Chapter 5

Direct Signal Detection Without Data-Aided: A MIMO Functional Network Approach

Xiukai Ruan , Yanhua Tan , Yuxing Dai , Guihua Cui ,
Xiaojing Shi , Qibo Cai , Chang Li , Han Li ,
Yaoju Zhang and Dameng Dai

Additional information is available at the end of the chapter

http://dx.doi.org/10.5772/63213

Abstract

Functional network (FN) has been successfully applied in many fields, but so far no methods of direct signal detection (DSD) using FN have been published. In this chapter, a novel DSD approach using FN, which can be applied to cases with a plural source signal sequence, with short sequence, and even with the absence of a training sequence, is presented. Firstly, a multiple-input multiple-output FN (MIMOFN), in which the initial input vector is devised via QR decomposition of the receiving signal matrix, is constructed to solve the special issues of DSD. In the meantime, the design method for the neural function of this special MIMOFN is proposed. Then the learning rule for the parameters of neural functions is trained and updated by back-propagation (BP) algorithm. The correctness and effectiveness of the new approach are verified by simulation results, together with some special simulation phenomena of the algorithm. The proposed method can detect the source sequence directly from the observed output data by utilizing MIMOFN without a training sequence and estimating the channel impulse response.

Keywords: direct signal detection, MIMO, functional neural network, QAM, back propagation

1. Introduction

Signal detection has become an integral part of applications in many areas, including wireless communications, optical telecommunications, automatic control system, magnetic resonance

imaging, underwater acoustics, and radio astronomy. Typical digital communication environments involve transmission of analog pulses over a dispersive medium, inevitably corrupting the received signal by inter-symbol interference (ISI). The present sequence estimation approaches, for example, maximum-likelihood sequence estimation (MLSE) [1] and 'Bussgang' [2–4], usually estimate first the communication channel impulse response and then the transmitted sequence by an optimum method. MLSE is a useful approach for equalization of ISI but requires knowledge of the possibly time-variant channel impulse response, and its complexity increases exponentially with the length of the channel delay spread. Consequently, adaptive channel estimation algorithms have to be employed [1]. "Bussgang" and its modified algorithms [2–4] are well-known channel estimation algorithms but are costly and indirect due to their necessity for a long block of data to achieve algorithm convergence since they exploit implicit (embedded) statistical knowledge. Direct signal detection (DSD) is a new emerging method in communication systems, which can directly estimate the input sequence without estimating the channel impulse response [5–7].

In this work, we propose a new method that merges the functional network (FN) [8–9] models into a DSD technique for removing ISI without the training sequence. FN is a very useful general framework for modeling a wide range of probabilistic, statistical, mathematical, and engineering problems [10]. For instance, Iglesias et al. [11] adopted FN to improve the results of a determined artificial neural networks (ANN) application for the estimation of fishing possibilities, Zhou et al. [12] employed FN to solve classification problem, Emad et al. [13] found that FN (separable and generalized associativity) architecture with polynomial basis was accurate, was reliable, and outperforms most of the existing predictive data mining modeling approaches for the issue of permeability prediction, and Alonso-Betanzos et al. [14] applied FN to predict the failure shear effort in concrete beams. As mentioned above, FN has been successfully applied in many fields, but so far methods of DSD using FN have not been reported.

The chapter is organized as follows. In Section 2, a system model is described. Then the network structure of MIMOFN to solve the special issues of DSD is constructed in Section 3. A multiple-input multiple-output FN (MIMOFN), in which the initial input vector is devised via QR decomposition of receiving signal matrix, is constructed to solve the special issues of DSD. The design method of the neural function of this special MIMOFN is proposed. Then the learning rule for the parameters of neural functions is trained and updated by back-propagation (BP) algorithm in Section 4. In Section 5, simulation results are shown to verify the new approach's correctness and effectiveness, and some special simulation phenomena of this algorithm are given out followed by Conclusion and Discussion in Section 6.

2. The system model and basic assumptions

This system model is a linear single-input multiple-output (SIMO) (see **Figure 1**) finite impulse response channel with N output [15]. The following basic assumptions are adopted [16].

A1: Channel order L_h is assumed to be known as a priori.

http://dx.doi.org/10.5772/63213

A2: The input signal $\mathbf{s}(t)$ is zero-mean and temporally independent and identically distributed (i.i.d).

A3: Additive noise is spatially and temporally white noise and is statistically independent of the source.

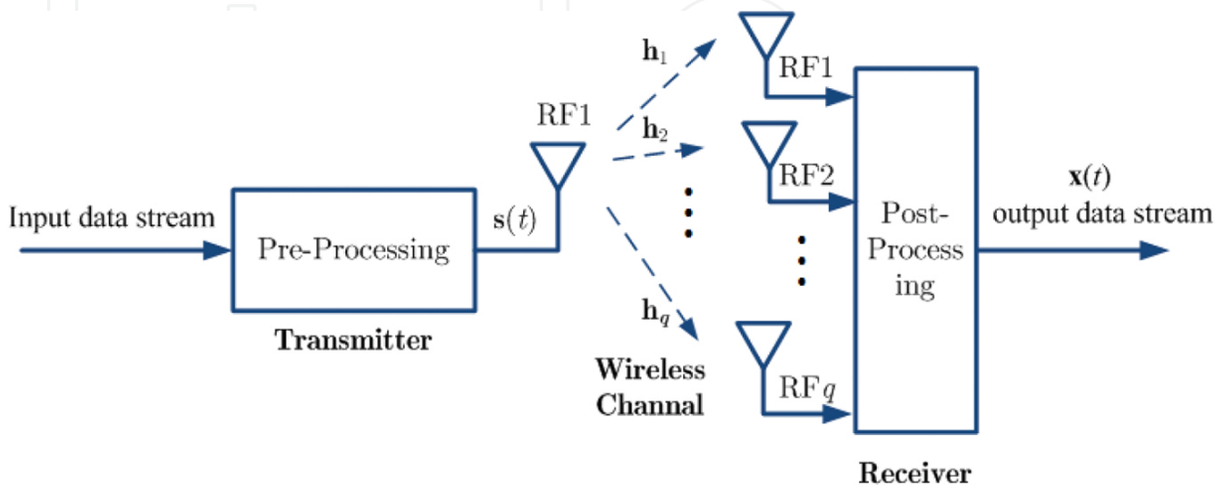


Figure 1. Example of SIMO channel.

For the simplification of the presentation of the proposed DSD without loss of generality, the i-th sub-channel output of an SIMO channel system in a noise-free environment is expressed as

$$x_i(t) = \sum_{\tau=0}^{L_h-1} h_i(\tau) s(t-\tau) \tag{1}$$

where $s(t)$ is the Quadrature Amplitude Modulation (QAM) input signal, and $h_i(\tau)$ is the i-th system impulse response of length L_h[5]. Clearly, the column vector $\mathbf{x}(t)$ can be expressed as

$$\left(\mathbf{x}(t)\right)_{q\times 1} = \left[\mathbf{h}_1, \mathbf{h}_2, \cdots, \mathbf{h}_q\right]^T_{q\times(L_h+1)} \cdot \left(\mathbf{s}(t)\right)_{(L_h+1)\times 1} \tag{2}$$

where q and L_h denote the over-sample factor and the order of channel impulse response, respectively, and superscript T denotes transpose of a matrix, and

$$\mathbf{h}_p = \left[h_p(0), h_p(1), \cdots, h_p(L_h)\right]^T_{1\times L_h}, p = 1,2,\cdots,q \tag{3}$$

and

$$\mathbf{s}(t)=\left[s(t),s(t-1),\cdots,s(t-L_h)\right]^{\mathrm{T}}_{(L_h+1)\times 1} \tag{4}$$

Thus, the received data matrix can be formulated as

$$(\mathbf{X})_{N\times(L_w+1)q}=\mathbf{S}\Gamma \tag{5}$$

where $(\mathbf{S})_{N\times\left(L_h+L_w+1\right)}=\left[\mathbf{s}_N(t),\mathbf{s}_N(t-1),\cdots,\mathbf{s}_N\left(t-L_h-L_w\right)\right]$,

and $\mathbf{s}_N(t)=[s(t),s(t-1),\cdots,s(t-N+1)]^T$ is the transmitted signal matrix, N and L_w are the source signal length and length of the equalizer, respectively, and Toeplitz matrix $\mathbf{\Gamma}=\left[\mathbf{H}_1^T,\mathbf{H}_2^T,\cdots\mathbf{H}_q^T\right]$ with

$$\mathbf{H}_p=\begin{bmatrix}\mathbf{h}_p^T & 0 & \cdots & 0 & 0\\ 0 & \mathbf{h}_p^T & 0 & \ddots & 0\\ \vdots & \ddots & \ddots & \ddots & \vdots\\ 0 & \cdots & 0 & \mathbf{h}_p^T & 0\end{bmatrix},\ p=1,2,\cdots q \tag{6}$$

3. The FN structure for DSD

FN is a generalization of neural networks achieved by using multi-argument and learnable functions, that is, in these models, transfer functions, associated with neurons, are not fixed but learned from data. There is a need to include weights to ponder links among neurons since their effect is subsumed by the neural functions. **Figure 2** shows an example of a general FN topology, where the input layer consists of the units $G=(x,y,z)$; the first layer contains neurons P, G, N, Q; the second layer contains neurons J, K, F, L; and the output layer contains α, β, γ, θ. According to Castillo's approach, each neuron function, P, G, N, Q, J, K, F, L, is represented as a linear combination of the known functions of a given family such as polynomials, trigonometric functions, and Fourier expansions and is estimated during the learning process. Generally, instead of fixed functions, FNs extend the standard neural networks by allowing neuron functions P, G, N, Q , J, K, F, L to be not only true multi-argument and multivariate functions but also different and learnable. Two types of learning methods exist: structural learning and parametric learning. The latter estimates the activation functions with the consideration of the combination of "basis" functions such as the least square, steepest descent, and mini-max methods [13]. In this chapter, the least square method for estimating sequence is used.

http://dx.doi.org/10.5772/63213

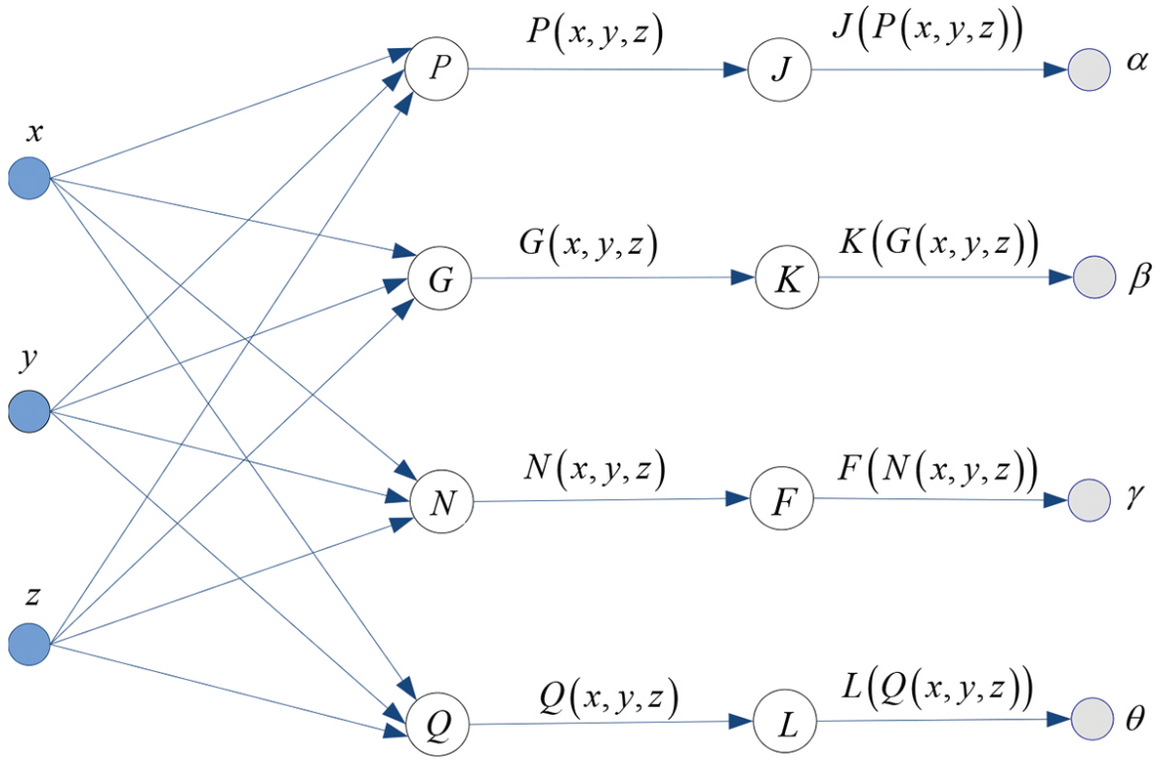


Figure 2. A basic FN topology model.

***Definition 1*:** Assume there exists a function set F that transforms an n-dimensional input vector into an m-dimensional output vector in a complex regime $F: \mathbf{C}^n \to \mathbf{C}^m$, and

$X \in \mathbf{C}^n$, $X = (x_1, x_2, \cdots, x_n)$, $f_1 = f_1(x_1, x_2, \cdots, x_n)$, $f_2 = f_2(x_1, x_2, \cdots, x_n)$, $\cdots$, $f_m = f_m(x_1, x_2, \cdots, x_n)$, then the FN model $F = F(f_1, f_2, \cdots, f_m)$ is a basic MIMOFN model.

***Definition 2*:** Let X be a linear combination of a column in $\mathbf{C}^n$, and $f_i = \sum_{j=1}^{n} a_{ij} x_j + c_i$, $(i = 1, 2, \cdots, m)$, $x_j \in X, j = 1, 2, ..., n$, where both a_{ij} and c_i are constant values, then $F = F(f_1, f_2, \cdots, f_m)$ is a linear MIMOFN.

Generally, FNs are driven by specific problems, that is, the initial architecture is designed based on problems in hand. Here, an MIMOFN, shown in **Figure 3**, is designed for the special issue of DSD. In this topology, $\mathbf{w}_1$, $\mathbf{w}_2$,…, $\mathbf{w}_N$ is the input vector, and f(.), g(.), and h^{-1}(.) denote the neuron functions of the first, second, and output layer, respectively. Re(.) and Im(.) denote the real part and imaginary part, respectively.

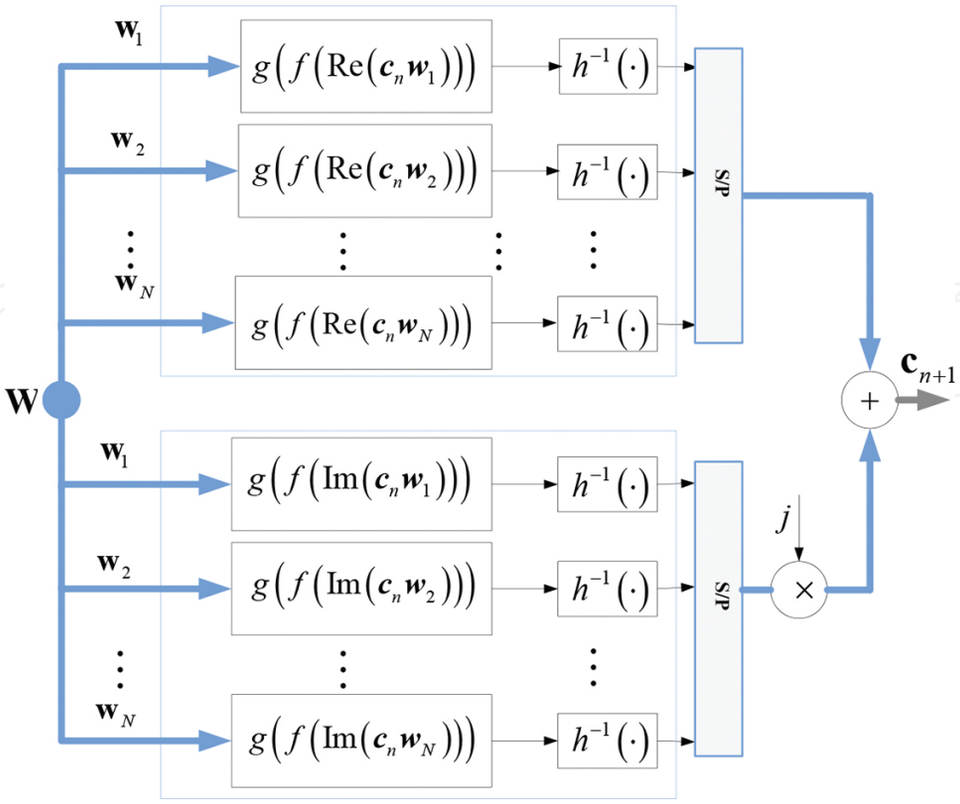


Figure 3. The MIMOFN topology for DSD.

In this case, $\mathbf{c}_{n+1}$ is the output of MIMOFN and its elements are

$$\begin{aligned} c_{n+1,1} &= h^{-1}\left(g\left(f\left(\mathrm{Re}\left(\mathbf{c}_n \mathbf{w}_1\right)\right)\right)\right) + j \cdot h^{-1}\left(g\left(f\left(\mathrm{Im}\left(\mathbf{c}_n \mathbf{w}_1\right)\right)\right)\right) \\ c_{n+1,2} &= h^{-1}\left(g\left(f\left(\mathrm{Re}\left(\mathbf{c}_n \mathbf{w}_2\right)\right)\right)\right) + j \cdot h^{-1}\left(g\left(f\left(\mathrm{Im}\left(\mathbf{c}_n \mathbf{w}_2\right)\right)\right)\right) \\ &\cdots \\ c_{n+1,N} &= h^{-1}\left(g\left(f\left(\mathrm{Re}\left(\mathbf{c}_n \mathbf{w}_N\right)\right)\right)\right) + j \cdot h^{-1}\left(g\left(f\left(\mathrm{Im}\left(\mathbf{c}_n \mathbf{w}_N\right)\right)\right)\right) \end{aligned} \tag{7}$$

where all of $f(.)$, $g(.)$, and $h^{-1}(.)$ are arbitrary continuous and strictly monotonic functions, and j is the imaginary unit.

4. DSD using MIMOFN

4.1. The input vector of MIMOFN

Our objective is to detect the source sequence directly from the observed output data by utilizing MIMOFN without training sequence and estimating the channel impulse response.

Since $\mathbf{X} \in \mathbf{C}^{N \times (L_w+1)q}$ with $N \geq (L_w+1)q$, QR decomposition can be applied to $\mathbf{X}$, and thus $\mathbf{X}$ can be rewritten as

$$\mathbf{X} = \bar{\mathbf{Q}}\bar{\mathbf{R}} \tag{8}$$

where $\bar{\mathbf{Q}} \in \mathbf{C}^{N \times (L_w+1)q}$ is a matrix with orthonormal columns and $\bar{\mathbf{R}} \in \mathbf{C}^{(L_w+1)q \times (L_w+1)q}$ is an upper triangular matrix such that $\bar{\mathbf{R}}(j,j) \neq 0, j = 1, \cdots, (L_w+1)q$ [17].

For the DSD issues, there is a well-defined gap in the singular values of $\mathbf{X}$; in other words, it exists as an index r that $\sigma(r+1) \ll \sigma(r)$, where $\sigma(\cdot)$ denotes the singular of $\mathbf{X}$, and then the subset selection will tend to produce a subset containing the most important columns (rules) of $\mathbf{X}$.

In order to obtain the full QR factorization, we proceed with SVD and extend $\bar{\mathbf{Q}}$ to a unitary matrix $\mathbf{Q}_1$. Then, $\mathbf{X} = \mathbf{Q}_1\mathbf{R}_1$ with unitary $\mathbf{Q}_1 \in \mathbf{C}^{N \times N}$ and upper triangular $\mathbf{R}_1 \in \mathbf{C}^{N \times (L_w+1)q}$, with the last $N - (L_w+1)q$ rows being zero. So we obtain

$$\mathbf{X} = \mathbf{Q}_1\mathbf{R}_1 = [\mathbf{Q}, \mathbf{Q}_c] \cdot \begin{bmatrix} \mathbf{R} \\ \mathbf{0} \end{bmatrix} \cdot \mathbf{E} \tag{9}$$

where $\mathbf{E}$ is a permutation matrix. The $\mathbf{R}$-values are arranged in decreasing order and incline to track the singular values of $\mathbf{X}$.

Thus, the input matrix of MIMOFN W can be formulated as

$$\mathbf{W} = \mathbf{Q}\mathbf{Q}^H \tag{10}$$

where superscript H denotes conjugate transpose of a matrix. Clearly, W is a non-negative definite idempotent Hermitian matrix.

Since the column vectors of W are a base set of signal space, they can be adopted to the input vector of MIMOFN and remain unchanged.

4.2. Cost function

The next issue to be considered is construction of a cost function. The cost function of MIMOFN can be formulated as

$$J(\mathbf{c}) = \frac{1}{2}E\left(\left\|\mathbf{c}_n - \mathbf{c}_{n-1}\right\|_2^2\right) = \frac{1}{2}E\left(\left\|\mathbf{c}_{n+1} - h^{-1}\left(g\left(\mathbf{W}\mathbf{c}_n\right)\right)\right\|_2^2\right) \tag{11}$$

where $h^{-1}(x) = x$, $E\{.\}$ is expectation operation, and $\|\cdot\|_2^2$ is two-norm. Clearly, it is true that $\mathbf{W}\mathbf{s}_N(t-d) = \mathbf{s}_N(t-d)$ is satisfied, here $\left\{s_N(t-d)\middle| d = 0, \cdots, L_h + L\right\}$, the value of cost function is minimum.

4.3. Design of neural functions

Neural function $g(.)$ must be easy to compute and complement, and hence needs to satisfy the following conditions:

1. $g(.)$ should be simple, easy to calculate, continuously differentiable, and bounded.
2. The derivative of $g(.)$ should be simple too, and it is the ordinary transformation of $g(.)$.
3. A priori knowledge of special issue must be considered, which makes the network easy to be trained with strong generalization ability in a smaller scale.

Since both the in-phase and quadrature-part of QAM signals belong to P^N, we only consider the in-phase part of QAM signals and design the following neural function $f(.)$ as

$$g(x, N_s) = \left[2\sum_{i=1}^{N_s}\sigma_s(x + b_i)\right] - N_s \tag{12}$$

where $b_i = (N_s + 1) - 2i, i = 1, 2, \cdots, N_s$, N_s are the number of multi-levels and the unstable inflexion points of the basis function $\sigma_s(x + b_i) = \left[1 + e^{-a(x + b_i)}\right]^{-1}$, here $\sigma_s(x + b_i)$ is the arbitrary continuous and strictly monotonic function. a is the attenuation factor of the basis function. Furthermore, the derivative of $g(.)$ can be calculated by

$$\begin{aligned} g'(x, N_s) &= \left\{\left[2\sum_{i=1}^{N_s}\sigma_s(x + b_i)\right] - N_s\right\}' \\ &= 2\sum_{i=1}^{N_s}\sigma'_s(x + b_i) = 2\sum_{i=1}^{N_s}a\sigma_s^2(x + b_i)\cdot e^{-a(x+b_i)} \end{aligned} \tag{13}$$

Figure 4 shows the curves of $g(.)$ neural function and its derivatives g'(.) for 16QAM with a in the Descartes coordinate. The value of a influences the convergence rate and performance of the algorithm significantly. A small value of a (e.g. $a = 5$) will accelerate the convergence rate of algorithm.

http://dx.doi.org/10.5772/63213

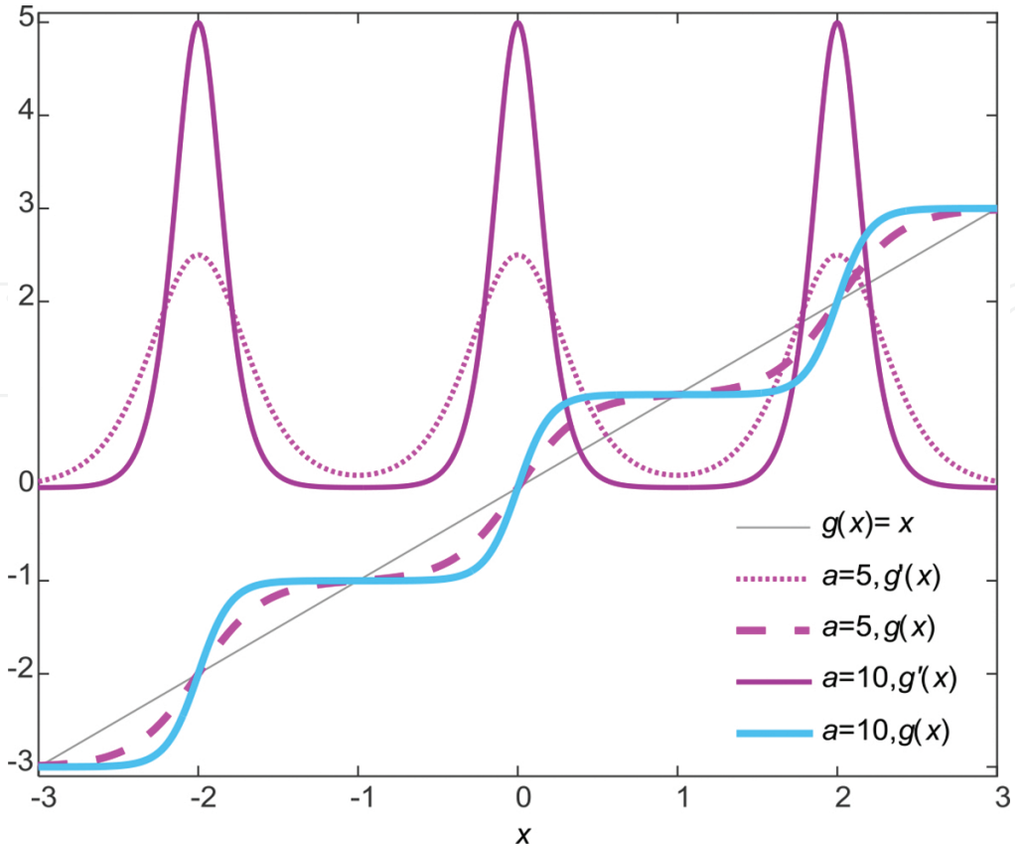


Figure 4. 16QAM, the curves of $g(.)$ neural function and its derivative $g'(.)$ with $a = 5,10$ in the x–y plane.

5. The learning rule for the parameters of neural functions

Herein, the input vector and the cost function of MIMOFN are determined, and the following mission is design of a learning strategy of the MIMOFN's neural function.

(1) The learning strategy of the parameters of $f(.)$.

A back-propagation(BP) algorithm is adopted to update the parameters of $f(.)$. Since the error function has been shown in Eq. (11), we can obtain the following equation

$$\mathbf{c}_{n+1} = \mathbf{c}_n - \mu \frac{\partial J}{\partial c_n} = \mathbf{c}_n - \mu\left(\mathbf{I} - \mathbf{W}g'\left(\mathbf{W}\mathbf{c}_\mathbf{n}\right)\right) \tag{14}$$

where $\mu \in (0, 1]$ is the learning rate factor.

Let us look into the i-th neuron. At the first iteration, the neural functions $f(.)$ are multi-dimensional linear functions.

$$f\left(\mathrm{Re}\left(\mathbf{c}_n\mathbf{w}_N\right)\right)=\mathrm{Re}\left(c_{0,1}w_{i,1}+c_{0,2}w_{i,2}+\cdots+c_{0,N}w_{i,N}\right) \tag{15}$$

$$f\left(\mathrm{Im}\left(\mathbf{c}_n\mathbf{w}_N\right)\right)=\mathrm{Im}\left(c_{0,1}w_{i,1}+c_{0,2}w_{i,2}+\cdots+c_{0,N}w_{i,N}\right) \tag{16}$$

where $c_{0,1}, c_{0,2}, \cdots, c_{0,N}$ are the parameters of $f(.)$ and $w_{i,1}, w_{i,2}, \cdots, w_{i,N}$ denote the input variables.

At the n-th iteration, we have

$$\mathbf{c}_{n+1}=\mathbf{c}_n-\mu\frac{\partial J}{\partial c_n}=\mathbf{c}_n-\mu\left(\mathbf{I}\ \text{-}\mathbf{W}\mathbf{g}'\left(\mathbf{W}\mathbf{c}_\mathbf{n}\right)\right) \tag{17}$$

and $f(.)$ become

$$f\left(\mathrm{Re}\left(\mathbf{c}_n\mathbf{w}_N\right)\right)=\mathrm{Re}\left(c_{n,1}w_{i,1}+c_{n,2}w_{i,2}+\cdots+c_{n,N}w_{i,N}\right) \tag{18}$$

$$f\left(\mathrm{Im}\left(\mathbf{c}_n\mathbf{w}_N\right)\right)=\mathrm{Im}\left(c_{0,1}w_{i,1}+c_{0,2}w_{i,2}+\cdots+c_{0,N}w_{i,N}\right) \tag{19}$$

To quicken the learning rate, an adaptive BP training algorithm with a momentum term is adopted, has shown in Eq. (20), is used

$$\mathbf{c}_{n+1}=\mathbf{c}_n-\mu\left(\mathbf{I}\ \text{-}\mathbf{W}\mathbf{g}'\left(\mathbf{W}\mathbf{c}_n\right)\right)+\Delta\mathbf{c}_n=\mathbf{c}_n-\mu\left(\mathbf{I}\ \text{-}\mathbf{W}\mathbf{g}'\left(\mathbf{W}\mathbf{c}_n\right)\right)+\left(\mathbf{c}_n-\mathbf{c}_{n-1}\right) \tag{20}$$

(2) The learning strategy of the parameters of $g(.)$.

The attenuation factor a of the $g(.)$ function is not fixed but adjusted in accordance with the value of error function. First, when the value of error function is large, a takes the small value in order to expedite the adjusting proceedings of MIMOFN output signals, in other words, to accelerate the convergence rate, making the input sequence leave the origin of coordinate quickly. As the iteration proceeds, the value of error function will become smaller and smaller, and then a large value of a should be taken to make the output sequence go to the ideal constellation points.

http://dx.doi.org/10.5772/63213

$$a(t) = -C \cdot \left(\frac{\sin(t-A)}{t-A} + B \right) \cdot \varepsilon(t-A) + \min\left(\frac{\sin(t-A)}{t-A} + B \right) \cdot \left(\varepsilon(t) - \varepsilon(t-A) \right) \tag{21}$$

where $\varepsilon(t)$ is the unit step function, B and C are constants, and A is the iteration times of $J(\mathbf{c}_{A+1}) < J(\mathbf{c}_A)$. The curve of $a(t)$ when $A = 10$, $B = 10$, and $C = 5$ is shown in **Figure 5**.

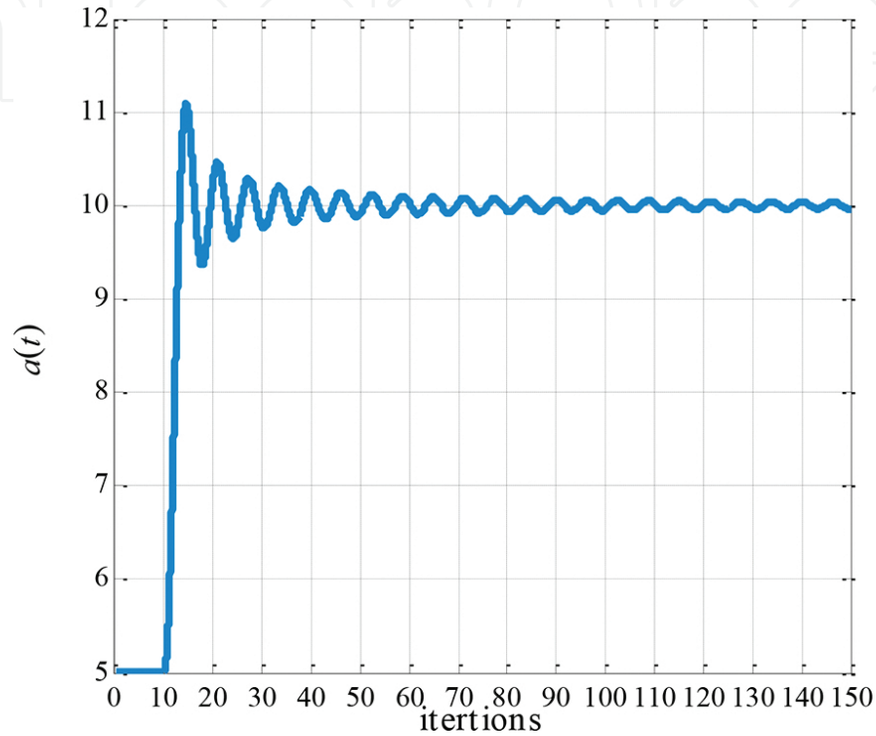


Figure 5. The curve of $a(t)$ when $A = 10$, $B = 10$, and $C = 5$, respectively.

6. Simulation and results

In this section, simulation results are provided to illustrate and verify the developed theory. Unless noted otherwise, these experiments are based on a multi-path channel $h(t) = \sum_{k=1}^{N_L} \left(\gamma_k^R \left(h^R(v, t-\tau_k^R) \right) + j \cdot \gamma_k^I \left(h^I(v, t-\tau_k^I) \right) \right)$, with an over-sampling factor $q = 4$ and a multi-path number $N_L = 5$. Here $h^R\left(v, t-\tau_k^R\right)$ and $h^I(v, t-\tau_k^I)$ are the raised cosine roll-off finite impulse responses with a roll-factor $v = 0.1$, delay factors τ_k^R and τ_k^I are random constants, and γ_k^R and $\gamma_k^I \in (0, 1]$ are random weight coefficients (see **Figure 6**). And the order L of $f(.)$ is [5, 18]. The signal modulation type is 16QAM and satisfies block-fading feature. Results are averaged over 500 Monte Carlo runs.

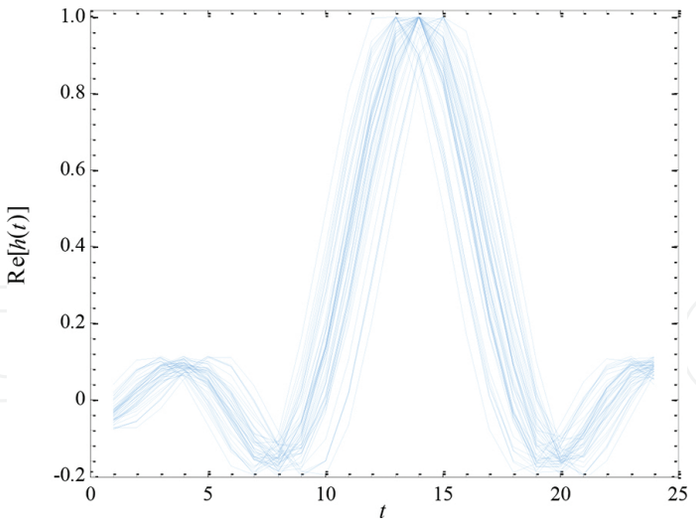


Figure 6. The sample of the real part of $h(t)$.

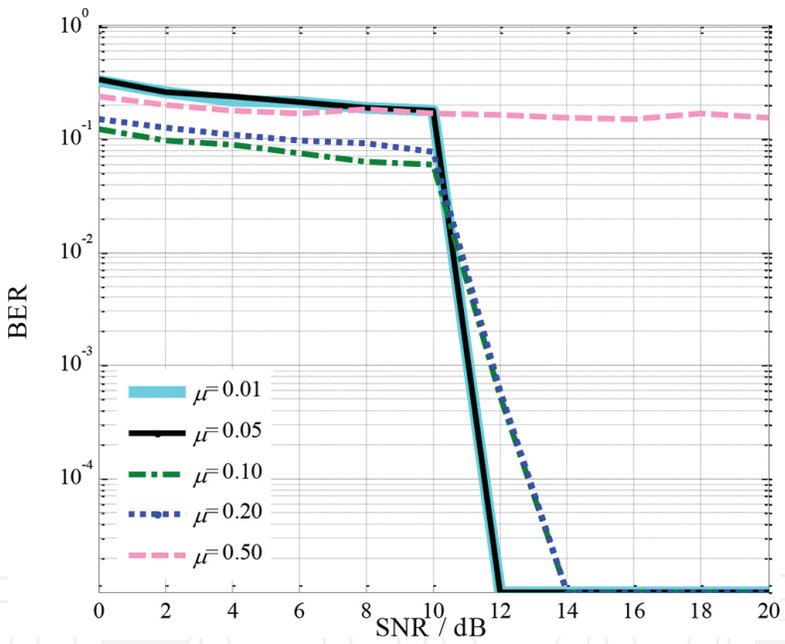


Figure 7. The average Bit Error Rate (BER) of DSD using the MIMOFN approach with the learning rate μ when data length $N = 500$.

Figure 7 shows the average BER curves of DSD using an MIMOFN approach with the learning rate μ when data length $N = 500$. This improved performance results from the decrease of μ. In the meantime, it is shown that the performance will stop improving when μ is small enough (e.g. $\mu \leq 0.05$). As we all know that increasing μ will lead to faster convergence rate of this approach, however, too large μ (e.g. $\mu = 0.5$) will make the approach fail. In the meantime, **Figure 7** also illustrates that the MIMOFN approach can work well even if the signal-to-noise ratio (SNR) = 12 dB, which is hard to reach for most existing DSD.

http://dx.doi.org/10.5772/63213

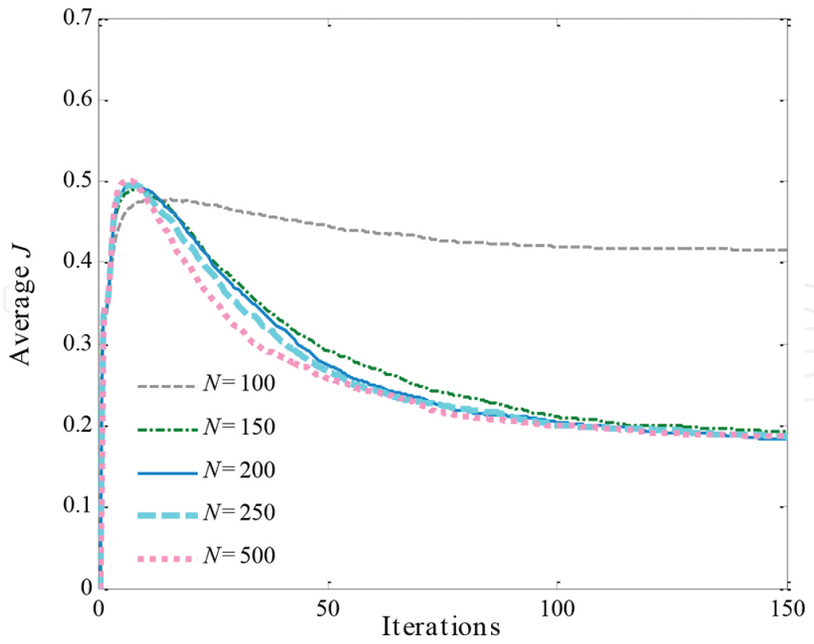


Figure 8. The curves of the average cost function J of the MIMOFN approach with several different lengths of data N = (100, 150, 200, 250, 500), when $\mu = 0.05$ and SNR = 15 dB, respectively.

Figure 8 illustrates the curves of the average cost function J of the MIMOFN approach with several different lengths of data N = (100, 150, 200, 250, 500), when $\mu = 0.05$ and SNR = 15 dB, respectively. In the initial iteration stage, the elements of sequence are amplified from the original point to the whole area of $[-3, 3]$. This makes the value of cost function to progressively increase. Regardless of the initial iteration stage, it is obvious that the average cost function keeps decreasing until its value is not changed, which means the algorithm is always convergent. In addition, the iteration is only about 150 times for 16QAM input case. Furthermore, it is implied that the cost function value is independent of data length.

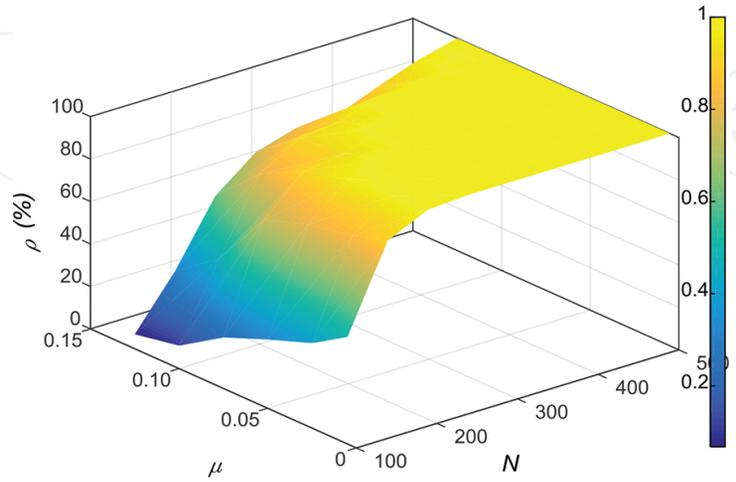


Figure 9. The 3D surface of the rate of correct recognition ρ with μ and N.

					N				
μN	**100**	**150**	**200**	**250**	**300**	**350**	**400**	**450**	**500**
0.01	50.4	90	98.8	100	100	100	100	100	100
0.05	40.0	84.8	97.8	100	100	100	100	100	100
0.10	32.4	80.8	96.2	100	100	100	100	100	100
0.15	24.0	73.4	90.8	93.2	100	100	100	100	100
0.20	10.8	46.0	80	91.2	93.2	100	100	100	100
0.25	6.8	30.8	60.4	75.2	80.8	80.8	88.4	95.2	100

Table 1. The relation of the rate of correct recognition ρ (%) with μ and N.

Figure 9 and **Table 1** illustrate the rate of correct recognition ρ of the MIMOFN approach with several different learning rate factors μ (= 0.01, 0.05, = 0.1, 0.15, 0.2, 0.25) and N = (100, 150 ,…, 500) for SNR = 15 dB.

$$\rho(\%) = \frac{\eta}{\xi} \times 100\% \tag{22}$$

where η, ξ are the total number of independent trials and the total number of correct recognition, respectively. The rate of correct recognition of the MIMOFN approach is inversely proportional to the values of μ and N.

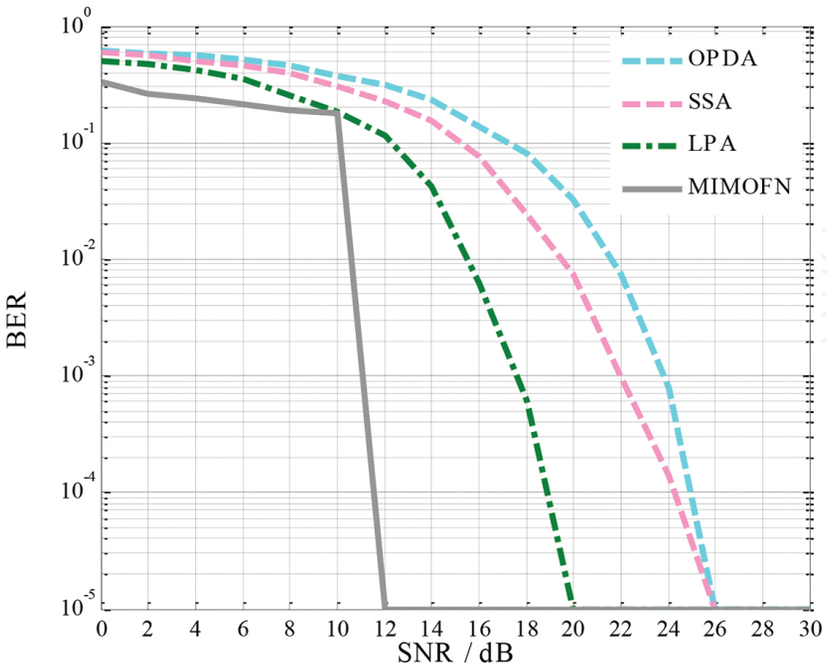


Figure 10. BER performance comparison between sequence estimation algorithms for N = 500.

http://dx.doi.org/10.5772/63213

We compare our method (MIMOFN) with the classical subspace algorithm (SSA), linear prediction algorithm (LPA), and outer-product decomposition algorithm (OPDA) [15], which are shown in **Figure 10**. It is found that the MIMOFN approach is superior to those of the above-mentioned DSD in performance.

The following experiments are based on 16QAM, SNR = 15 dB, $N = 200$, and $\mu = 0.05$. These figures are drawn by one independent trial.

In **Figure 11**, the solid circles and hollow circles denote ideal 16QAM signal constellation points and the positions of the signal points with the iteration, respectively. **Figure 12** illustrates the phase trajectories of signal points using the MIMOFN approach. All lines denote the phase trajectories of signal points with the iteration, respectively, and the eight bold lines express the trajectories of the given different signal points, respectively. We can see that the phase trajectories are different and irregular, but all of them will reach their respective true signal points when the algorithm is convergent. In addition, the iteration is only about 150 times even for 16QAM input case.

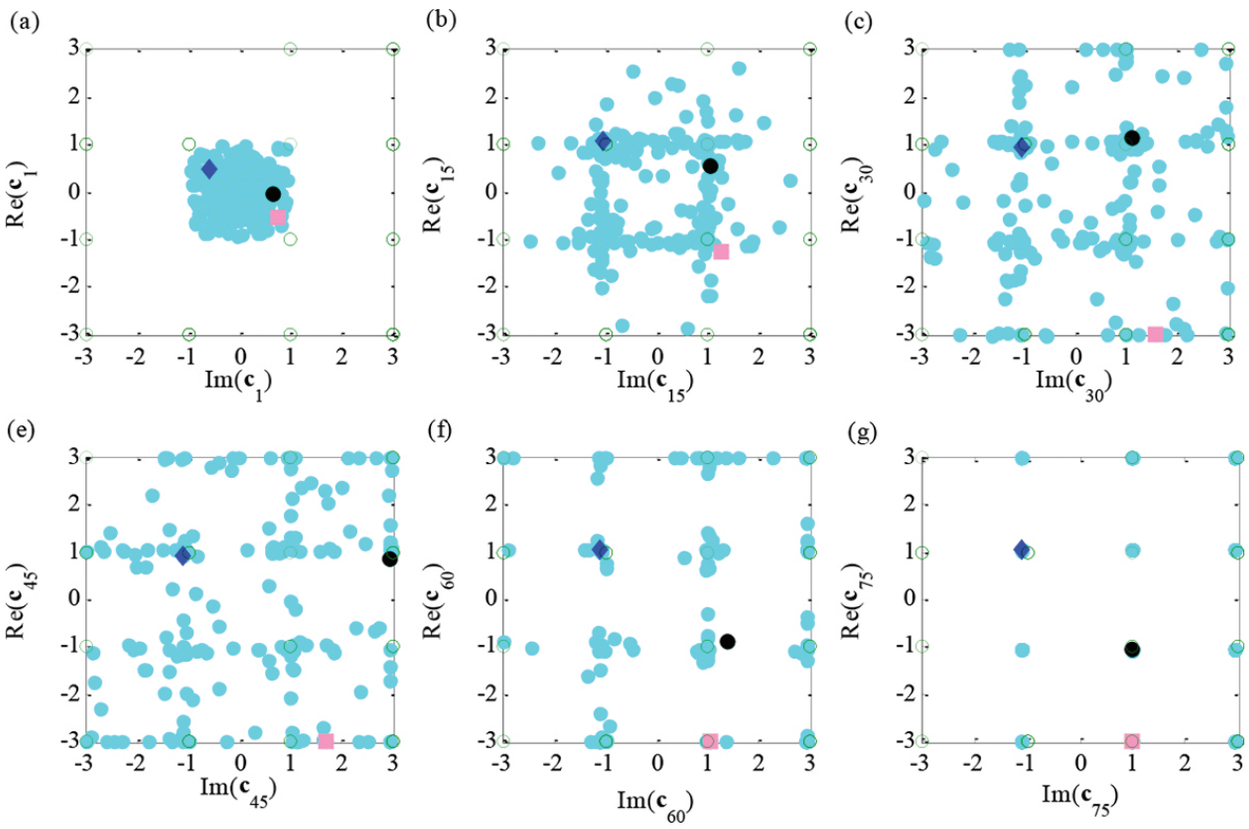
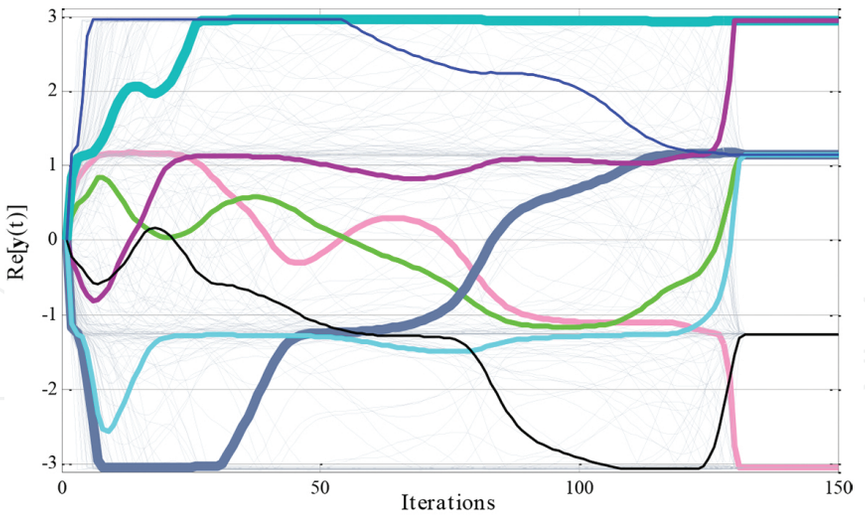
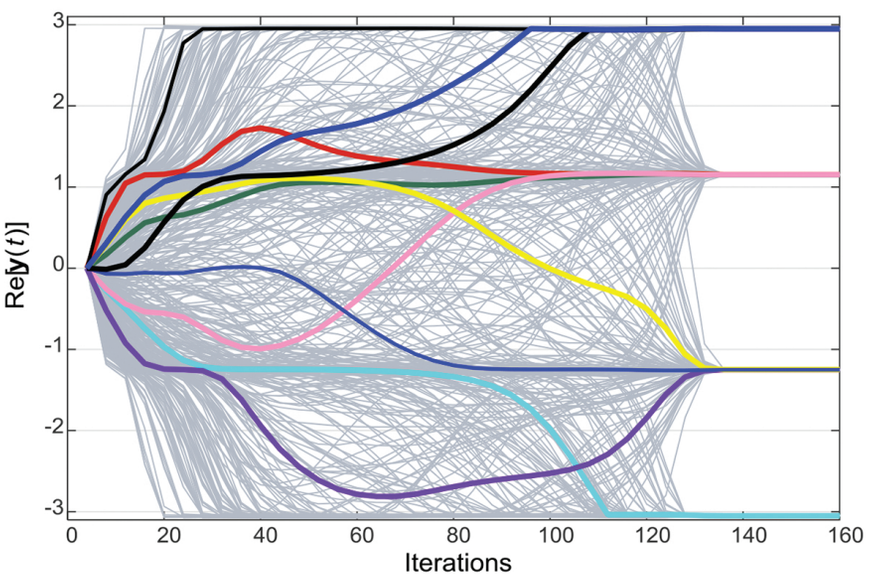


Figure 11. The constellations of input and output sequences for 16QAM signal when the MIMOFN approach is performed.



(a) N=200



(b) N=500

Figure 12. The phase trajectories of signal points using the MIMOFN approach with (a) N = 200 and (b) N=500, when μ and SNR = 15 dB, respectively.

7. Conclusions

In this chapter, a unified approach based on MIMOFN to solve DSD issues, even if the sequence is short and the training sequence is absent, is shown. The proposed method can be applied to those cases where the constellation of the source signal is dense and the data frame is short. The structure of MIMOFN not only is suitable to the DSD of square-QAM (e.g. 16QAM) issues but also can be extended to the cross-QAM (e.g. 8QAM, 32QAM) cases.

Acknowledgements

The authors would like to acknowledge the national support of this work from the National Natural Science Foundation of China (NSFC) (grant nos. 61201426, 61501331, 61303210), Zhejiang Provincial Natural Science Foundation of China (grant nos. LQ16F01001, LQ14F030007, LY16F010016), and Scientific Research Project of Education Department of Zhejiang Province of China (grant nos. Y201327231, Y201430529).

Author details

Xiukai Ruan*, Yanhua Tan, Yuxing Dai, Guihua Cui, Xiaojing Shi, Qibo Cai, Chang Li, Han Li, Yaoju Zhang and Dameng Dai

*Address all correspondence to: ruanxiukai@wzu.edu.cn

College of Physics and Electronic Information Engineering, Wenzhou University, Wenzhou, Zhejiang, China

References

[1] Robet, S., Lutz, L., & Wolfgang H.G. (2001). Noncoherent sequence estimation combined with noncoherent adaptive channel estimation. http://www.lit.lnt.de/papers/ew00cr.pdf.

[2] Pinchas, M., & Bobrovsky B.Z. (2007) A novel HOS approach for blind channel equalization. *IEEE Transactions on Wireless Communications*, 6(3): 875–886.

[3] Abra,r S., & Nandi, A.K. (2010) Blind equalization of square-QAM signal: a multimodulus approach. *IEEE Transactions on Communications*, 58(6): 1674–1685.

[4] Ruan, X. K, Jiang, X, and Li, C. (2012). A novel method of Bussgang-type blind equalization in high-order QAM systems. *Journal of Electronics and Information Technology*, 34(8): 2018–2022.

[5] Ruan, X. K, & Zhang, Y. J. (2014). Blind sequence estimation of MPSK signals using dynamically driven recurrent neural networks. *Neurocomputing*, 129: 421–427.

[6] Ruan, X. K, & Zhang, Y. J. (2012). Blind optical baseband signals detection using recurrent neural network based on continuous multi-valued neurons. *Acta Optica Sinica*, 32(11): 1106001–1106011.

[7] Ruan, X. K, and Zhang, Z. Y. (2011). Blind detection of QAM signals using continuous Hopfield-type neural network. *Journal of Electronics and Information Technology*, 33(7): 1600–1605.

[8] Castillo, E. (1998). Functional networks. *Neural Processing Letters*, 7: 151–159.

[9] Rajasekaran S.(2004). Functional networks in structural engineering. *Journal of Computing in Civil Engineering*, 18(2): 172–181.

[10] Castillo, E., Gutiérrez, J.M., Hadi, A. S., & Lacruz, B. (2001). Some applications of functional networks in statistics and engineering. *Technometrics*. 43(1): 10–24.

[11] Iglesias, A., Arcay, B., Cotos, J.M., Taboada, J.A., & Dafonte, C. (2004). A comparison between functional networks and artificial neural networks for the prediction of fishing catches. *Neural Computing and Applications*, 13: 21–31.

[12] Zhou, Y., He, D., & Nong, Z. (2007). Application of functional network to solving classification problems. *World Academy of Science, Engineering and Technology*, 12: 12–24.

[13] Emad, A.E., Asparouhov O., Abdulraheemd A.A., et al. (2012). Functional networks as a new data mining predictive paradigm to predict permeability in a carbonate reservoir. *Journal Expert Systems with Application: An International Journal*. 39(12): 10359–10375.

[14] Alonso-Betanzos, A., Castillo, E., Fontenla-Romero, O., & Sáchez-Maron˜o, N. (2004). Sheer strength prediction using dimensional analysis and functional networks. *ESANN'2004 Proceedings—European Symposium on Artificial Neural Networks Bruges (Belgium)*: 251–256.

[15] Liao, L., Khong, A. W. H.(2013). An adaptive subsystems based algorithm for channel equalization in a SIMO system. *IEEE Transactions on Circuits and Systems. Part I: Regular Papers*. 60(6): 1559–1569.

[16] Fang, J., Leyman A. R., Chew Y. H. (2006). Blind equalization of SIMO FIR channels driven by colored signals with unknown statistics. *IEEE Transactions on Signal Processing*. 54(6): 1998–2008.

[17] Franklin J. N.(2000). Matrix theory. *Dover Publications*. Chapter 4.

[18] Shen, J., Ding, Z. (2000). Direct blind MMSE channel equalization based on second-order statistics. *IEEE Transactions on Signal Processing*, 48(4): 1015–1022.