}_{41}, \ldots, \tilde{h}_{44}]$ are the estimates of channel coefficients, the fuzzy inference can be developed from the risk functional $d(h_{ij}, \tilde{h}_{ij})$ where $d(\cdot)$ represents the distance measure between actual and expected responses. The fuzzified inputs form a fuzzy set $\tilde{A}$ given as

$$\tilde{A} = \{[x, \mu_A(x) \mid x \in X]\}, \tag{24}$$

where $\mu_A(x)$ is called the membership function or grade of membership and is generated by the fuzzification process as described in Section 2.1. The fuzzified inputs thus obtained are next passed on to the inference engine which is a multi-layered set-up designed by the following two approaches based on fuzzy multi layer perceptron (FMLP) and fuzzy time delay fully recurrent neural network (FTDFRNN).

Fuzzy Multi Layer Perceptron (FMLP) based Inference Engine

The FMLP is formed by multiple layers of fuzzy perceptrons (Figure 4 (a)). The inputs (X_{kj}), weights (W_j) and outputs are all fuzzified [7]. The output of such a set of perceptrons can be expressed as

$$Y_k = \sum_k S_{\beta,k}\left(\sum_{j=0}^{n} W_j X_{kj}\right), \tag{25}$$

where S_β is a sigmoid function for a certain steepness parameter β. The calculation covered by the summation sign in (25) is done using principles of fuzzy arithmetic. The output thus obtained is next passed on to the hidden and output layers such that the final response is expressed as

$$Y_m = \sum_m S_{\beta,m}\left\{\left(\sum_l S_{\beta,l}\left(\sum_k Y_k W_{kl}\right) W_{lm}\right)\right\}. \tag{26}$$

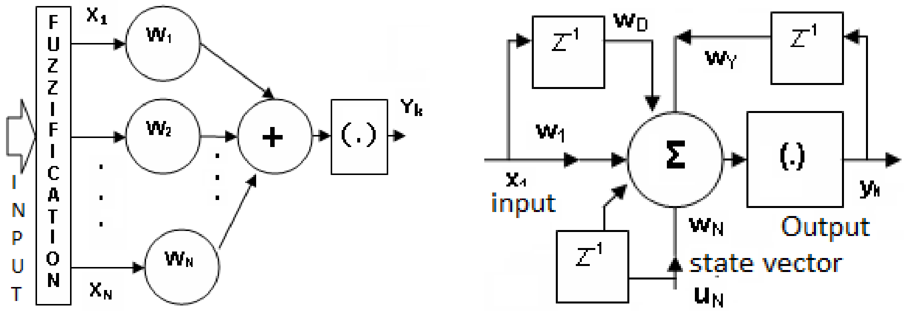
For separate real and imaginary inputs the respective outputs are combined so that the final response is found as

$$Y_{mF} = \sum_m S_{\beta,m}\left\{\sum_l \left(Y_R + jY_I\right)\right\}, \tag{27}$$

where

$$Y_R = \sum \left\{S_{\beta,l} \sum_k Y_{kR} W_{kl}\right\}, \tag{28}$$

$$Y_I = \sum \left\{S_{\beta,l} \sum_k Y_{kI} W_{kl}\right\}. \tag{29}$$



(a) Fuzzy artificial perceptron (b) Fuzzy recurrent neuron (FRN)

Figure 4. Fuzzified neural processing units.

For determining the channel coefficients

$$[\tilde{h}_{11}, \tilde{h}_{12}, \ldots, \tilde{h}_{21}, \tilde{h}_{22}, \ldots, \tilde{h}_{31}, \tilde{h}_{32}, \tilde{h}_{33}, \ldots, \tilde{h}_{41}, \ldots, \tilde{h}_{44}],$$

the expression given by (27) is used after completing the training using the back-propagation (BP) algorithm with Levenberg–Marquardt (LM) optimization.

Fuzzy Time Delay Fully Recurrent Neural Network (FTDFRNN) based Inference Engine

The FMLP based inference system is effective in case of static, slowly varying channels. The training and estimation time required, however, is much less than the pure ANN based systems as described in [15, 16]. It can also deal with time-varying channels but there is always a scope for further improvement. However, as RNNs are known to be suitable for time-varying inputs, hence an option emerges for experimenting with these systems to explore if their abilities are expanded when fuzzy attributes are incorporated. Such a combination called fuzzy RNN (FRNN) is specially configured to form a fuzzy inference system. The core of a FRNN is a fuzzy recurrent neuron (FRN) shown in Figure 4 (b). The MLP block of a FMLP is replaced by a TDFRNN [16] with split-activation so that the combination can use de-coupled in-phase and quadrature components of input signals and use a SOM to combine the outputs in an optimized form. The output of such a set of neurons for a set of inputs X_j and state vectors $u_{N,j}$ is expressed as

$$Y_{rk}(n) = \sum_{k} S_{\beta,k} \left(\sum_{j=0}^{n} (Y_{ff} + Y_{fb}) \right), \tag{30}$$

where

$$Y_{ff} = W_j X_j + W_{j-1} X_{j-1} + W_{j,y} Y_{k-1}, \tag{31}$$

$$Y_{fb} = W_{jN} u_{N,j} + W_{jN} u_{N-1,j}. \tag{32}$$

After the input propagates through the hidden and the output layers, for a real input X_{Rj}, the response shall be

$$Y_{\mathrm{rmR}} = \sum_m S_{\beta,m} \left\{ \sum_l S_{\beta,l} (Y_{\mathrm{rmRtemp}}) \right\}, \tag{33}$$

$$Y_{\mathrm{rmRtemp}} = \sum_k (Y_{\mathrm{rkR}} W_{kl} + u_{M-i,j} W_{jM}) W_{lm}. \tag{34}$$

A similar expression for imaginary inputs can also be obtained:

$$Y_{\mathrm{rmI}} = \sum_m S_{\beta,m} \left\{ \sum_l S_{\beta,l} \left(\sum_k Y_{\mathrm{rmItemp}} \right) \right\}, \tag{35}$$

where

$$Y_{\mathrm{rmItemp}} = (Y_{rkI} W_{kl} + u_{M-i,j} W_{jM}) W_{lm}. \tag{36}$$

If the training samples are presented to the inference engine for some duration of time, the output matrix generated is optimized by a SOM such that the final result obtained has a form given by

$$Y_{rF} = \mathrm{Opt}\{ Y_{\mathrm{rmR}}(n) + j Y_{\mathrm{rmI}}(n) \}, \tag{37}$$

where Opt(·) is an optimization process carried out followed by a ‘winners take all’ competition such that the winning neuron index, J, satisfies

$$Y_J(n) = \max_j \{ Y_{rFj}^T, W_j \}. \tag{38}$$

2.3 Defuzzification

After the fuzzified outputs are generated, a mapping is performed to convert each conclusion into a single real number. This mapping process provides the required estimation of the channel coefficients obtained from a fuzzy inference. There are several defuzzification methods but the ‘center of arc’ or ‘centroid method’ proposed by Sugeno (1985) and Lee (1990) is the most acceptable of all (see [7]).

3 Experimental Results

While performing the experiments, the channel model as described by expressions (1)–(16) and the system model represented by expressions (17)–(38) are used. During the membership generation process, the SOM-MLP combination is found to be less accurate by about 4% but it is faster by at least 21% compared to the Bell-shape function-MLP method, hence the former is more acceptable for speed critical applications. For time-varying cases including fast fading, a better option is the use of TDFRNN blocks in place of the MLP. This is because RNNs are known to have the capacity to deal with time-varying inputs [5], hence in fuzzified form it is also likely to have such capability. The inference engine of the FNS for MIMO channel estimation is formed by a FTDFRNN with fuzzy inputs, connectionist weights and outputs. The FTDFRNN structures are trained with DEKF algorithm modified with fuzzy considerations.

These channel coefficients are generated using the standard Clarke–Gans model with parameters given in Table 2 and as described in [15, 16] and are combined with OFDM symbols generated using the parameters shown in Table 1. While the signal propagates through the channels, significant and CCI coefficients get involved. As a result, a composite signal form is generated which covers the significant signal segments with CCI from which the fuzzy based system recovers the required portion for modeling the MIMO channels. It involves certain sessions of training the fuzzy system during which it learns the patterns applied to it. Table 4 shows a truncated data set used to generate the significant and CCI channel components for four different path delays and four different frequency selective paths with Rayleigh fading generated using the Clarke–Gans model.

A training window of a few seconds is given to the two FTDFRNN structures during which several estimates of the signal samples are generated. The SOM placed at the end of the two FTDFRNN structures is used to combine the outputs and provides an optimized estimate of the response. From the estimates of the signal, the channel coefficients are derived. Training performance is judged using the mean square error (MSE) convergence rate and the precision derived. Figure 5 presents the normalized processing time of the fuzzy methods against statistical methods (LS, MMSE), ANN based standard methods (MLP [15], 3L-FF [8], Temporal-MLP [15]) and RNN based approaches (CTDFRNN-MNSOM [16]). The fuzzy approaches generate minimum 2–6% difference in processing time while providing better precision performance consistently over at least fifteen trails. Time is provided in normalized form because, with processors and other related system variations, the time taken varies. The present work is performed using an Intel Core 2 Duo CPU T7250 at 2 GHz and 2 GB RAM with 1024 MHz FSB. Table 5 shows the precision of FMLP and FTDFRFRNN based MIMO estimators

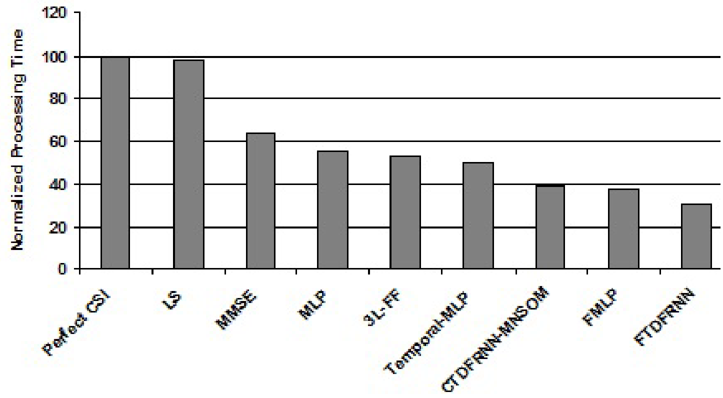


Figure 5. Relative time taken by estimation process carried out with (i) statistical methods (LS, MMSE), (ii) ANN based standard methods (MLP [15], 3L-FF [8], Temporal-MLP [15]), (iii) RNN based approaches (CTDFRNN-MNSOM [16]) and (iv) the proposed architecture (FTDFRNN).

Case	1	2	3	4
Delays	1×10^{-5}	1.5×10^{-5}	2×10^{-5}	2.5×10^{-5}
Path gain 1	0.467	0.353	0.555	0.036
Path gain 2	0.281	0.031	0.815	0.732
Path gain 3	0.367	0.893	0.462	0.381
Path gain 4	0.212	0.691	0.021	0.811

Table 4. Truncated data set used to generate significant and CCI channel components for four different delays and frequency selective paths of a Rayleigh faded channel.

Item	TDFRNN-MNSOM	FMLP	FTDFRNN
Data size	500–1000	500–1000	500–1000
Precision in %	97.4	98.2	98.8

Table 5. Precision of FMLP and FTDFRNN with respect to CTDFRNN-MNSOM.

Model	Number of additions	Number of multiplications
CTDFRNN-MNSOM	$(P + R)34{+}2{+}l$	$(P + R)24{+}2l$
FTDFRNN	$12N{+}2{+}l$	$11N{+}2{+}l$

Table 6. Computational complexity of FNS based design compared to temporal-MLP and RNN-based architectures with N length filter (for temporal MLP structures) or delay blocks (for RNN-based structures), n number of parallel structures, l size competitive layer and $(P + R)$ length of the signal.

as compared to CTDFRNN-MNSOM method generated for a VOIP based transmission in different blocks of sizes between 500 and 1000. The shown results are average values of ten trials carried out by training the two FNS estimators with at least ten repetitions of the transmissions between the source and destination having a four path Rayleigh fading. The precision values shown by the CTDFRNN-MNSOM approach are marginally lower, but in such critical applications, little improvement in accuracy is always desirable.

The estimation of the channel coefficients carried out by the FTDFRNN approach always oscillates around a small value close to the 10^{-6} mark but never reaches zero which makes it a bias estimator. Hence, Cramer–Rao (CR) bound is not applied to the MSE convergence limits with SNR variation during training, validation and testing of the system. With N delay blocks for RNN-based structures (CTDFRNN-MNSOM) [16], n number of parallel structures, l size competitive layer and $(P + R)$ length of the signal, the computational complexity shown by the FTRFRNN based method is provided in Table 6.

The CTDFRNN-MNSOM [16] is taken as the comparative model in Table 6 since it generates the best performance in terms of BER and precision among the proposed RNN-based architectures. The FTDFRNN approach has an RNN-based structure followed by a competitive layer which for each input generates sizable amount of computation which is fast and converges to the desired level within a few number of training epochs. The fuzzy approach not only shows fall in computational complexity but also demonstrates better performance in terms of precision and BER values while performing modeling of the MIMO channels. This is outlined in the subsequent description. The performance of the system is dependent on the inference rule set adopted for carnying out the estimation process. A standard set of six rules gives optimal performance but experiments are carried out to see the effect of variation of the inference stage. Table 7 summarizes the results obtained.

Case	Network structure	Number of rules	MSE $\times 10^{-5}$	Epochs	Precision in %
1	20-24-12-4	6	0.3	34	96.5
	($N1$)	9	0.06	102	97.3
2	20-30-15-4	6	0.26	34	96.8
	($N2$)	9	0.045	96	97.0
3	20-40-20-4	6	0.33	39	96.0
	($N3$)	9	0.05	101	97.0
4	20-50-25-4	6	0.24	44	96.0
	($N4$)	9	0.038	101	97.0
5	20-50-30-4	6	0.25	47	95.4
	($N5$)	9	0.039	103	96.1

Table 7. Effect in performance due to variation in network structure adopted for implementation of inference engine.

Table 7 also summarizes the variation in performance of the FNS with change in the network structure adopted for the implementation of the inference engine. The inference network structure is considered to be formed by an input, two hidden and one output layers. Four different FNS structures are used to implement the inference rules. While all the four networks using six inference rules generate an MSE convergence between 0.24×10^{-5} and 0.33×10^{-5}, the value comes down significantly to a range of 0.038×10^{-5} to 0.05×10^{-5} with nine inference rules. It indicates that the MSE convergence rate improves and falls to lower limits with more inference stages. With more inference rules, the networks learn better and approach the level of optimality with greater closeness. It amounts to an improvement between 84 to 87% in MSE convergence rates. But this happens at the cost of greater processing time. The number of epochs increases by about 1.32 to 1.7 times when twelve inference rules are used compared to the case when the system is designed with a set of six such sets. However, this increase in processing time and lowering of the MSE values results in an improvement of precision marginally between 1.04 to 1.5%. Hence, the six inference rule format is adopted to carry out the FNS based MIMO channel estimation. The network named $N4$ with a structure of 20-50-25-4 gives the best performance. MSE value converges, on an average, to about 0.24×10^{-5} during the stipulated training slots below 50

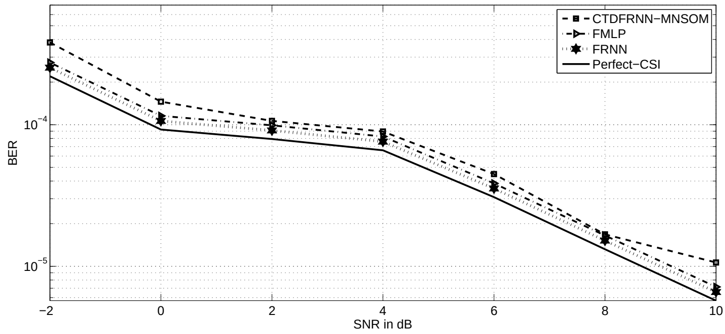


Figure 6. BER generated by FMLP and FTDFRNN in comparison to an estimator with perfect CSI.

epochs with the data set taken. This network has an input layer of 20 FRNs in the input layer. This value is derived from the fact that the signal block is formed by 8-data bits and three parity prefixes with a channel length of ten. After convolution between the signal block and the channel coefficients, the composite sample input has a sequence length of twenty. Similarly, there are 50 and 25 FRNs in the two hidden layers. The output layer has four FRNs as it needs to retain only four sets of data for a 4×4 MIMO set-up designed for the purpose. The significant part of the data set is retained, interpretation derived after de-fuzzification and BER values calculated. The channel matrix has four numbers of channels with twenty taps of which four are significant for the estimation process. Figure 6 shows the average BER vs SNR provides by FMLP and FTDFRNN systems compared to an estimator with perfect CSI. The fuzzy systems, on an average, differ by about 3–8% with the BER values generated by the estimator with perfect CSI. This is at least 5% improvement in accuracy compared to the CTDFRNN-MNSOM based estimation of which the BER values are shown as well. This is in addition to the processing time advantage that the fuzzy systems provide. The FNS based MIMO estimator is also subjected to perform phase tracking of the signal samples received. During training, the FNS is initially given 50 samples to learn the phase pattern within the first 10–20 iterations. Once it attains a desired precision, it is subjected to track phase of data sequences of length 100 which it performs well.

The performance difference in terms of BER due to the variation of MIMO transmitter and receiver numbers as stated above is summarized in Table 8.

Figure 7 shows comparative BER plots obtained using (i) statistical methods (LS, MMSE), (ii) ANN based standard methods (MLP [15]), (iii) RNN based approaches (CTDFRNN-MNSOM [16]) and (iv) the proposed architecture (FTDFRNN) in 2×2 and 4×4 set-ups in severely faded channels. The proposed

MIMO Set-up	Average difference (%)
2×2	-11.2
3×3	-9.3
3×4	-4.1
5×4	$+11.6$
5×5	$+26.2$
8×8	$+38.3$

Table 8. BER performance variation of different MIMO blocks compared to a 4×4 set-up.

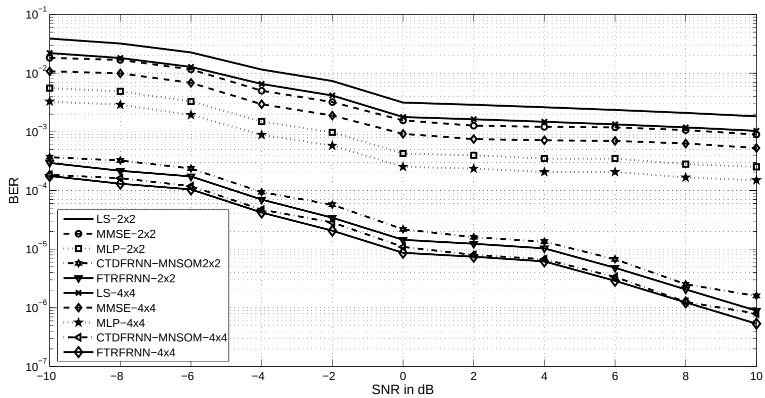


Figure 7. Comparative BER plot generated by FTDFRNN in comparison to (i) statistical methods (LS, MMSE), (ii) ANN based standard methods (MLP [15]), (iii) RNN based approaches (CTDFRNN-MNSOM [16]) and (iv) the proposed architecture (FTDFRNN) in 2×2 and 4×4 set-ups in severely faded channels.

fuzzy-based approach clearly shows improvement in the values obtained for both 2×2 and 4×4 systems.

A CCI pattern of the channels is also generated and is depicted in Figure 8. The plot shows the normalized correlation of the CCI on significant estimated channel coefficients. It indicates that though the significant channel coefficients are estimated with adequate path gains, yet significant interference exists. This stems from the fact that the FNS learns CCI patterns as well along with the significant channel coefficients. The effect of CCI can be minimized by training the FNS es-

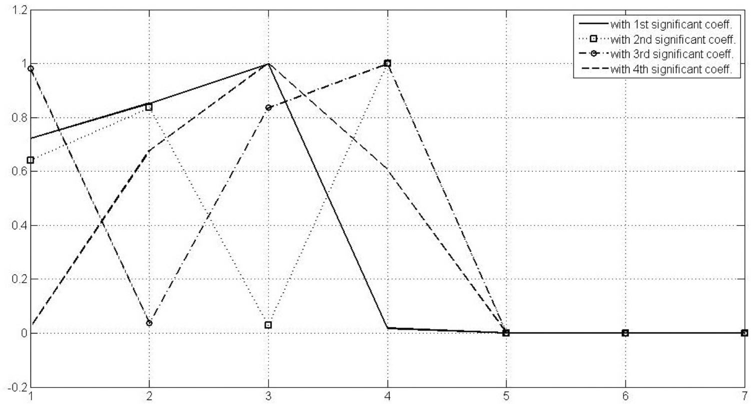


Figure 8. CCI plot generated by channel coefficients with estimated path gains.

timator with signal content that has higher significant channel coefficients which is done usually. But the depiction of Figure 8 represents a worst case scenario of certain CCI being captured significantly by the FTDFRNN blocks. Moreover despite the presence of significant co-channel interference, the FTDFRNN inference engine provides significant improvement in decision making and thereby improves the overall performance of the system. The fuzzy-based methods, thus, clearly provide advantages of faster processing time, lower BERs and better precision while carrying out symbol recovery.

4 Conclusion

We explored certain attributes of fuzzy-based composite systems with stress on architectural expansion in order to improve performance and precision compared to conventional methods while modeling the stochastic nature of the MIMO channels. We found that in the fuzzified form the inputs provide the fuzzy-based system ample of finer variations including the stochastic nature of the MIMO system to learn and thereby generate appropriate decision states. An FTDFRNN based FNS has been proposed here for MIMO channel estimation which shows significant improvement in performance compared to statistical, ANN and RNN approaches in terms of faster processing time, lower BERs and better precision while carrying out symbol recovery. The proposed method yields around 46% less computational complexity, on an average, and at least 5% improvement in accuracy compared to the best RNN-based method, namely, the CTDFRNN-MNSOM. This represents a significant improvement in performance compared to statistical and ANN based methods configured for modeling MIMO channel. Moreover, despite the presence of CCI, the FTDFRNN based MIMO modeling method proposed here provides significant improvement in decision making and thereby improves the overall performance of the system. Such a system is well suited for the design of adaptive receivers configured to tackle high data rate communication carried out using the MIMO wireless channels.

Bibliography

[1] R. H. Abiyev and T. Al-shanableh, Neuro-fuzzy network for adaptive channel equalization, in: *Fifth Mexican International Conference on Artificial Intelligence*, IEEE (2006), 237–244.

[2] R. A. Aliev, B. G. Guirimov, B. Fazlollahi and R. R. Aliev, Evolutionary algorithm-based learning of fuzzy neural networks. Part 2: Recurrent fuzzy neural networks, *Journal of Fuzzy Sets and Systems* **160** (2009), 2553–2566.

[3] T. M. Duman and A. Ghrayeb, *Coding for MIMO Communication Systems*, John Wiley and Sons, 2007.

[4] S. Ghosh, Q. Razouqi, H. J. Schumacher and A. Celmins, A survey of recent advances in fuzzy logic in telecommunications networks and new challenges, *IEEE Transactions on Fuzzy Systems* **6** (1998), 443–447.

[5] S. Haykin, *Neural Networks A Comprehensive Foundation*, 2nd ed., Pearson Education, New Delhi, 2003.

[6] M. Jiang and L. Hanzo, Multiuser MIMO-OFDM for next-generation wireless systems, *Proceedings of the IEEE* **95** (2007), 1430–1469.

[7] G. J. Klir and B. Yuan, *Fuzzy Sets and Fuzzy Logic Theory and Applications*, Prenitce Hall of India, New Delhi, 2002.

[8] Z. Ling and Z. Xianda, MIMO channel estimation and equalization using three-layer neural networks with feedback, *Tsinghua Science and Technology Journal* **12** (2007), 658–661.

[9] S. Mitra and Y. Hayashi, Neuro-fuzzy rule generation: Survey in soft computing framework, *IEEE Transactions on Neural Networks* **II** (2000), 748–768.

[10] A. Niemi, J. Joutsensalo and T. Ristaniemi, Fuzzy channel estimation in multipath fading CDMA channel, in: *Proceedings of 11th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, vol. 2, London (2000), 1131–1135.

[11] M. Nuri Seyman and N. Taspinar, Channel estimation based on adaptive neuro-fuzzy inference system in OFDM, *IEICE Transactions on Communications* **E91.B** (2008), 2426–2430.

[12] C. Oestges and B. Clerck, *MIMO Wireless Communications. From Real-World Propagation to Space-Time Code Design*, Academic Press, Elsevier, London, 2007.

[13] T. J. Ross, *Fuzzy Logic with Engineering Applications*, 2nd ed., Wiley India, New Delhi, 2008.

[14] P. K. Sahu, S. K. Patra and S. P. Panigrahi, Non-linear channel equalization using computationally efficient neuro-fuzzy channel equalizer, in: *Proceedings of 18th Iranian Conference on Electrical Engineering* (ICEE), Isfahan (2010), 326–330.

[15] K. K. Sarma and A. Mitra, Estimation of MIMO wireless channels using artificial neural networks, in: *Cross-Disciplinary Applications of Artificial Intelligence and Pattern Recognition: Advancing Technologies*, IGI Global (2011), 509–545.

[16] K. K. Sarma and A. Mitra, Modeling MIMO channels using a class of complex recurrent neural network architectures, *AEU International Journal of Electronics and Communication* **66** (2012), 322–331.

[17] H. Shatila, M. Khedr and J. H. Reed, Channel estimation for WiMaX systems using fuzzy logic cognitive radio, in: *Proceedings of IFIP International Conference on Wireless and Optical Communications Networks* (WOCN ’09), Cairo (2009), 1–6.

[18] J. Wen, C. Chang, G. Lee and C. Huang, OFDM channel prediction using fuzzy update LMS algorithm in time-variant mobile channels, in: *Proceedings of IEEE 64th Vehicular Technology Conference*, Montreal (2006), 1–5.

[19] J. Zhang, Z. M. He, X. Wang and Y. Huang, A TSK fuzzy approach to channel estimation for OFDM systems, *Journal of Electronic Science and Technology in China* **4** (2006), 101–105.

Received December 10, 2011.

Author information

Kandarpa Kumar Sarma, Department of Electronics and Electrical Engineering,
Indian Institute of Technology Guwahati, Guwahati 781014, Assam, India.
E-mail: `s.kandarpa@iitg.ernet.in`

Abhijit Mitra, Department of Electronics and Electrical Engineering,
Indian Institute of Technology Guwahati, Guwahati 781014, Assam, India.
E-mail: `a.mitra@iitg.ernet.in`