

Research Article

The Control Packet Collision Avoidance Algorithm for the Underwater Multichannel MAC Protocols via Time-Frequency Masking

Yang Yu, Jie Shi, Ke He, and Peng Han

School of Marine Science and Technology, Northwestern Polytechnical University, Xi'an 710072, China

Correspondence should be addressed to Yang Yu; nwpuyuy@nwpu.edu.cn

Received 24 December 2015; Revised 28 April 2016; Accepted 9 May 2016

Academic Editor: Paolo Renna

Copyright © 2016 Yang Yu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Establishing high-speed and reliable underwater acoustic networks among multiunmanned underwater vehicles (UUVs) is basic to realize cooperative and intelligent control among different UUVs. Nevertheless, different from terrestrial network, the propagation speed of the underwater acoustic network is 1500 m/s, which makes the design of the underwater acoustic network MAC protocols a big challenge. In accordance with multichannel MAC protocols, data packets and control packets are transferred through different channels, which lowers the adverse effect of acoustic network and gradually becomes the popular issues of underwater acoustic networks MAC protocol research. In this paper, we proposed a control packet collision avoidance algorithm utilizing time-frequency masking to deal with the control packets collision in the control channel. This algorithm is based on the scarcity of the noncoherent underwater acoustic communication signals, which regards collision avoiding as separation of the mixtures of communication signals from different nodes. We first measure the W-Disjoint Orthogonality of the MFSK signals and the simulation result demonstrates that there exists time-frequency mask which can separate the source signals from the mixture of the communication signals. Then we present a pairwise hydrophones separation system based on deep networks and the location information of the nodes. Consequently, the time-frequency mask can be estimated.

1. Introduction

Underwater acoustic networks are the key technology to realize cooperative and intelligent control among multi-UUVs [1, 2]. However, compared with terrestrial wireless network, the propagation velocity of underwater acoustic networks is only 1500 m/s and the available bandwidth of underwater acoustic networks is very limited. Moreover, the time delay, Doppler extension, and noise interference cannot be avoided either. The adverse factors mentioned above make the key technology of underwater acoustic network communication MAC protocol design big challenges and restrict the improvements of underwater acoustic network. Presently, most researchers classify underwater acoustic network MAC protocols into 3 types: contention-free, contention-based, and hybrid based on the difference of multiuser access mechanism. Figure 1 has illustrated the existing underwater acoustic network MAC protocols and its categorizing [3, 4].

Considering the long-time delay of underwater acoustic channel, maintaining the real-time state between adjacent nodes is different. Concise contention-free protocol is firstly used in the underwater acoustic network, which includes frequency division multiple access (FDMA), time division multiple access (TDMA), and code division multiple access (CDMA). FDMA separates available frequency band into different subbands and allocates the subband specifically to the node, which is simple and reliable. However, the low bandwidth availability ratio is a big disadvantage [3–5]. In order to solve the problem, orthogonal frequency division multiplexing (OFDM) has been introduced to FDMA. The principle of FDMA is choosing proper frequency band to communicate according to different communication distances and therefore improves the bandwidth availability ratio [6, 7]. To improve channel utilization fundamentally, researchers begin to study the MAC protocol based on TDMA. The ST-MAC protocol, which translates the issue of

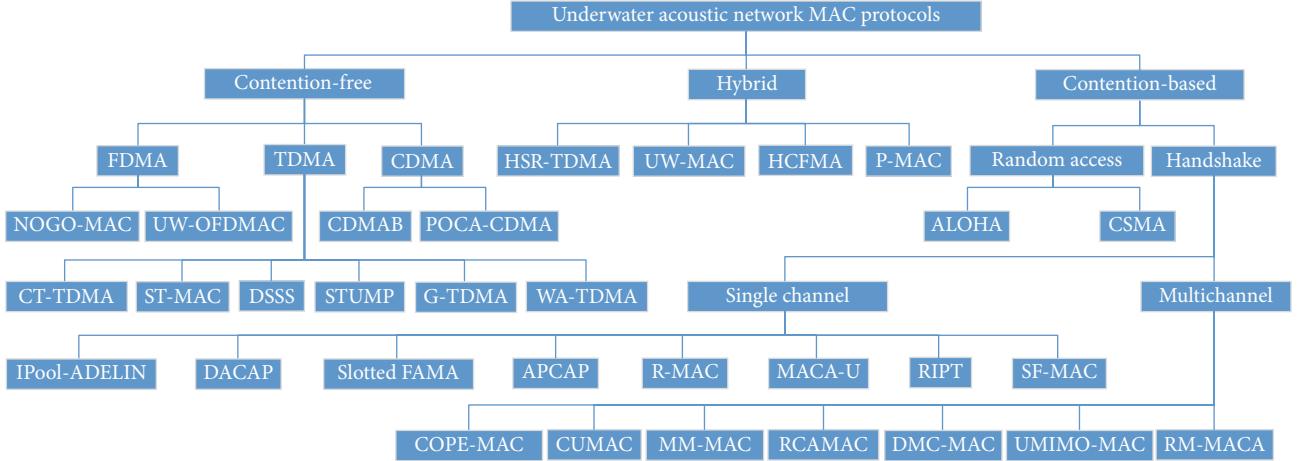


FIGURE 1: Most researchers classify underwater acoustic network MAC protocols into 3 types: contention-free, contention-based, and hybrid.

multiple joints time slot to the problem of vertex-coloring, was proposed in [8]. Reference [9] puts forward DSSS protocol, which utilizes transmission delay of underwater acoustic channel and arranges conflict-free transmission concurrently. The STUMP protocol which eases the limitation of time synchronization was present in [10]. Different from TDMA, CDMA protocols distinguish different users through pseudonoise, which has high channel utilization and simple algorithm but still cannot avoid its inherent “near-far effect.”

Compared with the above-mentioned contention-free protocol this allocates channel beforehand and contention-based protocol which allocates channel based on the need of nodes has higher channel utilization. The essence of contention-based protocol is based on channel reservation. Nodes reserve channel resources through handshaking exchange message before initiating data communication [11, 12]. The multichannel MAC protocols transmit handshaking exchange message via independent channels, initiates information transmission among multinode pairs concurrently, utilizes network bandwidth and reduces the consumption when network load is heavy [13–15], and attracts attention of the researchers recently. The prospective of the multichannel MAC protocols is solving new problems which are faced with multichannel protocols, especially the collision problem in control channel. Zhou et al. from University of Connecticut adopt joint detection of adjacent nodes to tackle triple hidden terminal problems typical in multichannel MAC protocols [16].

In this paper, we proposed a control packet collision avoidance algorithm utilizing time-frequency masking to deal with the control packets collision in the control channel. This algorithm is based on the scarcity of incoherent underwater acoustic communication signals and regards collision avoiding as the separation of the mixtures of communication signals from different nodes. The remaining contents of the paper are organized as follows. Section 2 briefly discusses the W-Disjoint Orthogonality and the scarcity of the MFSK signal. The simulation result demonstrates that there exists time-frequency mask which can separate the source signals

from the mixture of the communication signals. Section 3 outlines the proposed separation system, discusses the low-level feature we used, and gives details about the deep network including its structure as well as training method. Section 4 shows the simulation result about the source separation system in different conditions, including different signal-noise ratio and different bandwidth ratio.

2. W-Disjoint Orthogonality of the MFSK Signals

As is known to all, the MFSK is a classic noncoherent communication modulation scheme that has been considered as a robust modulation to the complex underwater acoustic channel. Because of its lower bandwidth efficiency than the coherent modulation, such as PSK modulation, the MFSK modulation is not considered as a good choice for the physical layer of the underwater acoustic networks. However, the lower bandwidth efficiency means that the MFSK signal is sparse in time and frequency domain. Same as the speech signals mentioned in [17], the MFSK mixtures can be separated into several sources by using time-frequency masking. The received signal can be seen as MFSK mixture when the control packets collide, and the sparsity of the MFSK signal in time and frequency domain offers the potential for dealing with the collision of the control packets.

In this section, we focus on the W-Disjoint Orthogonality and the sparsity of the MFSK signals, showing that there exist time-frequency mask which can separate the source signals from the mixture of the MFSK signals. We only consider the MFSK modulation as the physical layer of the underwater acoustic networks in this paper.

Same as the model of speech mixture, the model of the MFSK mixture can be written as follows:

$$x(t) = \sum_{i=1}^N s_i. \quad (1)$$

With the short-time Fourier transform, we obtain the model of the MFSK in time and frequency domain through

$$X(m, f) = \sum_{i=1}^N S_i(m, f), \quad (2)$$

where $m = 1, \dots, M$, $f = 1, \dots, F$ are the indexes of time frame and frequency point, respectively. Assuming $X(m, f)$ is W-Disjoint Orthogonality, at least one of the N nodes signals will be nonzero for a given (m, f) . To separate node signal from the mixture $X(m, f)$, we create the time-frequency mask for each node, respectively, and apply these masks to the mixture to obtain the original node signal. For instance, with the defining mask \mathcal{M}_i for the node i ,

$$\mathcal{M}_i = \begin{cases} 1 & S_k(m, f) \neq 0 \\ 0 & \text{otherwise,} \end{cases} \quad (3)$$

which actually is an indicator function, the MFSK signal from node k will be derived via

$$\widehat{S}_i(m, f) = \mathcal{M}_i \times X(m, f) \quad \forall m, f. \quad (4)$$

However, for the MFSK signal, the assumption about the W-Disjoint Orthogonality is not strictly satisfied. When the sparsity of the MFSK signals in time and frequency domain is taken into consideration, the approximate W-Disjoint Orthogonality will be satisfied. In order to measure the W-Disjoint Orthogonality of the T-F mask, the combined performance criteria PSR and SIR, which are proposed by Yilmaz and Rickard in [17], will be used:

$$\begin{aligned} \text{PSR}_i &= \frac{\|\mathcal{M}_i S_i(m, f)\|^2}{\|S_i(m, f)\|^2}, \\ \text{SIR}_i &= \frac{\|\mathcal{M}_i S_i(m, f)\|^2}{\|\mathcal{M}_i \sum_{i=1, i \neq i}^N S_i(m, f)\|^2}, \\ \text{WDO}_i &= \frac{\|\mathcal{M}_i S_i(m, f)\|^2 - \|\mathcal{M}_i \sum_{i=1, i \neq i}^N S_i(m, f)\|^2}{\|S_i(m, f)\|^2} \\ &= \text{PSR}_i - \frac{\text{PSR}_i}{\text{SIR}_i}. \end{aligned} \quad (5)$$

With a view to the quite small probability of the collision of more than two data packets in control channel, we have produced a series of MFSK mixture only including two node signals and calculated the PSR, SIR, and WDO of the T-F Mask by the use of Monte Carlo method. Thereinto, the T-T Mask for source separation is derived as follows:

$$\mathcal{M}_i = \begin{cases} 1 & 20 \log \frac{|S_i(m, f)|}{|X(m, f)|} \geq -6 \text{ dB} \\ 0 & \text{otherwise,} \end{cases} \quad (6)$$

$$\text{where } X(m, f) = \sum_{i=1}^N S_i(m, f), \quad N = 2.$$

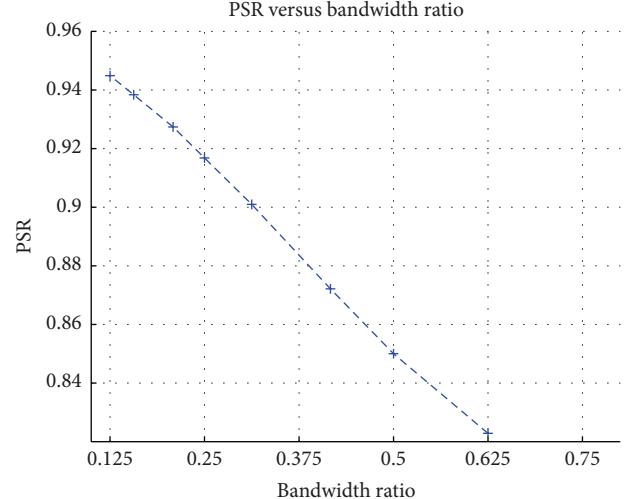


FIGURE 2: The PSR result.

According to the definition of the W-Disjoint Orthogonality, the corresponding T-F mask becomes closer to the W-Disjoint Orthogonality as the signal being more sparse. The sparsity of MFSK signal is reflected by bandwidth ratio, and the lower the bandwidth ratio of the signal is, the higher its sparsity becomes.

By the conclusion of [17], it could be thought that mixed signal is able to be demixed through a T-F mask when the value of WDO is close to 1. In accordance with the simulation result shown on Figures 2, 3, and 4, we believe that an existing T-F mask could separate sources from the MFSK mixture with high quality when the bandwidth ratio is greater than 0.5.

3. The Source Separation System Utilizes Deep Networks

In this section we will outline the proposed separation system, discuss the low-level features we used, and give details about the deep networks including its structure and training method.

3.1. Observation in Time and Frequency Domain and Low-Level Features Used in Deep Networks. We assume that there are K hydrophones in the node and the mixture includes several MFSK signals from N nodes. The mixture received by the hydrophone k can be obtained from

$$x_k(t) = \sum_{i=1}^K s_i * h_{ik} + n_k(t), \quad (7)$$

where h_{ik} is the channel impulse response between the hydrophone k and node i , and the $*$ denotes convolution. Then by using the short-time Fourier transform (STFT), we can obtain the mixture signal mapped into the time-frequency domain from

$$X_k(m, f) = \sum_{i=1}^J S_i(m, f) H_{ki}(f) + N_k(m, f), \quad (8)$$

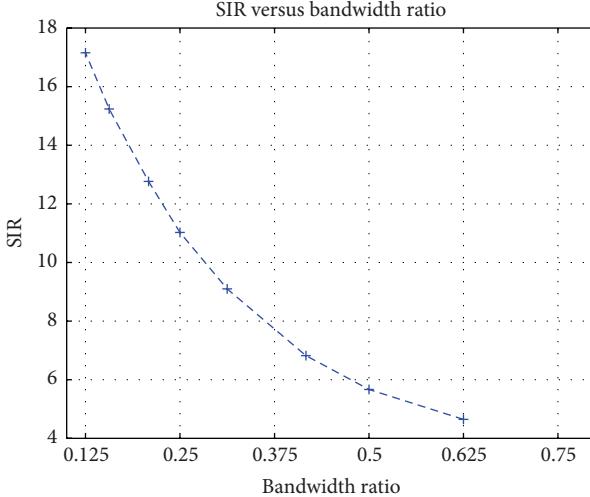


FIGURE 3: The SIR result.

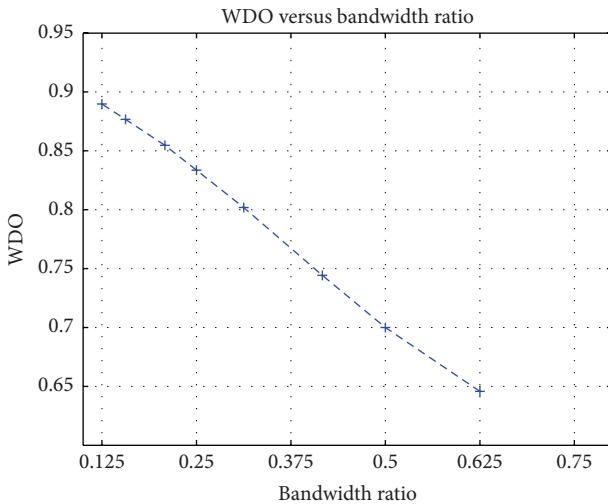


FIGURE 4: The WDO result.

where $m = 1, \dots, M$ and $f = 1, \dots, F$ are the time frame and frequency bin indices, respectively, and $X_k(m, f) = \mathcal{F}(x_k)$, $S_i = \mathcal{F}(s_i)$, $N_k = \mathcal{F}(n_k)$, $\mathcal{F}(\cdot)$ denote short-time Fourier transform. We use $X_{klj}(m, f)$ representing the component of the node i which takes in the mixture received by hydrophone k . Thus,

$$X_{klj}(m, f) = S_i(m, f) \cdot H_{ki}(f). \quad (9)$$

As shown in Section 2 MFSK signal is approximate W-Disjoint Orthogonality when bandwidth ratio is less than 0.5. Then, as shown in (10), the mixture X_k which received by the hydrophone k , can be demixed by using the T-F masks \mathcal{M}_i corresponding node i , $i = 1 \dots N$. Consider the following:

$$\widehat{X}_k(m, f) = \sum_{i=1}^N \mathcal{M}_i \cdot X_k(m, f) + N_k(m, f). \quad (10)$$

There is a nature choice that uses the orientation of the MFSK signals to estimate the corresponding T-F mask of

the nodes with the pairwise hydrophone. Thus, the time-frequency mask is related to the location information of the current input signals (the channel impulse response and array manifold). That is,

$$\begin{aligned} \mathcal{M}_i(m, f) \\ = P(X_i(m, f) \in \text{node } i | S_1 \dots S_N, H_{1k} \dots H_{Nk}). \end{aligned} \quad (11)$$

Obviously, for other hydrophone output, the time-frequency mask \mathcal{M} corresponding to the same node remains the same. When we obtain the time-frequency \mathcal{M}_i corresponding to node i , we can then separate the single user communication signals of node i through ISTFT operation. The mixture signals received by the multiarray include N single user signals from N nodes with different locations. The probability distribution of a single T-F point belonging to a certain node, such as i , can be described by a Gaussian distribution with mean μ_i and variance σ_i . Here, μ_i and variance σ_i can be interpreted as the mean value and variance of direction of arrival (DOA) of the signal coming from the node i , respectively. According to the central limit theorems, for all T-F points, the probability distribution of mixture signals including N node signals can be described by the mixture of N Gaussian distribution, namely, Gaussian mixture model (GMM). Therefore, we can describe the mixture pattern of signals through a GMM based on space features. The parameters of the GMM can be obtained on the basis of the existing observation datasets. Thus the probability of each T-F point belonging to node 1, node 2, ..., and node N can be estimated. Then, the source signals can be recovered through ISTFT operation.

3.2. Outline of the Source Separation System. As shown in Figure 5, the inputs to the system are the two channel MFSK mixtures. We perform short-time Fourier transform (STFT) for each channel and obtain the T-F representation of the input signals, $X_A(m, f)$ and $X_B(m, f)$ where $m = 1, \dots, M$ and $f = 1, \dots, F$ are the time frame and frequency bin indices, respectively. The low-level features, that is, mixing vector (MV), interaural level, and phase difference (IPD/ILD) which can be derived from (12)–(13), are then estimated at each T-F unit. Consider

$$\begin{aligned} \mathbf{z}(m, f) &= \frac{\mathbf{W}(f)\tilde{\mathbf{x}}(m, f)}{\|\mathbf{W}(f)\tilde{\mathbf{x}}(m, f)\|}, \\ \tilde{\mathbf{x}}(m, f) &= \frac{[X_A(m, f), X_B(m, f)]^T}{\|[X_A(m, f), X_B(m, f)]^T\|}, \end{aligned} \quad (12)$$

where $\mathbf{W}(f)$ is a whitening matrix, with each row being one eigenvector of $E(\tilde{\mathbf{x}}(m, f)\tilde{\mathbf{x}}^H(m, f))$, the superscript H is Hermitian transpose, and $\|\cdot\|$ is Frobenius norm:

$$\begin{aligned} \alpha(m, f) &= 20 \log_{10} \left(\left| \frac{X_A(m, f)}{X_B(m, f)} \right| \right), \\ \phi(m, f) &= \angle \left(\frac{X_A(m, f)}{X_B(m, f)} \right), \end{aligned} \quad (13)$$

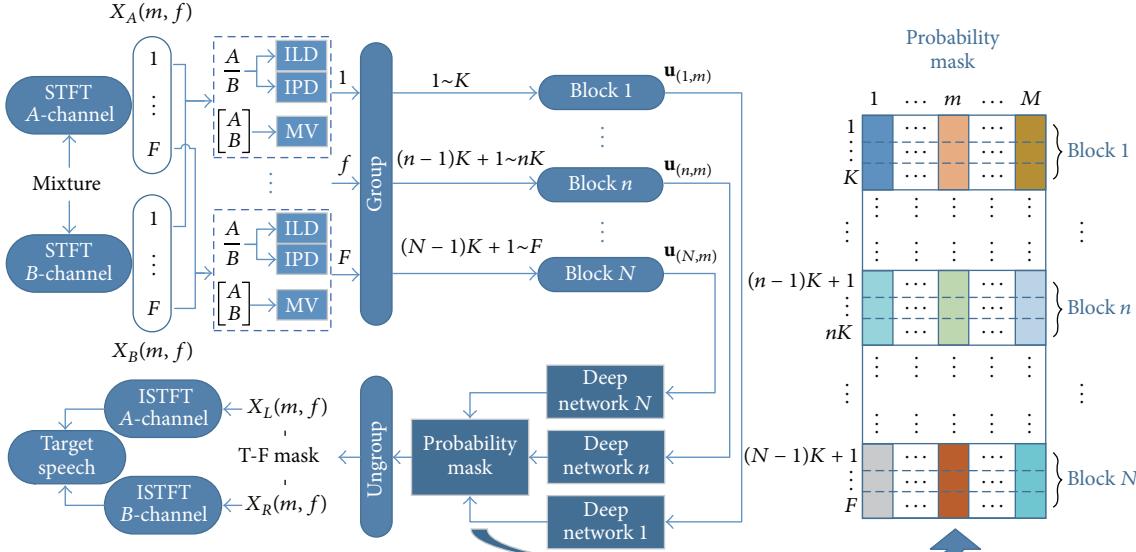


FIGURE 5: The architecture of the proposed system using deep neural network based time-frequency masking for blind source separation.

where $|\cdot|$ takes the absolute value of its argument and $\angle(\cdot)$ finds the phase angle.

Next, we group the low-level features into N blocks (only along the frequency bins f). The block n includes K frequency bins ($(n-1)K + 1, \dots, nK$), where $K = F/N$. We build N deep networks with each corresponding to one block and use them to estimate the direction of arrivals (DOAs) of the sources. The low-level features as the input of the deep networks are composed by the IPD, ILD and MV; that is, $\bar{\mathbf{u}}(m, f) = [\mathbf{z}^T(m, f), \alpha(m, f), \phi(m, f)]^T$. Through unsupervised learning and the sparse autoencoder [18] in deep networks, high-level features (coded positional information of the sources) are extracted and used as inputs for the output layer (i.e., the softmax regression) of the networks. The output of softmax regression is a source occupation probability (i.e., the time-frequency mask) of each block (through the ungroup operation, T-F units in the same block are assigned with the same source occupation probability) of the mixtures. Then the sources can be recovered applying the T-F mask to the mixtures followed by the inverse STFT (ISTFT). The deep networks are pretrained by using a greedy layer-wise training method.

3.3. The Deep Networks. As described in the beginning of this section, we group the low-level features into N blocks and build N individual deep networks which have the same architecture to classify the DOAs of the current input T-F point in each block. The deep network which is used to estimate the T-F mask is composed of two-layer deep autoencoder and one layer softmax classifier.

More specifically, we split the whole space to J ranges with respect to the hydrophones and separate the target and interferers based on different orientation ranges (DOAs with respect to the receiver node) where they are located. We apply the softmax classifier to perform the classification task, and the inputs to the classifier, that is, the high-level features $\mathbf{a}^{(2)}$,

are produced by the deep autoencoder. Assuming that the position of the target in the current input T-F point remains unchanged, the deep network estimates the probability $p(g_i = i \mid \mathbf{u}_{(m,f)})$ of the orientation of the current input sample belonging to the orientation index i . With the estimated orientation (obtained by selecting the maximum probability index) of each input T-F point, we cluster the T-F points which have the same orientation index to get the probability mask and obtain the T-F mask from the probability mask through the ungroup operation. Note that each T-F point in the same block is assigned the same probability. The number of sources can also be estimated from the probability mask by using a predefined probability threshold, typically chosen as 0.2 in our experiments.

3.3.1. Deep Autoencoder. An autoencoder is an unsupervised learning algorithm based on backpropagation. It aims to learn an approximation $\hat{\mathbf{u}}$ of the input \mathbf{u} . It appears to be learning a trivial identity function; but by using some constraints on the learning process, such as limiting the number of neurons activated, it discloses some interesting structures about the data. Figure 6 shows the architecture of a single layer autoencoder. The difference between classic neural network and autoencoder is the training objective. The objective of the classic neural networks is to minimize the difference between the label of input training data and the output of the network. However, the objective of the autoencoder is to minimize the difference between the input training dataset and the output of the network. As shown in Figure 6, the output of the autoencoders can be defined as $\hat{\mathbf{u}} = \text{sigm}(\widehat{\mathbf{W}}^{(2)} \mathbf{a}^{(1)} + \widehat{\mathbf{b}}^{(2)})$ with $\mathbf{a}^{(1)} = \text{sigm}(\mathbf{W}^{(1)} \mathbf{u} + \mathbf{b}^{(1)})$, where the function $\text{sigm}(\mathbf{u}) = 1/(1 + \exp(-\mathbf{u}))$ is the logistic function, $\mathbf{W}^{(1)} \in \mathbb{R}^{V \times Y}$, $\mathbf{b}^{(1)} \in \mathbb{R}^V$, $\widehat{\mathbf{W}}^{(2)} \in \mathbb{R}^{Y \times V}$, and $\widehat{\mathbf{b}}^{(2)} \in \mathbb{R}^Y$, V is the number of hidden layer neurons, and Y is the number of input layer neurons, which is the same as

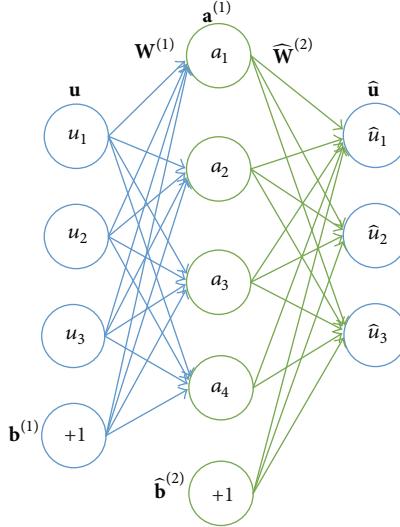


FIGURE 6: The single layer autoencoder.

that of the output layer neurons. $\mathbf{W}^{(1)}$ is a matrix containing the weights of connections between the input layer neurons and hidden layer neurons. Similar to $\mathbf{W}^{(1)}$, $\widehat{\mathbf{W}}^{(2)}$ contains the weights of connections between the hidden layer neurons and the output layer neurons. $\mathbf{b}^{(1)}$ is a vector of the bias values added to the hidden layer neurons, and $\mathbf{b}^{(2)}$ is the vector for the output layer neurons. Θ refers to the parameter set composed of weights \mathbf{W} and bias \mathbf{b} . The neuron v is “active” when the output a_v of this neuron is close to 1, which means that the function $\text{sigm}(\mathbf{W}_v^{(1)} \mathbf{u} + \mathbf{b}_v^{(1)}) \approx 1$. For “inactive” neurons, however, the output is close to 0, which means the function $\text{sigm}(\mathbf{W}_v^{(1)} \mathbf{u} + \mathbf{b}_v^{(1)}) \approx 0$, where $\mathbf{W}_v^{(1)}$ denotes the weights of connections between the hidden layer neuron v and the input layer neurons, which is the v th row of the matrix $\mathbf{W}^{(1)}$. $\mathbf{b}_v^{(1)}$ is the v th element of the vector $\mathbf{b}^{(1)}$, which is the bias value added to the hidden layer neuron v . The superscript i of $\mathbf{b}^{(i)}$, $\mathbf{W}^{(i)}$, and $\mathbf{a}^{(i)}$, denotes the i th layer of the deep network.

With the sparsity constraint, most of the neurons are assumed to be inactive. More specifically, $a_v = \text{sigm}(\mathbf{W}_v^{(1)} \mathbf{u} + \mathbf{b}_v^{(1)})$ denotes the activation value of the hidden layer unit v in the autoencoder. Generalizing this for the unit i in the hidden layer, the average activation $\hat{\rho}_v$ of unit v with the input sample $\mathbf{u}_{(m)}$ can be defined as follows:

$$\hat{\rho}_v = \frac{1}{M} \sum_{m=1}^M a_v(\mathbf{u}_{(m)}), \quad (14)$$

where M is the number of training samples, and $\mathbf{u}_{(m)}$ is the m th input training sample. Next, the sparsity constraint $\hat{\rho}_v = \rho$ is enforced, where ρ is the parameter preset before training, typically small, such as $\rho = 3 \times 10^{-3}$. To achieve the sparsity constraint, we use the penalty term in the cost function of sparse autoencoders as follows:

$$\sum_{v=1}^V \text{KL}(\rho \parallel \hat{\rho}_v) = \sum_{v=1}^V \rho \log \frac{\rho}{\hat{\rho}_v} + (1 - \rho) \log \frac{1 - \rho}{1 - \hat{\rho}_v}. \quad (15)$$

The penalty term is essentially a Kullback-Leibler (KL) divergence. Now the cost function $J_{\text{sparse}}(\mathbf{W}, \mathbf{b})$ of the sparse autoencoder can be written as follows:

$$J_{\text{sparse}}(\mathbf{W}, \mathbf{b}) = \frac{1}{2} \|\mathbf{u} - \hat{\mathbf{u}}\|^2 + \beta \sum_{v=1}^V \text{KL}(\rho \parallel \hat{\rho}_v), \quad (16)$$

where β controls the weight of the penalty term. In our proposed system, the cost function $J_{\text{sparse}}(\mathbf{W}, \mathbf{b})$ is minimized using the limited memory BFGS (L-BFGS) optimization algorithm, and the single layer sparse autoencoder is trained by using the backpropagation algorithm.

After finishing the training of single layer sparse autoencoder, we discard the output layer neurons, the relative weights $\widehat{\mathbf{W}}^{(2)}$, and bias $\widehat{\mathbf{b}}^{(2)}$ and only save the input layer neurons $\mathbf{W}^{(1)}$ and $\mathbf{b}^{(1)}$. The output of the hidden layer $\mathbf{a}^{(1)}$ is used as the input samples of the next single layer sparse autoencoder. Repeating these steps, like stacking the autoencoders, we could build a deep autoencoder from two or more single layer sparse autoencoders. In our proposed system, we use two single layer autoencoders to build a deep autoencoder. The stacking procedures show on the right part of Figure 7.

Lots of studies on deep autoencoders show that, with the deep architecture (more than one hidden layer), deep autoencoder could build up more complex representation from the sample low-level features, capture the underlying regularities of the data, and improve the qualities of recognition. That is why we use deep autoencoder in our proposed system.

There are, however, several difficulties associated with obtaining the optimized weights of deep autoencoders. One challenge is the presence of local optima. In particular, training a neural network using supervised learning involves solving a highly nonconvex optimization problem, that is, finding a set of network parameters (\mathbf{W}, \mathbf{b}) to minimize the training error $\|\mathbf{u} - \hat{\mathbf{u}}\|^2$. In the deep autoencoder, the optimization problem with bad local optima turns out to be rife, and training with gradient descent no longer works well. Another challenge is the “diffusion of gradients.” When using backpropagation to compute the derivatives, the gradients that are propagated backwards (from the output layer to the earlier layers of the network) rapidly diminish in magnitude as the depth of the network increases. As a result, the derivative of the overall cost with respect to the weights in the earlier layers is very small. Thus, when using gradient descent, the weights of the earlier layers change slowly. However, if the initializing parameter is already close to the optimized values, the gradient descent works well. That is the idea of “greedy layer-wise” training, where the layers of the networks are trained one by one, as shown in the left part of Figure 7.

First, we use the backpropagation algorithm to train the first sparse autoencoder (only including one hidden layer), with the data label being the inputs. In our proposed system, the input data is the $4K$ -dimensional feature vector \mathbf{u} . As a result of the first-layer training, we get a set of network parameters $\Theta^{(1)}$ (i.e., the parameters $\mathbf{W}^{(1)}$ and $\mathbf{b}^{(1)}$) of the first-layer sparse autoencoder, and a new dataset $\mathbf{a}^{(1)}$ (i.e., the features I shown in Figure 9) which is the output of the hidden

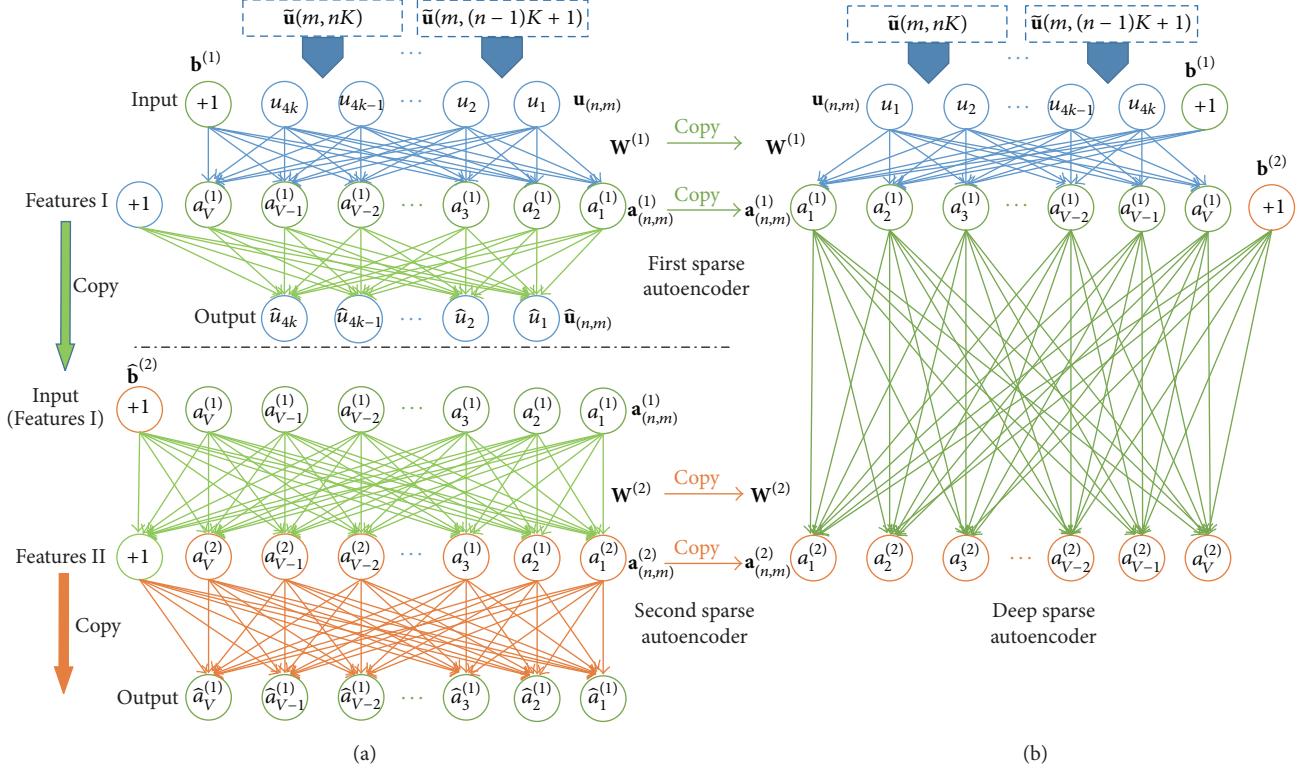


FIGURE 7: The illustration of greedy layer-wise training and stacking. (a) is the procedure of greedy layer-wise training and (b) is the procedure of stacking of sparse autoencoders.

layer neurons (the activity state of the hidden layer neurons) by using this parameter set. Next, we use $\mathbf{a}^{(1)}$ as the inputs to the second sparse autoencoder. After the second autoencoder is trained, we can get the network parameters $\Theta^{(2)}$ of the second sparse autoencoder and the new dataset $\mathbf{a}^{(1)}$ (i.e., the feature II in Figure 9) for the training of next single layer neural network. We then repeat the above steps until the last layer (i.e., the softmax regression in our proposed system). Finally, we obtain a pretrained deep autoencoder by stacking all the autoencoders and use $\Theta^{(1)}$ and $\Theta^{(2)}$ as the initialized parameters for this pretrained deep autoencoder. The feature II is the high-level feature and can be used as the training dataset for softmax regression discussed next.

3.3.2. Softmax Classifier. In our proposed system, the softmax classifier, based on softmax regression, was used to estimate the probabilities of the current input T-F point $\mathbf{u}_{(n,m)}$ belonging to the orientation index j , by the deep autoencoder with the extracted high-level features $\mathbf{a}_{(n,m)}^{(2)}$ as inputs.

The softmax regression generalizes the classical logistic regression (for binary classification) to multiclass classification problems. Different from the logistic regression, the data label of the softmax regression is an integer value between 1 and J ; here J is the number of data classes. More specifically, in our proposed system, for J classes, M samples dataset was used to train the n th deep network:

$$\left\{ \left(\mathbf{a}_{(n,1)}^{(2)}, g_{(n,1)} \right), \dots, \left(\mathbf{a}_{(n,M)}^{(2)}, g_{(n,M)} \right), \dots, \left(\mathbf{a}_{(N,M)}^{(2)}, g_{(N,M)} \right) \right\}, \quad (17)$$

where $g_{(n,m)}$ is the label of the m th sample $\mathbf{a}_{(n,m)}^{(2)}$ and will be set to j if $\mathbf{a}_{(n,m)}^{(2)}$ belongs to class j . The architecture of the softmax classifier is shown in Figure 8.

Given an input $\mathbf{a}_{(n,m)}^{(2)}$, the output $p(g_{(n,m)} = j | \mathbf{a}_{(n,m)}^{(2)}) = e^{\theta_j^T \mathbf{a}_{(n,m)}^{(2)}} / \sum_{j=1}^J e^{\theta_j^T \mathbf{a}_{(n,m)}^{(2)}}$ of the layer neuron j gives the probability of the input $\mathbf{a}_{(n,m)}^{(2)}$ belonging to the class j . Similar to the logistic regression, the output of the softmax regression can be written as follows:

$$\begin{aligned} \mathbf{h}_\Theta(\mathbf{a}_{(n,m)}^{(2)}) &= \begin{bmatrix} p(g_{(n,m)} = 1 | \mathbf{a}_{(n,m)}^{(2)}; \Theta) \\ \vdots \\ p(g_{(n,m)} = j | \mathbf{a}_{(n,m)}^{(2)}; \Theta) \\ \vdots \\ p(g_{(n,m)} = J | \mathbf{a}_{(n,m)}^{(2)}; \Theta) \end{bmatrix} \\ &= \frac{1}{\sum_{j=1}^J e^{\theta_j^T \mathbf{a}_{(n,m)}^{(2)}}} \begin{bmatrix} e^{\theta_1^T \mathbf{a}_{(n,m)}^{(2)}} \\ \vdots \\ e^{\theta_j^T \mathbf{a}_{(n,m)}^{(2)}} \\ \vdots \\ e^{\theta_J^T \mathbf{a}_{(n,m)}^{(2)}} \end{bmatrix}. \end{aligned} \quad (18)$$

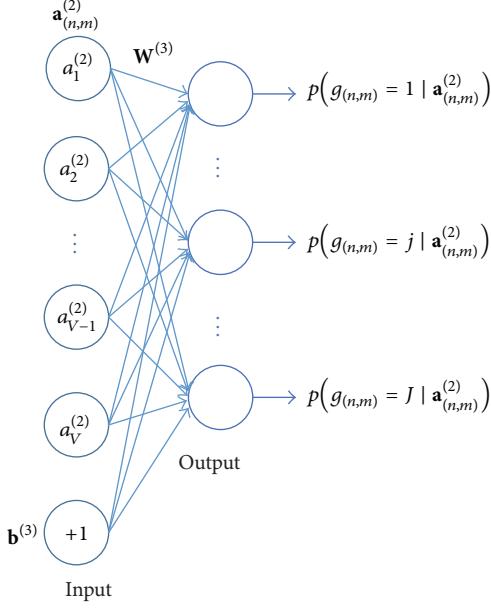


FIGURE 8: A classical architecture of the softmax classifier.

Here $\Theta \in \mathbb{R}^{J \times V}$ is the network parameter of the softmax (V is the dimension of the input $\mathbf{a}_{(n,m)}^{(2)}$, and J is the number of the classes within the input data). Θ_j^T represents the transpose of the j th row of Θ and contains the parameters of the connections between the neuron j of the output layer and the input sample $\mathbf{a}_{(n,m)}^{(2)}$.

The softmax regression and logistic regression have the same minimization objective. The cost function $J_{\text{softmax}}(\Theta)$ of the softmax classifier can be generalized from the logistic regression function and written as follows:

$$J_{\text{softmax}}(\Theta) = -\frac{1}{M} \left[\sum_{m=1}^M \sum_{j=1}^J 1\{\mathcal{G}_{(n,m)} = j\} \cdot \log p(g_{(n,m)} = j | \mathbf{a}_{(n,m)}^{(2)}; \Theta) \right] + \frac{\lambda}{2} \sum_{j=1}^J \sum_{v=1}^V [\Theta_{j,v}]^2, \quad (19)$$

where $1\{\cdot\}$ is the indicator function, so that $1\{\text{a true statement}\} = 1$ and $1\{\text{a false statement}\} = 0$. The element $\Theta_{j,v}$ of the Θ contains the parameters of the connections between the neuron j of the output layer and the neuron v of the input layer. $(\lambda/2) \sum_{j=1}^J \sum_{v=1}^V [\Theta_{j,v}]^2$ is a regularization term that enforces the cost function $J_{\text{softmax}}(\Theta)$ to be strictly convex, where λ is a weight decay parameter predefined by the users.

In our proposed system, we extend the label $g_{(n,m)}$ of the training dataset to a vector $\mathbf{g}_{(n,m)} \in \mathbb{R}^J$ and set the j th element g_j of $\mathbf{g}_{(n,m)}$ to 1 and other elements to 0 when the input sample $\mathbf{a}_{(n,m)}^{(2)}$ belongs to class j , such as $\mathbf{g}_{(n,m)} = [0, \dots, 0, 1, 0, \dots, 0]^T$.

The cost function $J_{\text{softmax}}(\Theta)$ can be written as follows by using vectorization:

$$J_{\text{softmax}}(\Theta) = -\frac{1}{M} \left[\sum_{m=1}^M (\mathbf{g}_{(n,m)})^T \mathbf{h}_\Theta(\mathbf{a}_{(n,m)}^{(2)}) \right] + \frac{\lambda}{2} \sum_{j=1}^J \sum_{i=1}^I (\Theta_{j,i})^2. \quad (20)$$

The softmax classifier can be trained by using the L-BFGS algorithm based on a dataset, in order to find an optimal parameter set Θ for minimizing the cost function $J_{\text{softmax}}(\Theta)$. In our proposed system, the dataset for softmax classifier training is composed by two parts. The first part is the input sample, $\mathbf{a}_{(n,m)}^{(2)}$ (feature II), calculated from the last hidden layer of the deep autoencoder. The second part is the data label, $\mathbf{g}_{(n,m)} \in \mathbb{R}^J$, where the j th element g_j of the $\mathbf{g}_{(n,m)}$ will be set to 1 when the input sample belongs to the source located in the range of DOAs of index j .

We stack the softmax classifier and deep autoencoder together after the training is completed, as shown on the left part of Figure 9. Finally, we use the training dataset and L-BFGS algorithm to fine-tune the deep network with the initialized parameters $\mathbf{W}^{(1)}, \mathbf{b}^{(1)}, \mathbf{W}^{(2)}, \mathbf{b}^{(2)}, \mathbf{W}^{(3)}$, and $\mathbf{b}^{(3)}$ obtained from the sparse autoencoders and softmax classifier training. The training phase of the sparse autoencoders and softmax classifier are called pretraining phase, and the stacking/training of the overall network, that is, deep network, is called fine-tuning phase. In the pretraining phase, the shallow neural networks, that is, sparse autoencoders and softmax classifier, are training individually, using the output of current layer as the input for the next layer. In the fine-tuning phase, we use the L-BFGS algorithm (i.e., a gradient descent method) to minimize the difference between the output of the deep network and the label of training dataset. The gradient descent works well because the initialized parameters obtained from the pretraining phase include a significant amount of “prior” information about the input data through unsupervised learning.

4. Experiments

In this section, we describe the communication system used in the simulation system and show the separation results in different SNR and data rates. It is important to note that bit error rate (BER) is an extremely significant performance index for a pragmatic communication system; in this section, therefore, we will evaluate the separation quality of pragmatic system by BER and regard the BER performance of single user MFSK communication as the baseline under the same condition.

There is no need to take the multipath propagation of channels into consideration because of the relatively close distance among each node, in view of the communication of multi-UUV. We thus adopt the additive white Gaussian noise channel model and Monte Carlo method to calculate the system BER in the simulation, such as the simulation system shown in Figure 10, and we also take the BER of

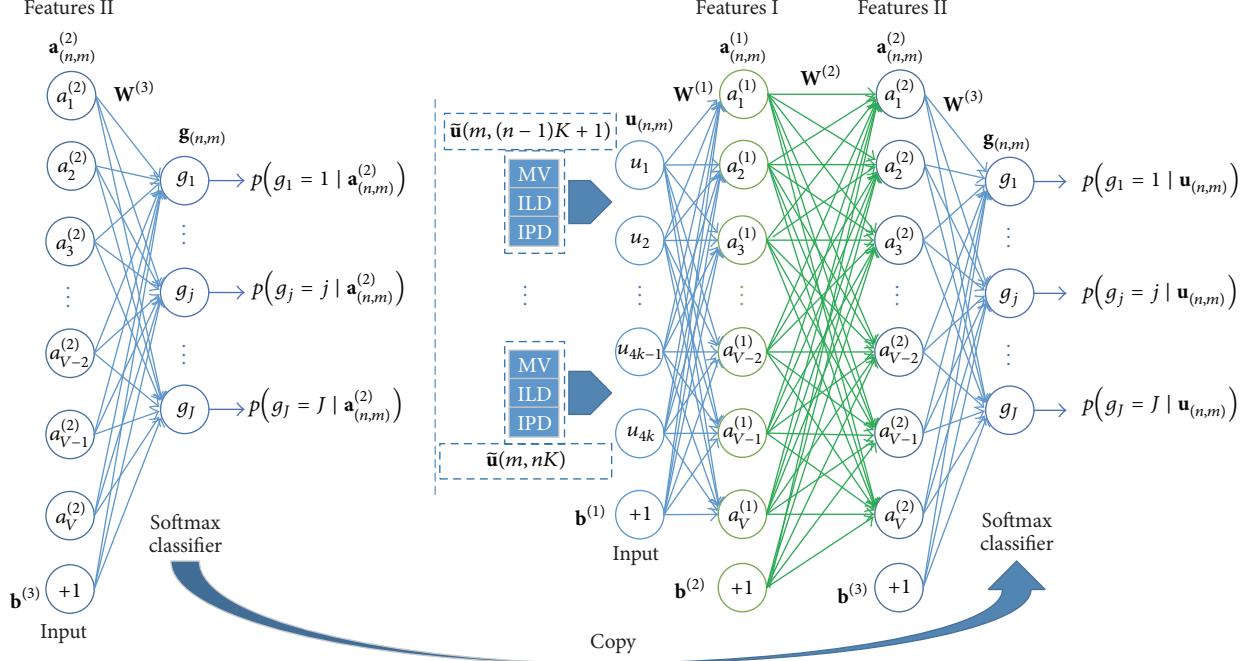


FIGURE 9: The illustration of stacking deep autoencoder and softmax classifier.

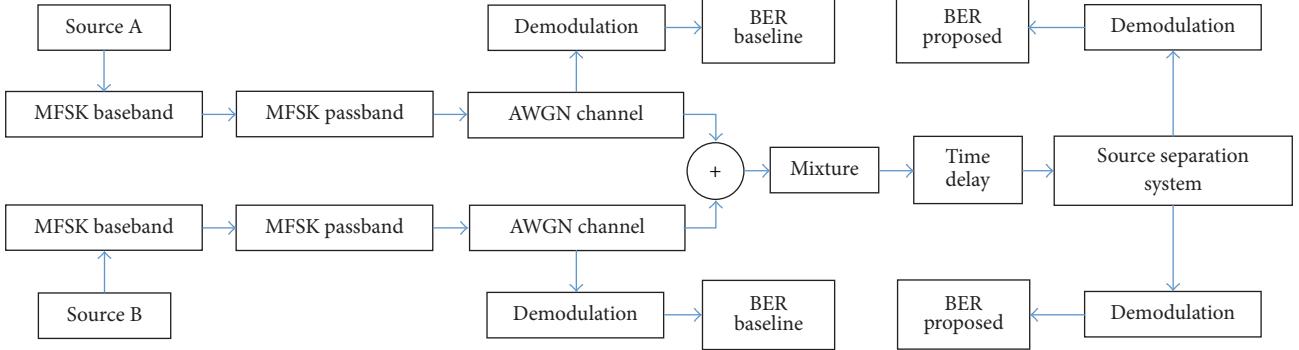


FIGURE 10: The simulation system used in the experiment.

the single user communication as the baseline. The sampling rate of our simulation is 16 kHz, and the order of the modulation is 16, without any error-correction coding. The bandwidth of the MFSK modulation/demodulation is 640 Hz, and the bandwidth ratio varies from 0.625 to 0.125, corresponding to the bit rate 400 bits/s to 80 bits/s.

First, suppose we can obtain the optimal T-F mask, that is, using (6) to calculate T-F mask, and then get the simulation result of BER being changing with SNR as well as bandwidth ratio and make a comparison with baseline.

As shown in Tables 1 and 2, it can be seen that, from the simulation result, the BER similar with baseline can be attained by T-F mask method when bandwidth ratio < 0.417. Furthermore, the BER with T-F mask would be even lower than that of baseline under the same condition when SNR is very low; that is, SNR = -20. The reason behind the above phenomenon is that, it can be seen from (6) that some

frequency points are adjusted to zero by the T-F mask, which promotes the SNR of signal objectively and obtains lower BER. In the actual system, however, it is a highly challenging task to accurately estimate the T-F mask under such low SNR, which can also be verified in later simulation.

According to Section 3, we estimate the T-F mask by using the orientation information of the nodes, that is, the DOA of the MFSK signal received by nodes. Therefore, we introduce time delay τ to simulate pairwise hydrophone receiver in the simulation and divide the space into 37 blocks along the horizontal direction at the same time, in order to correspond from -90 to +90 horizontally with the step size of 5 in a horizontal space. We estimate the T-F mask by source separation system described in Section 3 and compare it to the BER performance of baseline under the condition of different SNR and bandwidth ratio; the simulation result is shown in Table 3.

TABLE 1: The BER of the simulation system with the optimum T-F mask.

Bandwidth ratio	SNR = 30 dB	SNR = 20 dB	SNR = 10 dB	SNR = 5 dB	SNR = 0 dB
0.625	5.30E - 02	5.30E - 02	5.36E - 02	5.50E - 02	5.95E - 02
0.500	2.96E - 02	2.95E - 02	3.03E - 02	3.16E - 02	3.56E - 02
0.417	1.40E - 02	1.38E - 02	1.38E - 02	1.39E - 02	1.39E - 02
0.313	9.00E - 06	1.60E - 05	1.30E - 05	2.00E - 05	4.50E - 05
0.250	3.00E - 06	4.00E - 06	1.00E - 06	4.00E - 06	4.00E - 06
0.208	0.00E + 00	1.00E - 06	1.00E - 06	0.00E + 00	0.00E + 00
0.156	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00
0.125	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00
Bandwidth ratio	SNR = -5 dB	SNR = -10 dB	SNR = -15 dB	SNR = -20 dB	/
0.625	7.64E - 02	1.47E - 01	3.40E - 01	5.62E - 01	/
0.500	4.26E - 02	6.82E - 02	1.98E - 01	4.24E - 01	/
0.417	1.48E - 02	2.67E - 02	1.10E - 01	3.09E - 01	/
0.313	1.25E - 04	3.08E - 03	3.23E - 02	1.57E - 01	/
0.250	9.00E - 06	1.12E - 04	7.13E - 03	7.54E - 02	/
0.208	2.00E - 06	2.19E - 04	4.69E - 03	4.62E - 02	/
0.156	0.00E + 00	2.10E - 05	1.13E - 03	1.46E - 02	/
0.125	0.00E + 00	0.00E + 00	2.00E - 05	4.02E - 03	/

TABLE 2: The BER of the baseline.

Bandwidth ratio	SNR = 30 dB	SNR = 20 dB	SNR = 10 dB	SNR = 5 dB	SNR = 0 dB
0.625	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00
0.500	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00
0.417	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00
0.313	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00
0.250	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00
0.208	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00
0.156	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00
0.125	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00
Bandwidth ratio	SNR = -5 dB	SNR = -10 dB	SNR = -15 dB	SNR = -20 dB	/
0.625	4.43E - 04	5.08E - 02	3.62E - 01	7.26E - 01	/
0.500	1.00E - 06	7.91E - 03	2.16E - 01	6.47E - 01	/
0.417	0.00E + 00	7.13E - 04	1.19E - 01	5.70E - 01	/
0.313	0.00E + 00	0.00E + 00	3.42E - 02	4.35E - 01	/
0.250	0.00E + 00	0.00E + 00	9.65E - 03	3.28E - 01	/
0.208	0.00E + 00	0.00E + 00	3.40E - 03	2.53E - 01	/
0.156	0.00E + 00	0.00E + 00	2.92E - 04	1.41E - 01	/
0.125	0.00E + 00	0.00E + 00	1.90E - 05	7.23E - 02	/

As shown in Table 3, it is observed that the BER performance of the proposed system is much the same as the baseline when $\text{SNR} > 20 \text{ dB}$ and bandwidth ratio < 0.417 , which is consistent with the result when using the optimal T-F mask. When $\text{SNR} < 20 \text{ dB}$, however, the BER performance of the proposed system begins to decline for a big error of T-F mask estimation made by the lower SNR of the signals, which results in the system performance degradation.

5. Summary

In this paper, we point at the problem of control packets collision avoiding existing widely in multichannel MAC protocol

and, on the basis of the sparsity of noncoherent modulation-MFSK in time-frequency domain, separate the sources from the MFSK mixture caused by packets collision through the use of T-F masking method. First, we indicate the sparsity of MFSK signal with bandwidth ratio and demonstrate the relation between bandwidth ratio and PSR, SIR, and WDO by means of the simulation experiment. Then, we establish the source separation system based on deep networks and the model of MFSK communication system, taking single user MFSK communication as the baseline to compare the BER performance of proposed system and baseline under different condition of SNR and bandwidth ratio. The simulation result shows that, first, the optimal T-F masking could obtain

TABLE 3: The BER of the proposed system.

Bandwidth ratio	SNR = 30 dB	SNR = 20 dB	SNR = 10 dB	SNR = 5 dB	SNR = 0 dB
0.625	$5.80E - 02$	$7.30E - 02$	$6.36E - 02$	$6.30E - 02$	$3.59E - 01$
0.500	$3.26E - 02$	$3.45E - 02$	$3.93E - 02$	$4.16E - 02$	$3.36E - 01$
0.417	$1.46E - 02$	$1.78E - 02$	$1.48E - 02$	$4.99E - 02$	$2.14E - 01$
0.313	$1.00E - 05$	$6.89E - 04$	$9.50E - 03$	$3.00E - 02$	$1.90E - 01$
0.250	$7.00E - 06$	$2.10E - 04$	$6.00E - 03$	$1.00E - 02$	$1.60E - 01$
0.208	$1.00E - 06$	$9.00E - 05$	$3.00E - 03$	$8.00E - 03$	$1.30E - 01$
0.156	$0.00E + 00$	$3.00E - 05$	$9.80E - 04$	$2.00E - 03$	$1.30E - 01$
0.125	$0.00E + 00$	$7.00E - 06$	$1.00E - 05$	$9.00E - 04$	$1.20E - 01$
Bandwidth ratio	SNR = -5 dB	SNR = -10 dB	SNR = -15 dB	SNR = -20 dB	/
0.625	$4.94E - 01$	/	/	/	/
0.500	$4.26E - 01$	/	/	/	/
0.417	$4.48E - 01$	/	/	/	/
0.313	$4.25E - 01$	/	/	/	/
0.250	$4.90E - 01$	/	/	/	/
0.208	$4.20E - 01$	/	/	/	/
0.156	$4.00E - 01$	/	/	/	/
0.125	$2.50E - 01$	/	/	/	/

the same BER performance as the baseline under lower bandwidth ratio; second, the proposed system could obtain the similar BER performance as the baseline under higher SNR; third, the BER performance of the proposed system declines rapidly under the condition of lower SNR, for lower SNR leads to a greater error in the estimation of T-F mask. In the future work, we will adjust the structure of deep networks in subsequent research work to promote the performance of proposed system under the condition of low SNR and multipath propagation presenting in the underwater channel. As a future research topic, it also deserves the possibility that the bioinspired computing models and algorithms are used for the underwater multichannel MAC protocols, such as the P systems (inspired from the structure and the functioning of cells) [19, 20], and evolutionary computation (motivated by evolution theory of Darwin) [21, 22].

Competing Interests

The authors declare that there are no competing interests regarding the publication of this paper.

Acknowledgments

This research was supported partially by the Natural Science Basis Research Plan in Shaanxi Province of China (Program no. 2014JQ8355).

References

- [1] Z. Li, C. Yang, N. Ding, S. Bogdan, and T. Ge, "Robust adaptive motion control for underwater remotely operated vehicles with velocity constraints," *International Journal of Control, Automation and Systems*, vol. 10, no. 2, pp. 421–429, 2012.
- [2] A. S. K. Annamalai, R. Sutton, C. Yang, P. Culverhouse, and S. Sharma, "Robust adaptive control of an uninhabited surface vehicle," *Journal of Intelligent & Robotic Systems*, vol. 78, no. 2, pp. 319–338, 2015.
- [3] K. Chen, M. Ma, E. Cheng, F. Yuan, and W. Su, "A survey on MAC protocols for underwater wireless sensor networks," *IEEE Communications Surveys and Tutorials*, vol. 16, no. 3, pp. 1433–1447, 2014.
- [4] R. Otnes, A. Asterjadhi, P. Casari et al., *Underwater Acoustic Networking Techniques*, Springer, Berlin, Germany, 2012.
- [5] S. Climent, A. Sanchez, J. V. Capella, N. Meratnia, and J. J. Serrano, "Underwater acoustic wireless sensor networks: advances and future trends in physical, MAC and routing layers," *Sensors*, vol. 14, no. 1, pp. 795–833, 2014.
- [6] I. M. Khalil, Y. Gadallah, M. Hayajneh, and A. Khreishah, "An adaptive OFDMA-based MAC protocol for underwater acoustic wireless sensor networks," *Sensors*, vol. 12, no. 7, pp. 8782–8805, 2012.
- [7] F. Bouabdallah and R. Boutaba, "A distributed OFDMA medium access control for underwater acoustic sensors networks," in *Proceedings of the IEEE International Conference on Communications (ICC '11)*, June 2011.
- [8] C. C. Hsu, K. F. Lai, C. F. Chou, and K. C. J. Lin, "Stmac: spatial-temporal mac scheduling for underwater sensor networks," in *Proceedings of the 28th Conference on Computer Communications (INFOCOM '09)*, pp. 1827–1835, IEEE, Rio de Janeiro, Brazil, April 2009.
- [9] Y.-D. Chen, C.-Y. Lien, S.-W. Chuang, and K.-P. Shih, "DSSS: a TDMA-based MAC protocol with Dynamic Slot Scheduling Strategy for underwater acoustic sensor networks," in *Proceedings of the IEEE OCEANS*, pp. 1–6, Santander, Spain, June 2011.
- [10] K. Kredo II, P. Djukic, and P. Mohapatra, "Stump: exploiting position diversity in the staggered tdma underwater mac protocol," in *Proceedings of the IEEE 28th Conference on Computer Communications (INFOCOM '09)*, pp. 2961–2965, April 2009.

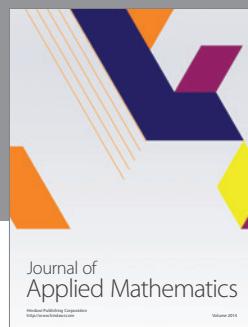
- [11] M. S. Marcal Molins, "Slotted FAMA: a MAC protocol for underwater acoustic networks," in *Proceedings of the IEEE OCEANS 2006*, Boston, Mass, USA, September 2006.
- [12] N. Chirdchoo, W.-S. Soh, and K. C. Chua, "RIPT: a receiver-initiated reservation-based protocol for underwater acoustic networks," *IEEE Journal on Selected Areas in Communications*, vol. 26, no. 9, pp. 1744–1753, 2008.
- [13] Z. Zhou, Z. Peng, P. Xie, J.-H. Cui, and Z. Jiang, "Exploring random access and handshaking techniques in underwater wireless acoustic networks," *Eurasip Journal on Wireless Communications and Networking*, vol. 2013, article 95, 2013.
- [14] Y. W. Chihmin Chao, "A multiple rendezvous multichannel mac protocol for underwater sensor networks," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 43, no. 1, pp. 128–138, 2010.
- [15] H. Ramezani and G. Leus, "DMC-MAC: dynamic multi-channel MAC in underwater acoustic networks," in *Proceedings of the 21st European Signal Processing Conference (EUSIPCO '13)*, pp. 1–5, Marrakech, Morocco, September 2013.
- [16] Z. Zhou, Z. Peng, J.-H. Cui, and Z. Jiang, "Handling triple hidden terminal problems for multichannel MAC in long-delay underwater sensor networks," *IEEE Transactions on Mobile Computing*, vol. 11, no. 1, pp. 139–154, 2010.
- [17] O. Yilmaz and S. Rickard, "Blind separation of speech mixtures via time-frequency masking," *IEEE Transactions on Signal Processing*, vol. 52, no. 7, pp. 1830–1847, 2004.
- [18] T. N. Sainath, B. Kingsbury, and B. Ramabhadran, "Auto-encoder bottleneck features using deep belief networks," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '12)*, pp. 4153–4156, Kyoto, Japan, March 2012.
- [19] X. Zhang, L. Pan, and A. Paun, "On the universality of axon p system," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 11, pp. 2816–2829, 2015.
- [20] X. Zhang, Y. Liu, B. Luo, and L. Pan, "Computational power of tissue P systems for generating control languages," *Information Sciences*, vol. 278, pp. 285–297, 2014.
- [21] X. Zhang, Y. Tian, and Y. Jin, "A knee point-driven evolutionary algorithm for many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 6, pp. 761–776, 2015.
- [22] X. Zhang, Y. Tian, R. Cheng, and Y. Jin, "An efficient approach to nondominated sorting for evolutionary multiobjective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 2, pp. 201–213, 2015.



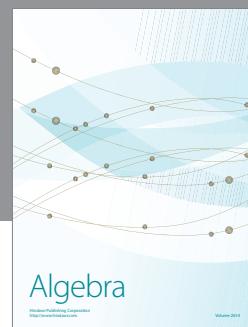
Advances in
Operations Research



Advances in
Decision Sciences



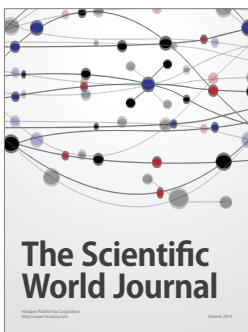
Journal of
Applied Mathematics



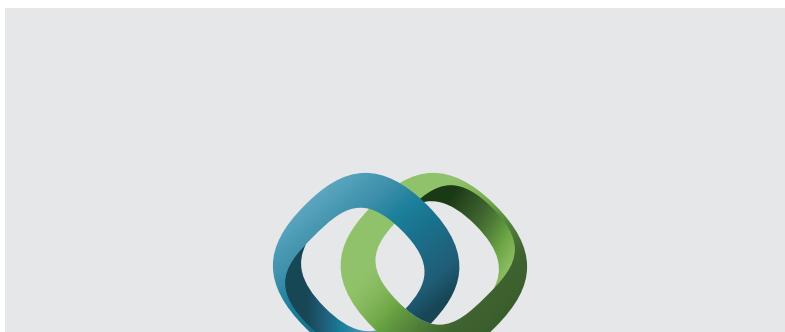
Algebra



Journal of
Probability and Statistics



The Scientific
World Journal

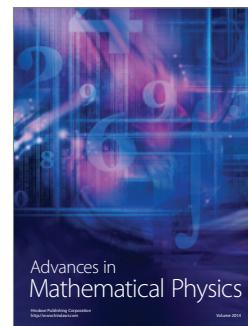


Hindawi

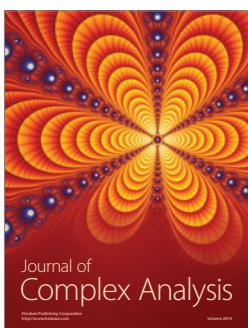
Submit your manuscripts at
<http://www.hindawi.com>



International Journal of
Combinatorics



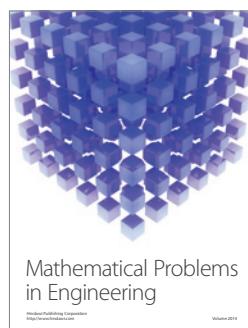
Advances in
Mathematical Physics



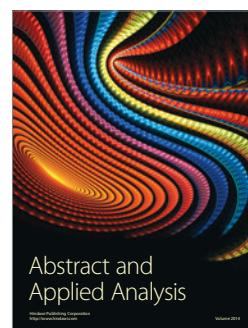
Journal of
Complex Analysis



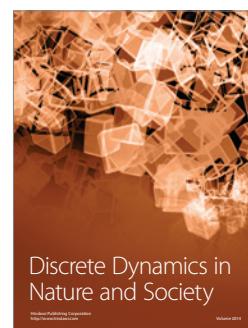
Journal of
Mathematics



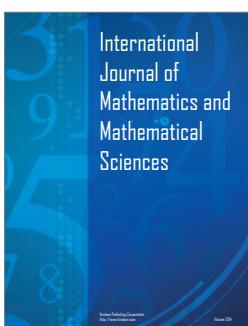
Mathematical Problems
in Engineering



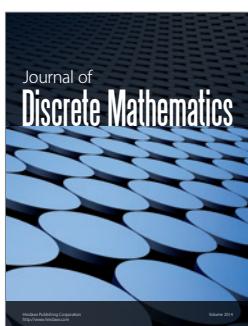
Abstract and
Applied Analysis



Discrete Dynamics in
Nature and Society



International
Journal of
Mathematics and
Mathematical
Sciences



Journal of
Discrete Mathematics



Journal of
Function Spaces



International Journal of
Stochastic Analysis



Journal of
Optimization