IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 29 January 2024, accepted 12 February 2024, date of publication 21 February 2024, date of current version 1 March 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3368857

APPLIED RESEARCH

Trigonometric Coordinate Transformation Blind Equalization Algorithm Based on Bi-Direction Long and Short-Term Memory Neural Networks

NA LIU[1], ZUOXUN WANG[2], AND HAIWEN WEI[3]
[1]College of Engineering, Shandong Xiehe University, Jinan 250000, China
[2]College of Information and Electronic Engineering, Shandong Technology and Business University, Yantai 264000, China
[3]Shandong Meteorological Bureau, Jinan 250000, China

Corresponding author: Haiwen Wei (cjweihaiwen@163.com)

This work was supported in part by the China Meteorological Administration Innovation Development Project under Grant CXFZ2023J008; and in part by the Shandong Province University Youth Entrepreneurship Talent Introduction Program, namely the Ultraviolet Pulsed Bright Light Disinfection Mobile Robot Project of Shandong Xiehe University–Medical Convenient Robot Technology Innovation Team under Project 2021KJ088.

ABSTRACT Aiming at the problem that the feedforward neural network blind equalization algorithm has a slow convergence rate and a large steady-state error when equalizing the high-order non-constant modulus signals, a trigonometric coordinate transformation blind equalization algorithm based on Bi-direction long and short-term memory (BLSTM) neural networks (BLSTM-TCT-CMA) is proposed. First, the BLSTM neural network has a strong processing ability for one-dimensional long-sequence signals, which is suitable for high-order signal sequence information. Secondly, a triangular coordinate transformation method was introduced in the BLSTM neural network loss function to transform the statistical modulus of the non-constant modulus signal into a constant modulus value, which speeds up convergence and further reduces the steady-state error. It was observed through the simulation that compared with the constant modulus blind equalization algorithm (CMA), the square contour blind equalization algorithm based on BP neural networks (SCA-BP-CMA) and the tunable activation functions blind equalization algorithm based on complex BP neural network (TAF-CBP-CMA). When the BLSTM-TCT-CMA equalized the 32QAM signal, the steady-state error was -13dB, and the loss function converged at 800 steps. When the 64QAM signal was equalized, the steady-state error was -10.5dB, and the loss function converged at 1200 steps. It is concluded that both indicators were optimal, and the CT-GRUNN-CMA output signal constellation was the clearest.

INDEX TERMS Blind equalization, long and short-term memory, convergence speed, neural networks, coordinate transformation.

I. INTRODUCTION

In modern wireless communication systems, Inter-Symbol Interference (ISI) caused by environmental noise and channel multipath effect is the main reason for the high bit error rate at the receiving end of communication systems. Blind equalization technology only needs the statistical mode value of the input signal without a training sequence. It adjusts the weight vector of the equalizer in real time through the selected algorithm model, effectively reducing the inter-symbol interference [1], [2].

The blind equalization algorithm based on a feedforward neural network can effectively reduce inter-symbol interference. Still, the network weight vector parameters are few, the input signals are independent of each other at the time before and after, and the feature extraction ability of the input sequence with rich information is less [3], [4]. Yu et al. [5] provided a detailed exposition on the evolution of RNN networks to LSTM and analyzed the gating cell structure in LSTM. Xu et al. [6] employed the SCLSTM model to

The associate editor coordinating the review of this manuscript and approving it for publication was Siddhartha Bhattacharyya.

predict the temporal relationship of RIS-UE channels, achieving more accurate channel state information acquisition. There are input gates, forgetting gates, and output gates in the hidden layer units of the BLSTM neural network, which control the retention and abandonment of input information in real time, and the weight parameters of each gate are controlled by the input at the current time and the output at the previous time, so it has strong adaptive adjustment and fault tolerance ability [7], [8]. The Trigonometric Coordinate Transformation based CMA (TCT-CMA) algorithm transforms the constellation coordinates of high-order nonconstant modulus signals to the origin of coordinates through trigonometric functions, and the statistical modulus values are transformed from nonconstant modulus to constant modulus [9], [10], [11], [12], [13]. Therefore, it will be a meaningful research topic to apply the BLSTM neural network and TCT-CMA to blind equalization technology.

In this paper, based on giving full play to the advantages of BLSTM neural network and TCT-CMA, a bidirectional long and short-term memory neural network triangular coordinate transformation blind equalization algorithm (BLSTM-TCT-CMA, BLSTM-based TCT-CMA) is proposed. The algorithm uses a BLSTM neural network to fit the inverse channel of the nonlinear noise channel [14]. It uses TCT-CMA to reduce the steady-state error further and accelerate the convergence speed of the loss function. The simulation results verify the excellent blind equalization effect of BLSTM-TCT-CMA for high-order QAM signals.

To ensure clarity in symbol usage throughout the paper and enhance readability, the symbols and mathematical notations used in the entire document are defined as follows: a represents the input signal, h denotes the channel impulse response, x represents the channel output, w stands for Gaussian white noise, y represents the equalizer input, l denotes the convolutional weight vector, z represents the equalizer output, $\hat{a}$ signifies the decision output, c represents the tap coefficient, C represents the cell state, s denotes the hidden layer state, i represents the input gate, g stands for the forget gate, o represents the output gate, W denotes the weight, J signifies the loss function, and n, t and k respectively represent the signal sequence, time step, and hidden layer unit.

II. BLIND EQUALIZER ALGORITHM BASED ON FEEDFORWARD NEURAL NETWORK

The structure of the blind equalizer algorithm based on a feedforward neural network is shown in FIGURE 1.

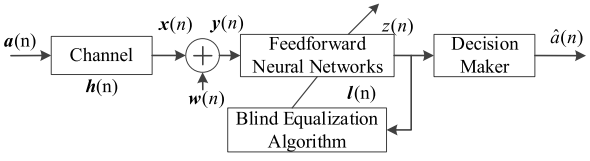

FIGURE 1. Block Diagram of Blind Equalizer Algorithm Based on Feedforward Neural Network.

In FIGURE 1, $\boldsymbol{a}(\boldsymbol{n})$ is the input signal sequence, $\boldsymbol{h}(\boldsymbol{n})$ is the impulse response of the channel, $\boldsymbol{x}(\boldsymbol{n})$ is the output sequence of the channel, $\boldsymbol{w}(\boldsymbol{n})$ is the white Gaussian noise, $\boldsymbol{y}(\boldsymbol{n})$ is the input sequence of the blind equalizer, $\boldsymbol{l}(\boldsymbol{n})$ is the equivalent convolution weight vector of the blind equalizer, and is the recovery output signal sequence. $\hat{a}(\boldsymbol{n})$ is the decision output signal sequence. The output of the equalizer is

$$\boldsymbol{z}(\boldsymbol{n}) = \boldsymbol{l}(\boldsymbol{n}) \otimes [\boldsymbol{a}(\boldsymbol{n}) \otimes \boldsymbol{h}(\boldsymbol{n}) + \boldsymbol{w}(\boldsymbol{n})] \tag{1}$$

In the formula, $\otimes$ represents the convolution operation. Since the equalizer is applied to the inter-symbol I nterference caused by channel distortion, the influence of noise is temporarily ignored here. Then, the output $\boldsymbol{z}(\boldsymbol{n})$ of the blind equalizer can be expressed as

$$z(n) = \boldsymbol{l}(n) \otimes \boldsymbol{h}(n) \otimes \boldsymbol{a}(n) \tag{2}$$

Let $\boldsymbol{c}(n)$ be the combined tap factor of the entire blind equalizer, then it can be written as

$$\boldsymbol{c}(n) = \boldsymbol{l}(n) \otimes \boldsymbol{h}(n) \tag{3}$$

The blind equalizer achieves optimal performance when the central tap scalar modulus is one, and all other elements are zero, i.e.,

$$c = \left[0, \ldots, 0, e^{j\emptyset}, 0, \ldots, 0\right]^T \tag{4}$$

In the above formula, $\emptyset$ is a constant phase shift, and can be removed using the discriminator.

III. BLSTM NEURAL NETWORK TRIANGULAR COORDINATE TRANSFORMATION BLIND EQUALIZATION ALGORITHM

A. BLSTM NEURAL NETWORK BLIND EQUALIZATION MODEL

The number of neurons in the input layer of the BLSTM neural network is the tap length of the transverse filter, and the weight vector in the network is adaptively adjusted through the loss function so that the network output sequence is gradually close to the original transmission sequence [15], [16], [17], [18]. BLSTM neural network hidden layer unit structure, as shown in FIGURE 2.

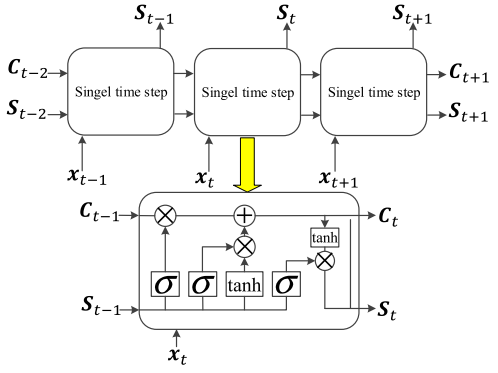

FIGURE 2. Bi-direction LSTM neural network hidden layer unit structure diagram.

In FIGURE 2, $\boldsymbol{C}_t$ represents the cell state at the time t, $\boldsymbol{x}_t$ represents the hidden layer input at the time t, $\boldsymbol{s}_t$ represents the hidden layer output at the time t, and σ represents the sigmoid function. Multiple isomorphic units constitute the hidden layer of the BLSTM neural network. The information flow transmission formula of the K-th unit at time t in the hidden layer of the BLSTM neural network is.

$$g_t(k) = \sigma\left[\boldsymbol{W}_{gs}(k) \cdot \boldsymbol{s}_{t-1}(k) + \boldsymbol{W}_{gx}(k) \cdot \boldsymbol{x}_t(k)\right] \quad (5)$$

$$i_t(k) = \sigma\left[\boldsymbol{W}_{is}(k) \cdot \boldsymbol{s}_{t-1}(k) + \boldsymbol{W}_{ix}(k) \cdot \boldsymbol{x}_t(k)\right] \quad (6)$$

$$\hat{\boldsymbol{C}}_t(k) = \tanh\left[\boldsymbol{W}_{cx}(k) \cdot \boldsymbol{s}_{t-1}(k) + \boldsymbol{W}_{cx}(k) \cdot \boldsymbol{x}_t(k)\right] \quad (7)$$

$$o_t(k) = \sigma\left[\boldsymbol{W}_{os}(k) \cdot \boldsymbol{s}_{t-1}(k) + \boldsymbol{W}_{ox}(k) \cdot \boldsymbol{x}_t(k)\right] \quad (8)$$

$$\boldsymbol{C}_t(k) = g_t(k) \cdot \boldsymbol{C}_{t-1}(k) + i_t(k) \cdot \hat{C}_t(k) \quad (9)$$

$$\boldsymbol{s}_t(k) = o_t(k) \cdot \tanh\left[\boldsymbol{C}_t(k)\right] \quad (10)$$

In the formula, $\boldsymbol{g}$ is the forgetting gate value, $\boldsymbol{i}$ is the input gate value, $\boldsymbol{o}$ is the output gate value, $\boldsymbol{W}$ is the connection weight vector, and $\hat{C}$ is the intermediate state. The Sigmoid function controls the three gate values between 0 and 1. When the gate value is close to 1, the cell state and the intermediate state at the previous time are fully preserved. Otherwise, they are discarded. The BLSTM neural network is unfolded along the time direction, as shown in FIGURE 3. The forward layer and the back layer are connected to the same output layer, which can process the bidirectional information of the signal sequence at the same time.

$$\vec{s}_t(k) = \boldsymbol{W}_x(k)\boldsymbol{x}_t(k) + \boldsymbol{W}_{\vec{s}}(k)\vec{s}_{t-1}(k) \quad (11)$$

$$\overleftarrow{s}_t(k) = \boldsymbol{W}_x(k)\boldsymbol{x}_t(k) + \boldsymbol{W}_{\overleftarrow{s}}(k)\overleftarrow{s}_{t-1}(k) \quad (12)$$

$$\boldsymbol{y}_t(k) = \boldsymbol{W}_{\vec{s}y}(k)\vec{s}_t(k) + \boldsymbol{W}_{\overleftarrow{s}y}(k)\overleftarrow{s}_t(k) \quad (13)$$

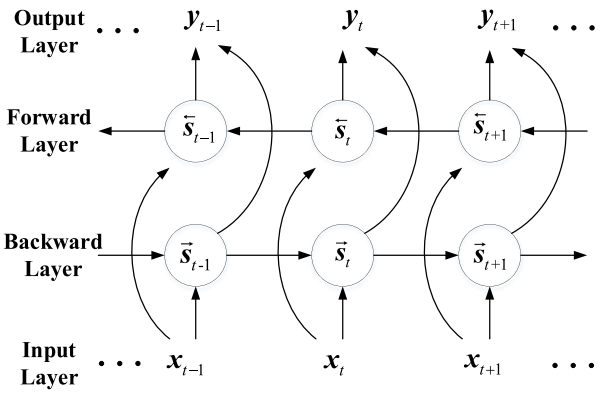

FIGURE 3. Bi-direction LSTM neural network structure.

In the formula, $\vec{s}_t(k)$ is the output at the time t of the KTH unit in the backward layer, $\overleftarrow{s}_t(k)$ is the output at the time t of the K-th unit in the forward layer, $\boldsymbol{y}_t(k)$ is the value at the time t of the KTH unit in the output layer, and $\boldsymbol{W}$ is the connection weight vector. Since the magnitude of the output signal modulus of the blind equalizer is determined by different modulation types, the BLSTM neural network output layer transfer function is defined as

$$f(x) = x + \beta\frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (14)$$

In the formula, the value of the transfer constant β is determined by the amplitude of the modulation signal. The output of the network at time t is

$$out(t) = f\left[\boldsymbol{W}_{out}(t) \cdot \boldsymbol{y}_t\right] \quad (15)$$

In the formula, $\boldsymbol{W}_{out}(t)$ is the output layer weight vector at time t.

B. CONSTANT MODULUS BLIND EQUALIZATION ALGORITHM BASED ON TRIANGULAR COORDINATE TRANSFORMATION

The tap coefficients of the Constant modulus algorithm (CMA) are updated continuously by iterative calculation of the loss function so that the modulus value of the output signal of the blind equalizer gradually approaches the statistical modulus value of the original signal. When the input signal is a constant mode signal, the blind equalizer can achieve the zero-forcing condition under ideal conditions. However, when the input signal is a non-constant modulus signal, CMA cannot achieve ideal equalization because the statistical modulus value does not match the input constellation diagram [19], [20]. This problem is solved effectively by modifying the loss function to transform the non-constant modulus constellation position to the constant modulus constellation position.

The coordinate transformation algorithm varies for different modulation signal types. The coordinate transformation constant modulus blind equalization algorithm is restricted to 16QAM signals. Although the enhanced coordinate transformation constant modulus blind equalization algorithm can be applied to MQAM and MPAM signals, the loss function term needs to be adjusted as the signal order increases. The expression of the loss function becomes more complex with higher order, making it less universally applicable and requiring more computational resources [17], [21]. In this paper, a constant modulus blind equalization algorithm based on triangular coordinate transformation is proposed for MQAM and MPAM signals. When the input is MPAM real signal, a new loss function is defined as

$$J(k) = \frac{E\left\{\sin^2\left[y(k)\cdot\pi\right] - R^2\right\}^2}{\pi} \quad (16)$$

$$R^2 = \frac{E\left\{\sin^4\left[a(k)\cdot\pi\right]\right\}}{E\left\{\sin^2\left[a(k)\cdot\pi\right]\right\}} \quad (17)$$

In the formula, $y(k)$ is the equalizer output and $a(k)$ is the original transmitted signal. Since $\sin\left[a(k)\cdot\pi\right] \equiv 0$, R^2 is meaningless, the constellation positions are equalized to the origin of coordinates, and the modulus value corresponding to the transmitted signal is 0. When the input is a MQAM complex signal, taking the 16QAM signal as an example, the constellation distribution after coordinate transformation is shown in FIGURE 4.

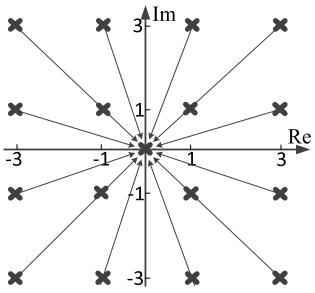

FIGURE 4. Constellation after coordinate transformation.

According to Equation (16), the complex signal loss function is defined as

$$J(k) = \frac{E\left\{\sin^2\left[y_r(k)\cdot\pi\right] - R_r^2\right\}^2}{\pi} + \frac{E\left\{\sin^2\left[y_i(k)\cdot\pi\right] - R_i^2\right\}^2}{\pi} \tag{18}$$

The simplified modulus value and the order are obtained.

$$J(k) = \frac{E\left\{\sin^2\left[y_r(k)\cdot\pi\right]\right\}}{\pi} + \frac{E\left\{\sin^2\left[y_i(k)\cdot\pi\right]\right\}}{\pi} \tag{19}$$

The tap coefficient update formula is

$$\boldsymbol{f}(k+1) = \boldsymbol{f}(k) + \mu \boldsymbol{x}_k^* \left\{\sin\left[y_r(k)\cdot 2\pi\right] + j\sin\left[y_i(k)\cdot 2\pi\right]\right\} \tag{20}$$

In the formula, μ is the length of the iteration step, $\boldsymbol{x}_k^*$ is the input sequence after taking the complex conjugate.

C. BLSTM NEURAL NETWORK TRIANGULAR COORDINATE TRANSFORMATION BLIND EQUALIZATION ALGORITHM

Aiming at the problem that the statistical modulus value of the non-constant modulus signal does not conform to the constellation of the input signal, and further reduce the residual error after the convergence of the loss function, this paper proposes the BLSTM-TCT-CMA. The weight vector of the BLSTM neural network is updated in real-time by the iterative error obtained from the triangular coordinate transformation loss function. Since the forward layer is like the back layer, only the back layer is discussed in the hidden layer, and the update formula of the network output weight vector is obtained by Equation (19) and Equation (20):

$$\frac{\partial J(k)}{\partial \boldsymbol{W}_{out}(t)} = \left\{\sin\left[y_r(k)\cdot 2\pi\right] + j\cdot\sin\left[y_i(k)\cdot 2\pi\right]\right\} \cdot f'\left[\boldsymbol{W}_{out}(t)\cdot\boldsymbol{y}_t\right]\cdot\boldsymbol{y}_t^* \tag{21}$$

$$\boldsymbol{W}_{out}(t+1) = \boldsymbol{W}_{out}(t) + \mu\frac{\partial J(k)}{\partial \boldsymbol{W}_{out}(t)} \tag{22}$$

The updated formula of the output weight vector of the backpropagation layer is

$$\frac{\partial J(k)}{\partial \boldsymbol{W}_{\vec{s}y}(k)} = \left\{\sin\left[y_r(k)\cdot 2\pi\right] + j\cdot\sin\left[y_i(k)\cdot 2\pi\right]\right\} \cdot f'\left[\boldsymbol{W}_{out}(t)\cdot\boldsymbol{y}_t\right]\cdot\boldsymbol{y}_t\cdot\vec{s}_t^*(k) \tag{23}$$

$$\boldsymbol{W}_{\vec{s}y}(k+1) = \boldsymbol{W}_{\vec{s}y}(k) + \mu\frac{\partial J(k)}{\partial \boldsymbol{W}_{\vec{s}y}(k)} \tag{24}$$

The gradient of the output layer values is

$$\frac{\partial J(k)}{\partial \boldsymbol{y}_t(k)} = \left\{\sin\left[y_r(k)\cdot 2\pi\right] + j\cdot\sin\left[y_i(k)\cdot 2\pi\right]\right\} \cdot f'\left[\boldsymbol{W}_{out}(t)\cdot\boldsymbol{y}_t\right]\cdot\boldsymbol{W}_{out}^*(t) \tag{25}$$

The gradient between the hidden layer value and the cell state value is

$$\frac{\partial J(k)}{\partial \vec{s}_t(k)} = \frac{\partial J(k)}{\partial \boldsymbol{y}_t(k)}\cdot\frac{\partial \boldsymbol{y}_t(k)}{\partial \vec{s}_t(k)} = \frac{\partial J(k)}{\partial \boldsymbol{y}_t(k)}\cdot\boldsymbol{W}_{\vec{s}y}(k) \tag{26}$$

$$\frac{\partial J(k)}{\partial \boldsymbol{C}_t(k)} = \frac{\partial J(k)}{\partial \vec{s}_t(k)}\cdot o_t(k)\cdot\tanh'\left(\boldsymbol{C}_t(k)\right) \tag{27}$$

The updated formula of the input weight vector of the backward layer is

$$\boldsymbol{W}_x(k+1) = \boldsymbol{W}_x(k) + \mu\cdot\frac{\partial J(k)}{\partial \vec{s}_t(k)}\cdot\boldsymbol{x}_t(k) \tag{28}$$

$$\boldsymbol{W}_{\vec{s}}(k+1) = \boldsymbol{W}_{\vec{s}}(k) + \mu\cdot\frac{\partial J(k)}{\partial \vec{s}_t(k)}\cdot\vec{s}_{t-1}(k) \tag{29}$$

The updated formula of the input gate weight vector in the backward layer unit is as follows

$$\begin{aligned}&\boldsymbol{W}_{ix}(k+1)\\ &= \boldsymbol{W}_{ix}(k) + \mu\cdot\frac{\partial J(k)}{\partial \boldsymbol{C}_t(k)}\cdot\hat{C}_t(k)\\ &\quad\cdot\sigma'\left[\boldsymbol{W}_{is}(k)\cdot\vec{s}_{t-1}(k) + \boldsymbol{W}_{ix}(k)\cdot\boldsymbol{x}_t(k)\right]\cdot\boldsymbol{x}_t^*(k)\end{aligned} \tag{30}$$

$$\begin{aligned}&\boldsymbol{W}_{is}(k+1)\\ &= \boldsymbol{W}_{is}(k) + \mu\cdot\frac{\partial J(k)}{\partial \boldsymbol{C}_t(k)}\cdot\hat{C}_t(k)\\ &\quad\cdot\sigma'\left[\boldsymbol{W}_{is}(k)\cdot\vec{s}_{t-1}(k) + \boldsymbol{W}_{ix}(k)\cdot\boldsymbol{x}_t(k)\right]\cdot\vec{s}_{t-1}^*(k)\end{aligned} \tag{31}$$

The update formula of the forgetting gate weight vector is given by

$$\begin{aligned}&\boldsymbol{W}_{gx}(k+1)\\ &= \boldsymbol{W}_{gx}(k) + \mu\cdot\frac{\partial J(k)}{\partial \boldsymbol{C}_t(k)}\cdot\boldsymbol{C}_{t-1}(k)\\ &\quad\cdot\sigma'\left[\boldsymbol{W}_{gs}(k)\cdot\vec{s}_{t-1}(k) + \boldsymbol{W}_{gx}(k)\cdot\boldsymbol{x}_t(k)\right]\cdot\boldsymbol{x}_t^*(k)\end{aligned} \tag{32}$$

$$\begin{aligned}&\boldsymbol{W}_{gs}(k+1)\\ &= \boldsymbol{W}_{gs}(k) + \mu\cdot\frac{\partial J(k)}{\partial \boldsymbol{C}_t(k)}\cdot\boldsymbol{C}_{t-1}(k)\\ &\quad\cdot\sigma'\left[\boldsymbol{W}_{gs}(k)\cdot\vec{s}_{t-1}(k) + \boldsymbol{W}_{gx}(k)\cdot\boldsymbol{x}_t(k)\right]\cdot\vec{s}_{t-1}^*(k)\end{aligned} \tag{33}$$

The updated formula of the output gate weight vector is given by

$$\begin{aligned}&\boldsymbol{W}_{ox}(k+1)\\ &= \boldsymbol{W}_{ox}(k) + \mu\cdot\frac{\partial J(k)}{\partial \vec{s}_t(k)}\cdot\tanh\left[\boldsymbol{C}_t(k)\right]\\ &\quad\cdot\sigma'\left[\boldsymbol{W}_{os}(k)\cdot\vec{s}_{t-1}(k) + \boldsymbol{W}_{ox}(k)\cdot\boldsymbol{x}_t(k)\right]\cdot\boldsymbol{x}_t^*(k)\end{aligned} \tag{34}$$

$$\boldsymbol{W}_{os}(k+1) = \boldsymbol{W}_{os}(k) + \mu \cdot \frac{\partial J(k)}{\partial \vec{s}_t(k)} \cdot \tanh\left[\boldsymbol{C}_t(k)\right] \cdot \sigma'\left[\boldsymbol{W}_{os}(k) \cdot \vec{s}_{t-1}(k) + \boldsymbol{W}_{ox}(k) \cdot \boldsymbol{x}_t(k)\right] \cdot \vec{s}^*_{t-1}(k) \tag{35}$$

The updated formula of the intermediate state weight vector is given by

$$\boldsymbol{W}_{cx}(k+1) = \boldsymbol{W}_{cx}(k) + \mu \cdot \frac{\partial J(k)}{\partial \boldsymbol{C}_t(k)} \cdot i_t(k) \cdot \tanh'\left[\boldsymbol{W}_{cs}(k) \cdot \vec{s}_{t-1}(k) + \boldsymbol{W}_{cx}(k) \cdot \boldsymbol{x}_t(k)\right] \cdot \boldsymbol{x}^*_t(k) \tag{36}$$

$$\boldsymbol{W}_{cs}(k+1) = \boldsymbol{W}_{cs}(k) + \mu \cdot \frac{\partial J(k)}{\partial \boldsymbol{C}_t(k)} \cdot i_t(k) \cdot \tanh'\left[\boldsymbol{W}_{cs}(k) \cdot \vec{s}_{t-1}(k) + \boldsymbol{W}_{cx}(k) \cdot \boldsymbol{x}_t(k)\right] \cdot \vec{s}^*_{t-1}(k) \tag{37}$$

IV. SIMULATION RESULTS

To verify the effectiveness of BLSTM-TCT-CMA, CMA, SCA-BP-CMA [14] and TAF-CBP-CMA [15] are compared with BLSTM-TCT-CMA. The transmitted signal adopts 32QAM signal and 64QAM signal, each transmitting 30000 characters; the channel is a typical wireless mobile communication channel h = [0.005, 0.009, −0.024, 0.854, −0.218, 0.049, −0.016], the signal to noise ratio of Gaussian white noise is 15db. The number of input units in the neural network is equal to the number of taps in the equalizer, and then perform computer MATLAB simulation.

A. MODEL PARAMETER OPTIMIZATION

Grid search [22] refers to projecting N parameters that need to be optimized into an N-dimensional space and dividing it into a grid based on the range and step size of each parameter, then traversing all the intersection points in the grid. When the parameter step size is small enough, grid search is more likely to find the global optimal solution. In the case of discrete values for the number of hidden units and propagation constants of BLSTM-TCT-CMA, grid search is used to determine these values. The multi-class hyperparameters for CMA, SCA-BP-CMA, and TAF-CBP-CMA algorithms are set according to reference literature. The sensitivity test of hyperparameters for 32QAM and 64QAM experiments of BLSTM-TCT-CMA is shown in FIGURE 5 and 6.

FIGURES 5 and 6 respectively demonstrate the process of grid search for selecting the number of hidden units and propagation constants for BLSTM-TCT-CMA. Only the results of the optimal parameter values within a range of 2 to 3 steps are shown in FIGURE 5 and 6. The numbers in the grid represent the final Mean Square Error (MSE) after convergence of the loss function, measured in dB. The experimental results in FIGURES 5 and 6 indicate that for 32QAM signals, the optimal number of hidden units and propagation constant for BLSTM-TCT-CMA are 25 and 8, respectively. For 64QAM signals, the optimal number of hidden units and propagation constant are 29 and 12, respectively.

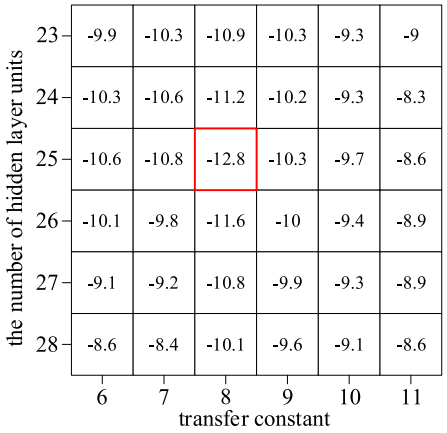

FIGURE 5. Model Parameter Optimization Experiment (32QAM).

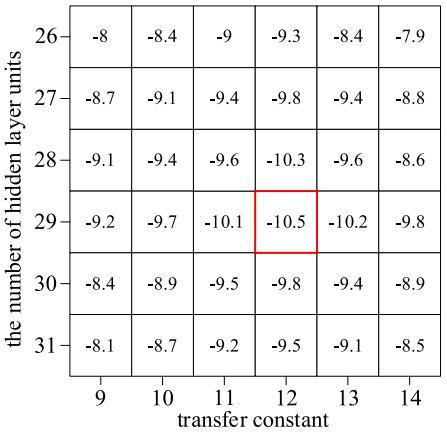

FIGURE 6. Model Parameter Optimization Experiment (64QAM).

B. 32QAM SIGNALS

When the transmitted signal is a 32QAM modulated signal, the number of taps of the CMA blind equalizer is 15, and the iteration step is 0.000006. The network structure of SCA-BP-CMA is (15, 10, 1), and the learning rate of the neural network is $\mu_{SCA-BP} = 0.0002$. The network structure of TAF-CBP-CMA is (15, 7, 1), and the learning rate of the neural network is $\mu_{TAF-CBP} = 0.0005$. The network structure of BLSTM-TCT-CMA is (15,25,25,1), the transfer constant is $\beta = 8$, and the learning rate of the neural network is $\mu_{BLSTM-TCT} = 0.00015$. Monte Carlo simulation experiments were performed 500 times, and the results are shown in FIGURES 7 and 8.

FIGURES 7and 8 compare CMA, SCA-BP-CMA, TAF-CBP-CMA, and BLSTM-TCT-CMA, respectively. In terms of convergence speed, BLSTM-TCT-CMA is about 1200 steps faster than TAF-CBP-CMA, about 2700 steps

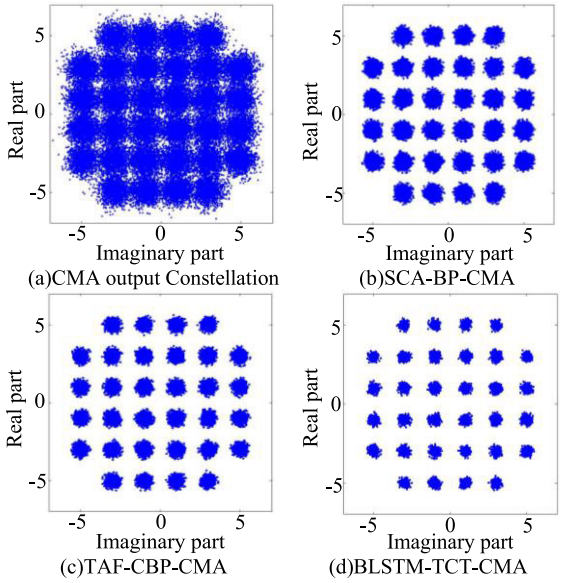

FIGURE 7. 32QAM Simulation experiment output constellation.

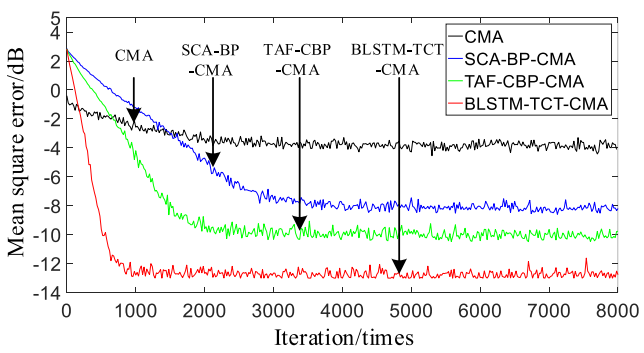

FIGURE 8. 32QAM Simulation experiments mean square error curve.

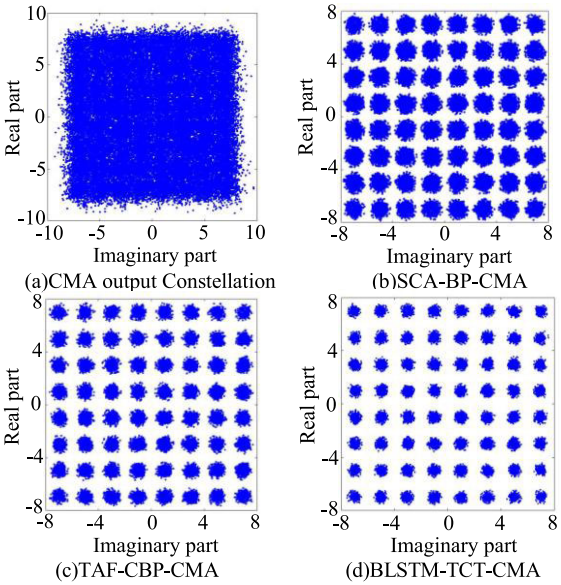

FIGURE 9. 64QAM Simulation experiment output constellation.

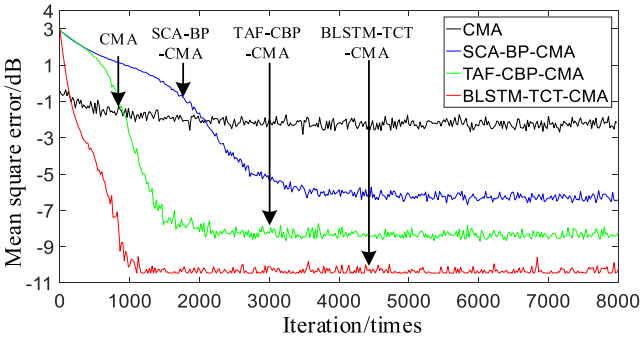

FIGURE 10. 64QAM Simulation experiments mean square error curve.

faster than SCA-BP-CMA, and about 1200 steps faster than CMA. In terms of steady-state error, BLSTM-TCT-CMA reduces about 3.5dB compared with TAF-CBP-CMA, about 5dB compared with SCA-BP-CMA, and about 11dB compared with CMA. The output constellation map of BLSTM-TCT-CMA is more focused and clearer than that of TAF-CBP-CMA, SCA-BP-CMA, and CMA.

C. 64QAM SIGNALS

When the transmitted signal is a 64QAM modulated signal, the number of taps of the CMA blind equalizer is 17, and the iteration step is 0.0000008. The network structure of SCA-BP- CMA is (17, 13, 1), and the learning rate of the neural network is $\mu_{SCA-BP} = 0.0005$. The network structure of TAF-CBP-CMA is (17, 9, 1), and the learning rate of the neural network is $\mu_{TAF-CBP} = 0.0065$. The network structure of BLSTM-TCT-CMA is (17,29,29,1), the transfer constant is $\beta = 12$, and the learning rate of the neural network is $\mu_{BLSTM-TCT} = 0.00015$. Monte Carlo simulation experiments were performed 500 times, and the results are shown in FIGURES 9 and 10.

FIGURES 9 and 10 compare CMA, SCA-BP-CMA, TAF-CBP-CMA, and BLSTM-TCT-CMA, respectively. In terms of convergence speed, BLSTM-TCT-CMA is about 1000 steps faster than TAF-CBP-CMA, about 2500 steps faster than SCA-BP-CMA, and about 1000 steps faster than CMA. In terms of steady-state error, BLSTM-TCT-CMA reduces about 2dB compared with TAF-CBP-CMA, about 4dB compared with SCA-BP-CMA, and about 8dB compared with CMA. The output constellation map of BLSTM-TCT-CMA is more focused and clearer than that of TAF-CBP-CMA, SCA-BP-CMA, and CMA.

V. CONCLUSION

This paper combines the BLSTM neural network with TCT for application in the field of nonlinear channel blind equalization. By leveraging the excellent sequence processing and feature recognition capabilities of the BLSTM neural network, as well as the adaptability of TCT to non-constant modulus signals, it fits the inverse process of nonlinear noisy

channels and achieves blind equalization of communication channels. Simulation experiments show that BLSTM-TCT-CMA improves the slow convergence speed and large steady-state error of traditional blind equalization algorithms such as CMA. The main conclusions of this study are as follows:

(1) The TCT-CMA algorithm replaces the conventional modulus cost function with a novel trigonometric coordinate transformation cost function, which utilizes the periodicity of trigonometric functions. It transforms all higher-order non-constant modulus MQAM signals to the origin of the coordinate system, expanding the algorithm's applicability and improving its nulling capability, while reducing the computational complexity.

(2) The BLSTM-CMA algorithm utilizes the advantage of the bidirectional hidden layers in the BLSTM neural network to optimize the network's processing capability for one-dimensional long sequence signals. It fully utilizes the BLSTM's excellent fast global search ability and channel equalization ability. By processing the signal sequence bidirectionally, it enhances the algorithm's robustness and learning capability.

(3) The BLSTM-TCT-CMA algorithm combines the advantages of TCT-CMA and BLSTM-CMA. It utilizes the real-time adjustment of the cost error by TCT-CMA to the network weight vector of BLSTM, thereby achieving both the computational convergence capability and the high-order transformation adaptability of the algorithm.

Blind equalization technology is an important field in modern digital communication. Inspired by the fact that artificial neural networks can fit any nonlinear process, this paper uses BLSTM as the blind equalizer of the channel. It also proposes TCT to transform constellation points at different modulus values to the same modulus value, allowing statistical modulus to summarize all signal features and achieve theoretical nulling equalization. The introduction of BLSTM-TCT-CMA overcomes the dependence of equalization technology on carrier synchronization. It achieves channel equalization through the inherent features of the received signal, increasing the available bandwidth and unleashing the system's potential. In special communication environments or objects such as seismic wave detection, radar communication systems, and underwater wireless communication systems, where the transmission of training sequences would occupy the effective channel of the signal, the system cannot emit training sequences, thus affecting communication quality. BLSTM-TCT-CMA can fit the inverse channel of the communication environment without relying on training sequences, realizing wireless mobile communication in complex environments. However, BLSTM-TCT-CMA also has certain limitations. The use of multi-gate structures to process high-order signals enhances computational power, but it requires strong computational support. Further research is needed on how to improve the calculation method of the gates to reduce computational complexity. In the future, introducing the Transformer network [23] into the equalizer can be attempted, utilizing its excellent feature recognition and temporal information processing capabilities to further improve the algorithm's fitting ability and computational speed.

REFERENCES

[1] Z. Lu, H. Chen, F. Chen, J. Nie, and G. Ou, "Blind adaptive channel mismatch equalisation method for GNSS antenna arrays," *IET Radar, Sonar Navigat.*, vol. 12, no. 4, pp. 383–389, Apr. 2018.

[2] Y. Guo, *Adaptive Blind Equalization Technology*. Hefei, China: Hefei University of Technology Press, 2007.

[3] E. Giacoumidis, A. Matin, J. Wei, N. J. Doran, L. P. Barry, and X. Wang, "Blind nonlinearity equalization by machine-learning-based clustering for single- and multichannel coherent optical OFDM," *J. Lightw. Technol.*, vol. 36, no. 3, pp. 721–727, Feb. 1, 2018.

[4] K. Cho, B. Van, C. Gulcehre, D. Bahdanau, F. Bougares, and H. Schwenk, "Learning phrase representations using RNN encoder–decoder for statistical machine translation," *Comput. Sci.*, vol. 28, no. 2, pp. 368–378, 2014.

[5] Y. Yu, X. Si, C. Hu, and J. Zhang, "A review of recurrent neural networks: LSTM cells and network architectures," *Neural Comput.*, vol. 31, no. 7, pp. 1235–1270, Jul. 2019.

[6] W. Xu, J. An, Y. Xu, C. Huang, L. Gan, and C. Yuen, "Time-varying channel prediction for RIS-assisted MU-MISO networks via deep learning," 2021, *arXiv:2111.04971*.

[7] X. Wang, J. Wu, and C. Liu, "Fault time series prediction based on LSTM recurrent neural network," *J. Beijing Univ. Aeronaut. Astronaut.*, vol. 44, no. 4, pp. 772–784, 2018.

[8] T. Ergen and S. S. Kozat, "Online training of LSTM networks in distributed systems for variable length data sequences," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 10, pp. 5159–5165, Oct. 2018.

[9] Y. Zheng, F. Li, and L. Zhang, "A human posture detection method based on long short term memory networks," *Comput. Appl.*, vol. 38, no. 6, pp. 1568–1574, 2018.

[10] Y. Guo and H. Bai, "Modeling and simulation of coordinate transformation multimode blind equalization algorithm based on Simulink," *Sci. Technol. Eng.*, vol. 14, no. 19, pp. 80–86, 2014.

[11] Y. Han, B. Li, and Y. Guo, "Research on blind equalizer based on wavelet transform at different positions," *Comput. Appl. Res.*, vol. 29, no. 1, pp. 282–285, 2012.

[12] W. Rao, Y. Guo, and S. Wang, "A new constant modulus blind equalization algorithm based on coordinate transformation," *J. Electron. Sci.*, vol. 29, no. 1, pp. 282–285, 2012.

[13] W. Rao, H. Xu, Y. Guo, and J. Liu, "Joint blind equalization and detection of QAM and PAM transmissions," in *Proc. Int. Conf. Microw. Millim. Wave Technol., Nov.*, May 2010, pp. 1248–1251.

[14] J. Zhao, "Application of the square contour algorithm in blind equalizers based on complex neural networks," in *Proc. Int. Conf. Consum. Electron., Commun. Netw., Nov.*, Nov. 2013, pp. 166–169.

[15] J. Zhao and Z. Gao, "Research on the blind equalization technology based on the complex BP neural network with tunable activation functions," in *Proc. IEEE 2nd Adv. Inf. Technol., Electron. Autom. Control Conf. (IAEAC)*, Nov. 2017, pp. 813–817.

[16] L. Yang, Q. Han, J. Du, L. Cheng, Q. Li, and A. Zhao, "Online blind equalization for QAM signals based on prediction principle via complex echo state network," *IEEE Commun. Lett.*, vol. 24, no. 6, pp. 1338–1341, Jun. 2020.

[17] Y. Zhang, S. Tian, H. Pan, and Y. Zhou, "Adaptive blind equalization of fast time-varying channel with frequency estimation in impulsive noise environment," *IET Commun.*, vol. 15, no. 11, pp. 1507–1517, Jul. 2021.

[18] M. N. Parvin, Y. Sugiura, and T. Shimamura, "Blind equalization based on normalized error in wireless sensor networks," *J. Signal Process.*, vol. 23, no. 5, pp. 215–225, 2019.

[19] J. Li, D.-Z. Feng, and W. X. Zheng, "A robust decision directed algorithm for blind equalization under α-stable noise," *IEEE Trans. Signal Process.*, vol. 69, pp. 4949–4960, 2021.

[20] S. Abrar, A. Zerguine, and K. Abed-Meraim, "Adaptive blind equalization in impulsive noise," in *Proc. 54th Asilomar Conf. Signals, Syst., Comput.*, Nov. 2020, pp. 1415–1419.

[21] N. Takeyama and J. Mitsugi, "Fast blind equalizer exploiting statistical features of symbol constellations," *J. Inf. Process.*, vol. 30, pp. 909–916, Jan. 2022.
[22] W. Xu, J. An, Y. Xu, C. Huang, L. Gan, and C. Yuen, "Time-varying channel prediction for RIS-assisted MU-MISO networks via deep learning," *IEEE Trans. Cogn. Commun. Netw.*, vol. 8, no. 4, pp. 1802–1815, Dec. 2022.
[23] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 5996–6008.

NA LIU received the B.S. and M.S. degrees in electrical engineering from Shandong University of Science and Technology, Zibo, China, in 2016 and 2019, respectively. She has published more than ten journals in the related design and art fields. Her research interests include intelligent power supply and distribution networks, big data mining, and the artificial intelligence applications.

ZUOXUN WANG received the B.E. degree in motor and control from Shandong University of Technology, in 1997, the M.E. degree in computer application technology from Northeast Electric Power University, in 2005, and the Ph.D. degree in control theory and control engineering from the Institute of Automation, Chinese Academy of Sciences, in 2015. He is currently a Professor with the School of Information and Electronic Engineering, Shandong Technology and Business University. He is the author/coauthor of over 40 journal articles and conference papers. His research interests include intelligent robot, control theory, and control engineering.

HAIWEN WEI received the B.S. and M.S. degrees in electronic information engineering from Nanjing University of Information Science and Technology, Nanjing, China, in 2016 and 2019, respectively. He is currently engaged in deep learning, computer vision, meteorological big data application, and other related fields. He has published more than ten journals in the related design and art fields.