Design of Digital FIR Filter Based on Long Short-Term Memory Neural Network

Fangxin Zhu, Xing Yang, Jiansheng Yang, Jiyao Yang, Mei Zhang and Shengjian Liu

***Abstract*—This paper proposes a long short-term memory (LSTM) neural network method for designing digital finite impulse response (FIR) low-pass, high-pass, band-pass, and band-stop filters. We first establish a mathematical model of the digital FIR filter to derive its practical amplitude response. Then, the amplitude error between the expected and practical amplitude responses is calculated. The objective of designing a digital FIR filter is to achieve the amplitude error minimization. Therefore, we propose an LSTM neural network to minimize the amplitude error, thereby obtaining the coefficients for the digital FIR filter. Several design examples and comparisons with back propagation neural network (BPNN), traditional neural network (TNN), and the rectangular window method illustrate the performance of our proposed LSTM neural network design. These design examples and comparisons highlight the improvements in the passband peak-to-peak deviation, the stopband attenuation and mean square error (MSE). However, these improvements result in higher computational complexity. Consequently, a tradeoff exists among the passband peak-to-peak deviation, the stopband attenuation, MSE, and computational complexity.**

***Index Terms*—Optimal design, Digital FIR filter, Amplitude error, Long Short-Term Memory, Neural network**

I. Introduction

THE development of modern electronic systems is closely related to advancements in digital signal processing (DSP) technology [1-4]. Digital finite impulse response (FIR) filters play a crucial role in digital signal processing (DSP) systems. They are extensively used for signal filtering [5], channel equalization [6], and data interpolation or extraction [7]

Digital FIR filters are typically categorized into four types: Type I, Type II, Type III, and Type IV [8-9]. Among these types, only the Type I FIR filter can be used to design low-pass, high-pass, band-pass, and band-stop filters. Therefore, the optimal design of Type I digital FIR filter has attracted the attention of many scholars.

The window function is widely used for designing Type I digital FIR filters due to its simplicity and low computational complexity. However, the window function method has worse effectiveness [10-13]. Currently, Type I digital FIR filter can be designed by minimizing the amplitude error between its expected and practical amplitude responses. For example, the least squares (LS) method can be used for minimizing this amplitude error [14]. Additionally, the non-iterative weighted least squares (WLS) method can also be employed for minimizing the amplitude error between the expected and practical amplitude responses [15]. All of these methods can be used to design Type I digital FIR filter and demonstrate superior effectiveness compared to the window function.

Machine learning is a popular research topic and one of its primary objectives is to minimize the amplitude error between the expected and practical parameters in the specific application [16-18]. For example, Y. Chen proposes a nonlinear voltage prediction method based on long short-term memory (LSTM) neural network to minimize the amplitude error between the expected and practical nonlinear voltages [19]. A. Pranolo et al. propose using an LSTM neural network to minimize the amplitude error between the expected and practical rainfall values for more accurate forecasts [20]. X. Qing et al. propose a solar prediction scheme based on LSTM neural network. This scheme minimizes the amplitude error between the expected and practical hourly day-ahead solar irradiance values [21]. These papers demonstrate that the LSTM neural network performs more effectively than the back propagation neural network (BPNN). Although these papers do not specifically address the use of the LSTM neural network for designing digital FIR filter, they provide valuable insights into designing digital FIR filter.

At present, only a limited number of papers have proposed to design digital FIR filter using machine learning methods. Among these papers, a traditional neural network (TNN) is proposed to design digital FIR filter [22]. Comparing with the window function, this TNN-based design achieves reduced passband peak-to-peak deviation and smaller stopband attenuation. An innovative BPNN method is proposed to design digital FIR filter [23]. Test results indicate that this BPNN-based design achieves reduced passband

Manuscript received April 23, 2024; revised August 30, 2024.

This work was supported by National Natural Science Foundation of China under Grant 62261005 and Grant 52466010, the Scientific Research Foundation of Guizhou Province under Grant No. Qiankehezhicheng [2022]133 and Grant No. Qiankehejichu-ZK [2022]135.

Fangxin Zhu is a postgraduate student of the School of Electrical Engineering, Guizhou University, Guiyang 550025, China (e-mail: fxzhu6@163.com).

Xing Yang is an associate professor of the School of Electrical Engineering, Guizhou University, Guiyang 550025, China (corresponding author to provide phone:86-18085045638; e-mail: xyang6@gzu.edu.cn).

Jiansheng Yang is an associate professor of the School of Electrical Engineering, Guizhou University, Guiyang 550025, China (e-mail: jsyang3@gzu.edu.cn).

Jiyao Yang is a postgraduate student of the School of Electrical Engineering, Guizhou University, Guiyang 550025, China (e-mail: yaoyj0920@163.com).

Mei Zhang is an associate professor of the School of Electrical Engineering, Guizhou University, Guiyang 550025, China (e-mail: uestc_yangxing@163.com).

Shengjian Liu is a doctoral student of the School of Electronics and Information Engineering, Harbin Institute of Technology, Harbin 150001, China (e-mail: z1571299351@163.com).

peak-to-peak deviation and smaller stopband attenuation comparing to the TNN-based design and the rectangular window design.

An LSTM neural network is proposed to design digital FIR filter in this paper. In the LSTM model, the LSTM neurons utilize *sigmoid* and *tanh* activation functions to capture complex relationship between the input data and expected amplitude response, thereby enhancing learning capabilities. It can also enhance the expressiveness of the network by stacking multiple LSTM layers. In addition, adaptive parameter adjustment using optimization algorithms. In our proposed design, Adam is used for minimizing the amplitude error between the expected and practical amplitude responses. Comparing with traditional digital FIR filter design methods, this continuous optimization improves filter performance and provides greater flexibility and adaptability. The main advantages of this method include memory cells, gating mechanisms, end-to-end learning, gradient change stabilization, global optimization, generalization, and iterative optimization. In our proposed design based on LSTM neural network, these features can effectively minimize of the amplitude error between the expected and practical amplitude responses, thereby improving filter performance.

The key contributions of this paper are outlined below:

1) We propose a digital FIR filter design based on LSTM neural network. This design aims at minimizing the amplitude error between the expected and practical amplitude responses, thereby obtaining the coefficients for the digital FIR filter.

2) We conduct a comprehensive analysis of the effectiveness, mean square error (MSE), and computational complexity of our proposed design based on LSTM neural network. Comparing with BPNN-based, TNN-based, and rectangular window designs, our proposed design outperforms in terms of the passband peak-to-peak deviation, the stopband attenuation, and MSE. However, these superior performances result in higher computational complexity.

The organization of this paper is as follows. In Section II, we derive the amplitude error equation between the expected and practical amplitude responses of the digital FIR filter. Section III introduces the LSTM algorithm and the LSTM neural network model in detail. Section IV details and evaluates the simulation results. Section V provides the conclusion.

II. Amplitude Error Equation

Based on [22] and [23], the transfer function of the digital FIR filter can be written as:

$$G(z)=\sum_{n=0}^{N} g(n)z^{-n} \tag{1}$$

If the digital FIR filter satisfies $g(n)=g(N-n)$ and N denotes an even number, the entire sequence will then be symmetric with respect to $n=\frac{N}{2}$. This digital FIR filter is commonly referred to as Type I linear phase filter. In addition, if it also satisfies $g\left(\frac{N}{2}-k\right)=g\left(\frac{N}{2}+k\right)$, the frequency response

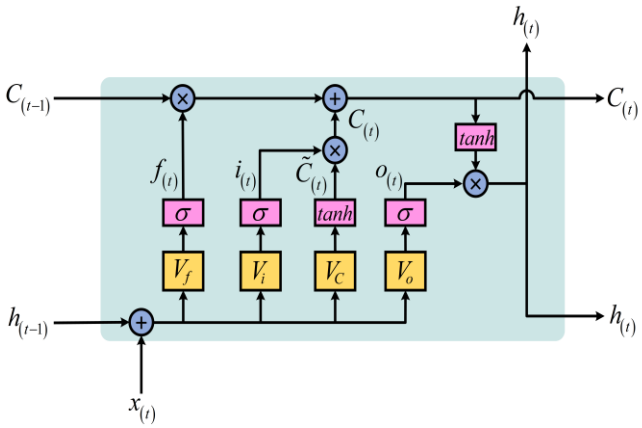

Fig. 1. LSTM neural network internal architecture.

of Type I digital FIR filter can be given by:

$$\begin{aligned} G\left(e^{j\omega}\right) &= g\left(\frac{N}{2}\right)e^{-j\frac{N}{2}\omega}+\sum_{k=1}^{\frac{N}{2}} g\left(\frac{N}{2}-k\right)e^{-j\left(\frac{N}{2}-k\right)\omega} \\ &\quad +\sum_{k=1}^{\frac{N}{2}} g\left(\frac{N}{2}+k\right)e^{-j\left(\frac{N}{2}+k\right)\omega} \end{aligned} \tag{2}$$

Based on the well-known Euler formula, Eq. (2) is thus rewritten as:

$$G\left(e^{j\omega}\right)=\left(g\left(\frac{N}{2}\right)+2\sum_{k=1}^{\frac{N}{2}} g\left(\frac{N}{2}-k\right)\cos(k\omega)\right)e^{-j\frac{N}{2}\omega} \tag{3}$$

Therefore, the practical amplitude response can be expressed as:

$$A(\omega)=\left|G\left(e^{j\omega}\right)\right|=\sum_{k=0}^{M} a_k\cos(k\omega) \tag{4}$$

where $M=\frac{N}{2}$, a_k is the coefficient of the digital FIR filter. Further,

$$\begin{cases} a_k = g(M) & k=0 \\ a_k = 2g(M-k) & k=1,2,\cdots,M \end{cases} \tag{5}$$

Assigning $L(\omega)$ as the expected amplitude response. The amplitude error $e(\omega)$ between the expected and practical amplitude responses is calculated as:

$$\begin{aligned} e(\omega) &= L(\omega)-A(\omega) \\ &= L(\omega)-\sum_{k=0}^{M} a_k\cos(k\omega),\ \omega\in W \end{aligned} \tag{6}$$

Consequently, the coefficients $(a_0,\cdots,a_k,\cdots,a_M)$ can be obtained by minimizing this amplitude error $e(\omega)$.

In the interested frequency band W, the continuous frequency variable ω is discretized into a assigned of uniformly distributed frequency points ω_r, $r=1,2,\cdots,R$, where R denotes the total number of discrete frequency points. Therefore, Eq. (6) is rewritten as:

$$e(\omega_r)=L(\omega_r)-\sum_{k=0}^{M} a_k\cos(k\omega_r),\ r=1,2,\cdots,R \tag{7}$$

Therefore, the minimization of $e(\omega)$ is converted to the minimization of $e(\omega_r)$ for achieving the coefficients $(a_0,\cdots,a_k,\cdots,a_M)$ for the digital FIR filter.

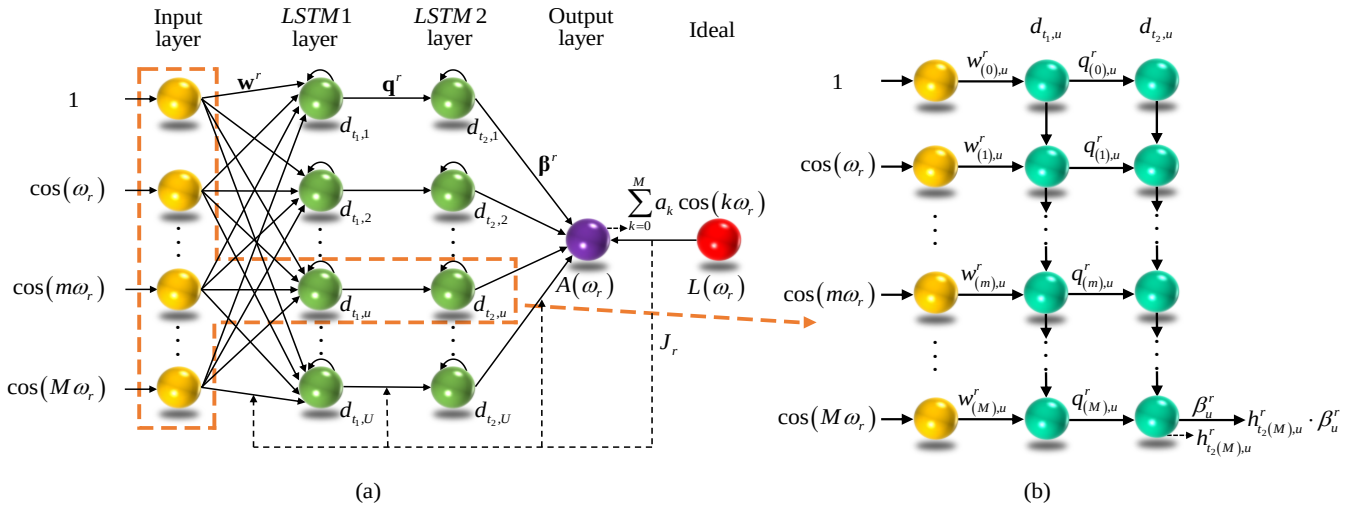

Fig. 2. Structure of LSTM neural network.

III. Digital FIR Filter Design Based on LSTM Neural Network

A. LSTM algorithm

As an extension of recurrent neural network (RNN) [24-27], LSTM [28-31] excels at predicting time series data. The LSTM model features specialized memory cells that replace the hidden layer neurons found in standard RNNs. The crucial element is the state of these memory cells. The LSTM model updates and maintains the memory cell states by processing information through its gate mechanisms. These gate mechanisms consist of the input gate, forget gate, and output gate. Fig. 1 illustrates the internal architecture of the LSTM neural network.

Firstly, in the "forget layer," the forget gate $f_{(t)}$ uses the current time step input $x_{(t)}$ and the previous time step output $h_{(t-1)}$ as input (t represents the current time step, (t-1) represents the previous time step, the initial value of $h_{(t-1)}$ is 0). The forget gate $f_{(t)}$ is calculated using the *sigmoid* activation function:

$$f_{(t)} = \sigma(V_f \boldsymbol{x}_{(t)} + V_{hf} h_{(t-1)} + b_f) \tag{8}$$

where σ denotes the *sigmoid* activation function $\sigma(x) = \frac{1}{1+e^{-x}}$, V_f and V_{hf} are the weight matrix and recurrent weight of the forget gate, respectively, and b_f represents the corresponding bias.

Secondly, in the "input layer," the input gate $i_{(t)}$ is calculated using the *sigmoid* activation function:

$$i_{(t)} = \sigma(V_i \boldsymbol{x}_{(t)} + V_{hi} h_{(t-1)} + b_i) \tag{9}$$

where V_i and V_{hi} are the weight matrix and recurrent weight of the input gate, respectively, and b_i represents the corresponding bias.

And then, the current time step candidate value $\tilde{C}_{(t)}$ is calculated using the *tanh* activation function:

$$\tilde{C}_{(t)} = tanh(V_C \boldsymbol{x}_{(t)} + V_{hC} h_{(t-1)} + b_C) \tag{10}$$

where *tanh* denotes the *tanh* activation function $tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}}$, V_C and V_{hC} are the weight matrix and recurrent weight of the candidate value, respectively, and b_c represents the corresponding bias.

The current time step cell state $C_{(t)}$ is determined by the forget gate $f_{(t)}$, the input gate $i_{(t)}$, the previous time step cell state $C_{(t-1)}$ (the initial value of $C_{(t-1)}$ is 0), and the current time step candidate value $\tilde{C}_{(t)}$:

$$C_{(t)} = f_{(t)} \bullet C_{(t-1)} + i_{(t)} \bullet \tilde{C}_{(t)} \tag{11}$$

In the "output layer," the output gate $o_{(t)}$ is calculated using the following *sigmoid* activation function:

$$o_{(t)} = \sigma(V_o \boldsymbol{x}_{(t)} + V_{ho} h_{(t-1)} + b_o) \tag{12}$$

where V_o and V_{ho} are the weight matrix and recurrent weight of the output gate, respectively, and b_o represents the corresponding bias.

Finally, the current time step cell state $C_{(t)}$ is processed by the *tanh* activation function and multiplied by the output gate $o_{(t)}$ to obtain the current time step output $h_{(t)}$:

$$h_{(t)} = o_{(t)} \bullet tanh(C_{(t)}) \tag{13}$$

B. LSTM neural network model

As described in Section I, the LSTM neural network can minimize of the amplitude error between the expected and practical parameters in specific applications. Therefore, to obtain the coefficients $(a_0, \cdots, a_k, \cdots, a_M)$ for the digital FIR filter, the LSTM neural network is employed to minimize the amplitude error $e(\omega_r)$. Table I outlines the algorithm process of the LSTM neural network.

Fig. 2(a) illustrates the structure of the LSTM neural network, while Fig. 2(b) shows a schematic of its unfolded architecture. Fig. 2 illustrates the input of the LSTM neural network as follows:

$$\begin{aligned} \mathbf{I} &= \left[1, \cos(\omega_r), \cdots, \cos(m\omega_r), \cdots, \cos(M\omega_r)\right]^T \\ &= \left[x_{(0)}, x_{(1)}, \cdots, x_{(m)}, \cdots, x_{(M)}\right]^T \end{aligned} \tag{14}$$

where $x_{(m)} = \cos(m\omega_r)$. Additionally, $\mathbf{w}^r$ represents the weight matrix connecting the input layer to the *LSTM*1 layer:

$$\mathbf{w}^r = \left[\mathbf{w}_1^r, \mathbf{w}_2^r, \cdots, \mathbf{w}_u^r, \cdots, \mathbf{w}_U^r\right] \tag{15}$$

with

$$\mathbf{w}_u^r = \left[w_{(0),u}^r, w_{(1),u}^r, \cdots, w_{(m),u}^r, \cdots, w_{(M),u}^r\right] \tag{16}$$

Here, $w_{(m),u}^r$ represents the weight connecting the *m*-th unit in the input layer to the *m*-th time step of the *u*-th LSTM unit in the *LSTM*1 layer.

$\mathbf{q}^r$ is the weight matrix connecting the *LSTM*1 layer to the *LSTM*2 layer:

$$\mathbf{q}^r = \left[\mathbf{q}_1^r, \mathbf{q}_2^r, \cdots, \mathbf{q}_u^r, \cdots, \mathbf{q}_U^r\right] \tag{17}$$

with

$$\mathbf{q}_u^r = \left[q_{(0),u}^r, q_{(1),u}^r, \cdots, q_{(m),u}^r, \cdots, q_{(M),u}^r\right] \tag{18}$$

Here, $q_{(m),u}^r$ is the weight connecting the *m*-th time step of the *u*-th LSTM unit in the *LSTM*1 layer to the *m*-th time step of the *u*-th LSTM unit in the *LSTM*2 layer.

In the *LSTM*1 layer, the *m*-th time step input of the *u*-th LSTM unit in the *LSTM*1 layer is:

$$\begin{aligned} x_{t_1(m),u}^r &= x_{(m)} \bullet w_{(m),u}^r \\ &= \cos\left(m\omega_r\right) \bullet w_{(m),u}^r \end{aligned} \tag{19}$$

When $m = 0$, the *m*-th time step input is only $x_{t_1(m),u}^r$. However, when $m = 1, 2, \cdots, M$, in addition to $x_{t_1(m),u}^r$, the *m*-th time step inputs of the *u*-th LSTM unit contain $C_{t_1(m-1),u}^r$ and $h_{t_1(m-1),u}^r$, where $C_{t_1(m-1),u}^r$ is the previous time step cell state and $h_{t_1(m-1),u}^r$ is the previous time step output.

According to the principles of the LSTM algorithm described in Section III.A, as represented by Eq. (8) to Eq. (13), the outputs at the *m*-th time step are represented by $C_{t_1(m),u}^r$ and $h_{t_1(m),u}^r$. Here, $C_{t_1(m),u}^r$ serves solely as the input at the (*m*+1)-th time step. In addition to being used as the input at the (*m*+1)-th time step, $h_{t_1(m),u}^r$ is also used to generate the *m*-th time step input $x_{t_2(m),u}^r$ for the *u*-th LSTM unit in the *LSTM*2 layer:

$$x_{t_2(m),u}^r = h_{t_1(m),u}^r \cdot q_{(m),u}^r \tag{20}$$

In the *LSTM*2 layer, similar to the *u*-th LSTM unit in the *LSTM*1 layer, the input at the *m*-th time step consists solely of $x_{t_2(m),u}^r$ when $m = 0$. However, when $m = 1, 2, \cdots, M$, in addition to $x_{t_2(m),u}^r$, the *m*-th time step inputs of the *u*-th LSTM unit in the *LSTM*2 layer include $C_{t_2(m-1),u}^r$ and $h_{t_2(m-1),u}^r$, where $C_{t_2(m-1),u}^r$ and $h_{t_2(m-1),u}^r$ denote the cell state and output of the previous time step, respectively.

The outputs $C_{t_2(m),u}^r$ and $h_{t_2(m),u}^r$ at the *m*-th time step of the *u*-th LSTM unit in the *LSTM*2 layer can be calculated using Eq. (8) – Eq. (13). $C_{t_2(m),u}^r$ and $h_{t_2(m),u}^r$ are used as inputs only at the (*m*+1)-th time step when $m = 1, 2, \cdots, M-1$.

Furthermore, $h_{t_2(M),u}^r$ denotes the output of the *u*-th LSTM unit in the *LSTM*2 layer when $m = M$.

The input x_o^r of the output layer can be calculated as:

$$x_o^r = \mathbf{h}_{t_2}^r \cdot \boldsymbol{\beta}^r \tag{21}$$

TABLE I
ALGORITHM PROCESS

Input:	Training assigned: $\left[1, \cos(\omega_r), \cdots, \cos(m\omega_r), \cdots, \cos(M\omega_r)\right]^T$
	Expected response: $L(\omega_r)$ $r = 1, 2, \cdots, R$
	Learning rate: η
Process:	1.Calculate the *m*-th time step input $x_{t_1(m),u}^r$ of the *u*-th LSTM unit based on Eq. (19).
	2.Calculate the *m*-th time step outputs $C_{t_1(m),u}^r$ and $h_{t_1(m),u}^r$ of the *u*-th LSTM unit in the *LSTM*1 layer based on Eq. (8) – Eq. (13).
	3.Generate the *m*-th time step input $x_{t_2(m),u}^r$ of the *u*-th LSTM unit in the *LSTM*2 layer based on Eq. (20).
	4.Calculate the *m*-th time step outputs $C_{t_2(m),u}^r$ and $h_{t_2(m),u}^r$ of the *u*-th LSTM unit in the *LSTM*2 layer based on Eq. (8) – Eq. (13).
	5.Obtain the output $h_{t_2(M),u}^r$ of the *u*-th LSTM unit in the *LSTM*2 layer.
	6.Calculate the input x_o^r of the output layer based on Eq. (21).
	7.Calculate $A(\omega_r)$ based on Eq. (24).
	8.Calculate the parameter J_r based on Eq. (26).
	9.Update $w_{(m),u}^r$, $q_{(m),u}^r$, β_u^r and θ^r based on Eq. (27) – Eq. (31).
	10.Combine the (*r*+1)-th LSTM neural network input with the updated $w_{(m),u}^r$, $q_{(m),u}^r$, β_u^r, and θ^r, and repeat the same process from Step (1) to Step (9) until $r = R$.
	11.When the maximum number of iterations is reached, the execution of the LSTM neural network is halted.
Purpose:	Find the coefficients of digital FIR filter

with

$$\mathbf{h}_{t_2}^r = \left[h_{t_2(M),1}^r, h_{t_2(M),2}^r, \cdots, h_{t_2(M),u}^r, \cdots, h_{t_2(M),U}^r\right] \tag{22}$$

$$\boldsymbol{\beta}^r = \left[\beta_1^r, \beta_2^r, \cdots, \beta_u^r, \cdots, \beta_U^r\right]^T \tag{23}$$

Here, β_u^r denotes the weight connecting the *u*-th LSTM unit in the *LSTM*2 layer to the output layer.

The output $A(\omega_r)$ of the output layer can be calculated as:

$$A(\omega_r) = f\left(x_o^r - \theta^r\right) \tag{24}$$

where θ^r is threshold value of the output layer and $f(\bullet)$ is the activation function.

According to the basic principle of the digital FIR filter [22], the expected amplitude response $L(\omega_r)$ can be expressed as:

$$L(\omega_r) = \begin{cases} 1 & \omega_r \text{ in the passband} \\ 0 & \omega_r \text{ in the stopband} \end{cases} \quad r = 1, 2, \cdots, R \tag{25}$$

Consequently, the parameter J_r used to update the weights in the LSTM neural network can be written as:

$$\begin{aligned} J_r &= \frac{1}{2}\left[L(\omega_r) - A(\omega_r)\right]^2 \\ &= \frac{1}{2}e^2(\omega_r) \end{aligned} \qquad r = 1, 2, \cdots, R \tag{26}$$

Using Adam as the optimizer. To achieve the amplitude error minimization, $w_{(m),u}^r$, $q_{(m),u}^r$, β_u^r, and θ^r should be updated as:

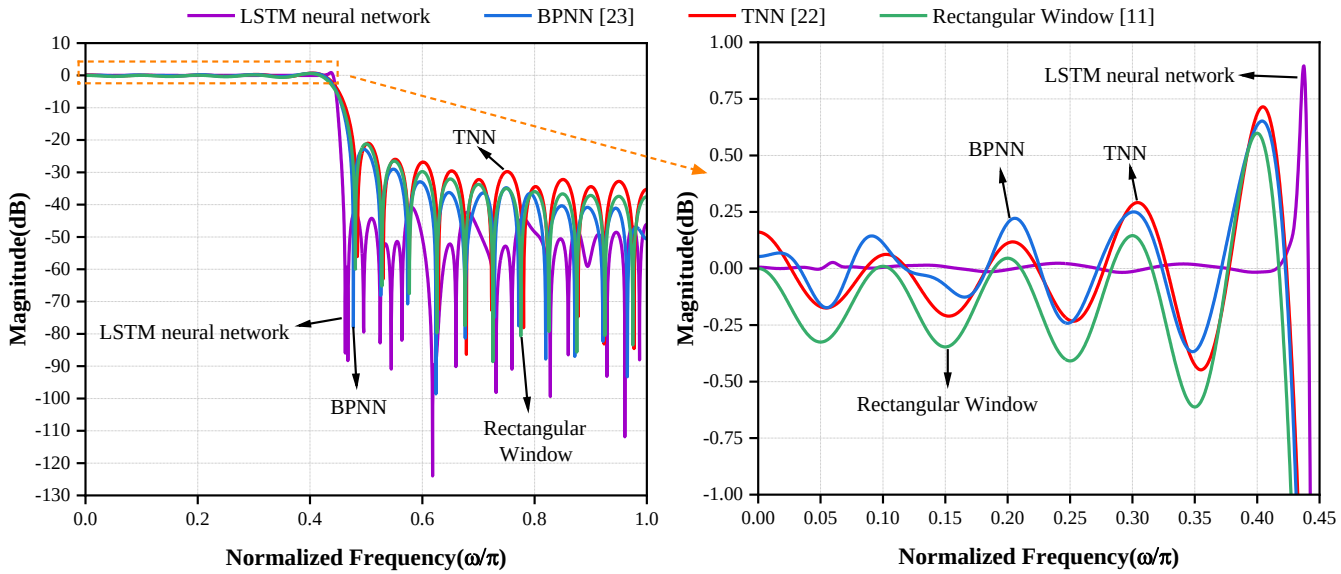

Fig. 3. Frequency response of low-pass digital FIR filter design using various methods.

$$\begin{cases} \Delta w^r_{(m),u} + w^r_{(m),u} \to w^r_{(m),u} \\ \Delta q^r_{(m),u} + q^r_{(m),u} \to q^r_{(m),u} \\ \Delta \beta^r_u + \beta^r_u \to \beta^r_u \\ \Delta \theta^r + \theta^r \to \theta^r \end{cases} \quad \begin{array}{l} r = 1,2,\cdots,R \\ m = 0,1,\cdots,M \\ u = 1,2,\cdots,U \end{array} \tag{27}$$

with

$$\Delta w^r_{(m),u} = -\eta \frac{\partial J_r}{\partial h^r_{t_1(m),u}} \frac{\partial h^r_{t_1(m),u}}{\partial x^r_{t_1(m),u}} \frac{\partial x^r_{t_1(m),u}}{\partial w^r_{(m),u}} \tag{28}$$

$$\Delta q^r_{(m),u} = -\eta \frac{\partial J_r}{\partial h^r_{t_2(m),u}} \frac{\partial h^r_{t_2(m),u}}{\partial x^r_{t_2(m),u}} \frac{\partial x^r_{t_2(m),u}}{\partial q^r_{(m),u}} \tag{29}$$

$$\Delta \beta^r_u = -\eta \frac{\partial J_r}{\partial A(\omega_r)} \frac{\partial A(\omega_r)}{\partial x^r_o} \frac{\partial x^r_o}{\partial \beta^r_u} \tag{30}$$

$$\Delta \theta^r = -\eta \frac{\partial J_r}{\partial A(\omega_r)} \frac{\partial A(\omega_r)}{\partial \theta^r} \tag{31}$$

where η is the learning rate. The updated values $w^r_{(m),u}$, $q^r_{(m),u}$, β^r_u, and θ^r will be utilized in combination with the input of the (r+1)-th LSTM neural network. This process is repeated as described in Section III.B until $r = R$. Upon reaching the maximum number of iterations, the LSTM neural network stops, allowing for the determination of the coefficients for the digital FIR filter.

IV. Simulation Results

The performance of our proposed design of digital FIR filter based on LSTM neural network is verified using serval different design examples. In these design examples, the digital FIR filter order N is assigned to 40. The number of discrete frequency points R is assigned to 2837. Furthermore, the activation function $f(\bullet)$ and the learning rate η are assigned to $f(\bullet) = \text{Linear}$ and $\eta = 0.1$, respectively.

A. Analysis of effectiveness

Design example 1: Low-pass digital FIR filter

In this design example, the passband and stopband cutoff frequencies are individually assigned to 0.45π and 0.55π.

Fig. 3 illustrates the frequency response of the low-pass digital FIR filter design using various methods such as LSTM neural network, BPNN, TNN, and rectangular window. The passband peak-to-peak deviation δ_{p-p}, the maximum stopband attenuation $\delta_{s\max}$, and the minimum stopband attenuation $\delta_{s\min}$ are summarized in Table II. It can be known from Table II and Fig. 3 that δ_{p-p} of our proposed design based on LSTM neural network is 0.9132 dB. However, δ_{p-p} of BPNN-based, TNN-based, and rectangular window designs are 1.0195 dB, 1.1629 dB, and 1.2104 dB, respectively. These values are all higher than δ_{p-p} obtained by our proposed design based on LSTM neural network.

Additionally, the maximum stopband attenuation $\delta_{s\max}$ achieved by these various methods such as LSTM neural network, BPNN, TNN, and rectangular window are -123.9678 dB, -98.5941 dB, -86.3435 dB, and -88.6180 dB, respectively. Moreover, the minimum stopband attenuation $\delta_{s\min}$ of these four optimal designs are individually -40.8170 dB, -29.0640 dB, -25.9529 dB, and -26.5051 dB. Our proposed design based on LSTM neural network achieves lower maximum and minimum stopband attenuations compared to the other three optimal designs.

Design example 2: High-pass digital FIR filter

In this design example, the passband and stopband cutoff frequencies are individually assigned to 0.55 π and 0.45 π.

Fig. 4 illustrates the frequency response of the high-pass digital FIR filter design using various methods such as LSTM neural network, BPNN, TNN, and rectangular window. It can be known from Table II and Fig. 4 that δ_{p-p} of our proposed design based on LSTM neural network is 0.3280 dB. However, δ_{p-p} of BPNN-based, TNN-based, and

TABLE II
SUMMARY OF THE PASSBAND PEAK-TO-PEAK DEVIATION, THE MAXIMUM AND MINIMUM STOPBAND ATTENUATION

Design Example	Method	δ_{p-p}	$\delta_{s\ \max}$	$\delta_{s\ \min}$
Low-pass digital FIR filter	**LSTM neural network**	**0.9132 dB**	**-123.9678 dB**	**-40.8170 dB**
	BPNN [23]	1.0195 dB	-98.5941 dB	-29.0640 dB
	TNN [22]	1.1629 dB	-86.3435 dB	-25.9529 dB
	Rectangular window [11]	1.2104 dB	-88.6180 dB	-26.5051 dB
High-pass digital FIR filter	**LSTM neural network**	**0.3280 dB**	**-123.6654 dB**	**-50.2704 dB**
	BPNN [23]	0.9195 dB	-105.5801 dB	-28.5215 dB
	TNN [22]	1.1232 dB	-91.4495 dB	-26.0572 dB
	Rectangular window [11]	1.2104 dB	-88.6180 dB	-26.5051 dB
Band-pass digital FIR filter	**LSTM neural network**	**0.1004 dB**	**-117.5967 dB**	**-36.0197 dB**
	BPNN [23]	1.2653 dB	-90.8968 dB	-22.4479 dB
	TNN [22]	1.4228 dB	-87.2071 dB	-19.9612 dB
	Rectangular window [11]	1.3245 dB	-82.2671 dB	-20.9115 dB
Band-stop digital FIR filter	**LSTM neural network**	**0.2524 dB**	**-114.7023 dB**	**-37.0812 dB**
	BPNN [23]	0.9329 dB	-100.0401 dB	-22.3030 dB
	TNN [22]	1.1210 dB	-92.1125 dB	-21.7347 dB
	Rectangular window [11]	1.2221 dB	-92.9414 dB	-21.0732 dB

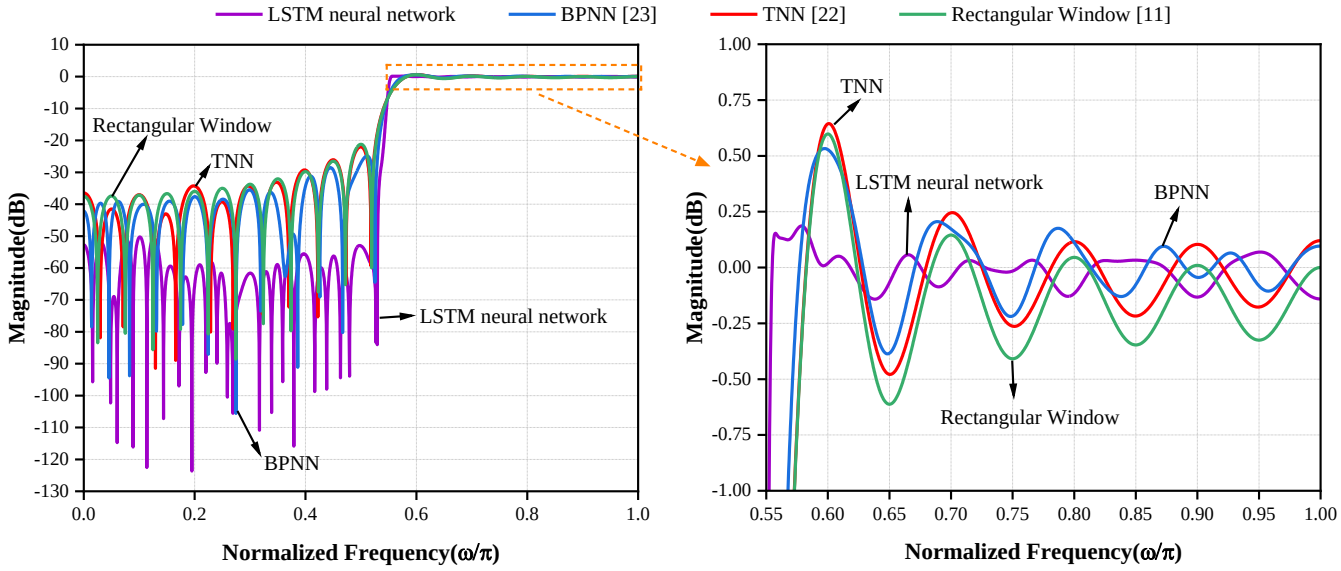

Fig. 4. Frequency response of high-pass digital FIR filter design using various methods.

rectangular window designs are 0.9195 dB, 1.1232 dB, and 1.2104 dB, respectively. Comparing with the other three designs, our proposed design based on LSTM neural network achieves smaller passband peak-to-peak deviation.

In addition, $\delta_{s\ \max}$ achieved by these various methods such as LSTM neural network, BPNN, TNN, and rectangular window are -123.6654 dB, -105.5801 dB, -91.4495 dB and -88.6180 dB, respectively. And $\delta_{s\ \min}$ of these four optimal designs are individually -50.2704 dB, -28.5215 dB, -26.0572 dB and -26.5051 dB. It is evident that our proposed design based on LSTM neural network achieves lower maximum and minimum stopband attenuations compared to the other three optimal designs.

Design example 3: Band-pass digital FIR filter

In this design example, the passband cutoff frequencies are individually assigned to 0.3 π and 0.7 π, and the stopband cutoff frequencies are individually assigned to 0.25 π and 0.75 π.

Fig. 5 illustrates the frequency response of the band-pass digital FIR filter designed using various methods such as LSTM neural network, BPNN, TNN, and rectangular window. Table II and Fig. 5 show that δ_{p-p} of our proposed design based on LSTM neural network is 0.1004 dB. However, δ_{p-p} of BPNN-based, TNN-based, and rectangular window designs are 1.2653 dB, 1.4228 dB, and 1.3245 dB, respectively. These values are all higher than δ_{p-p} obtained by our proposed design based on LSTM neural network.

Furthermore, $\delta_{s\ \max}$ achieved by these various methods such as LSTM neural network, BPNN, TNN, and rectangular window are -117.5967 dB, -90.8968 dB, -87.2071 dB, and -82.2671 dB, respectively. Moreover, $\delta_{s\ \min}$ of these four optimal designs are individually -36.0197 dB, -22.4479 dB, -19.9612 dB, and -20.9115 dB. Our proposed design based on LSTM neural network achieves lower maximum and minimum stopband attenuations compared to the other three optimal designs.

Design example 4: Band-stop digital FIR filter

In this design example, the passband cutoff frequencies are individually assigned to 0.25 π and 0.75 π, and the stopband cutoff frequencies are individually assigned to 0.3 π and 0.7 π.

Fig. 6 illustrates the frequency response of the band-stop digital FIR filter design using various methods such as LSTM neural network, BPNN, TNN, and rectangular window. Table II and Fig. 6 illustrate that δ_{p-p} of our proposed design based on LSTM neural network is 0.2524 dB. However, δ_{p-p} of BPNN-based, TNN-based, and rectangular window designs are 0.9329 dB, 1.1210 dB, and 1.2221 dB, respectively.

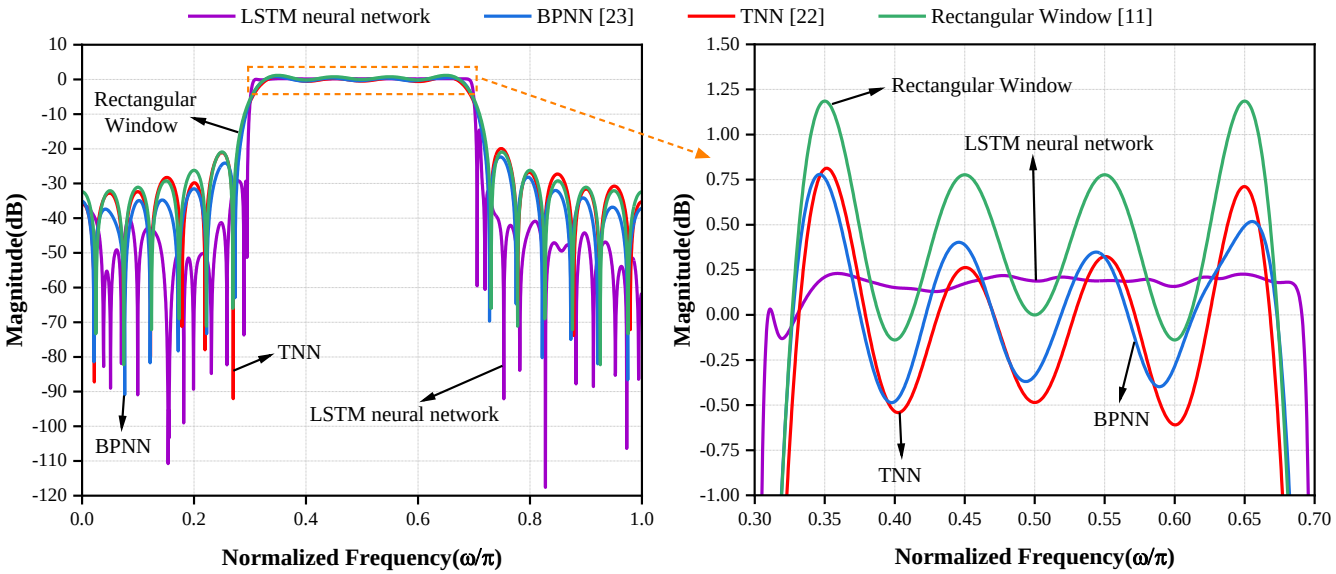

Fig. 5. Frequency response of band-pass digital FIR filter design using various methods.

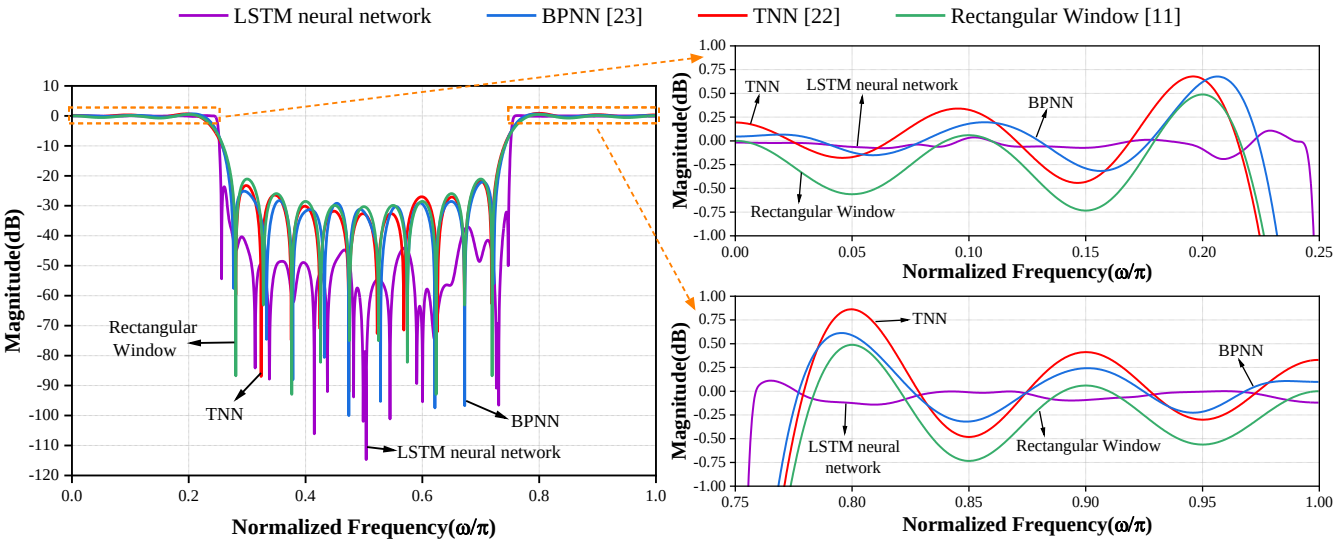

Fig. 6. Frequency response of band-stop digital FIR filter design using various methods.

Comparing with the other three designs, our proposed design based on LSTM neural network achieves smaller passband peak-to-peak deviation.

In addition, $\delta_{s\max}$ of these various methods such as LSTM neural network, BPNN, TNN, and rectangular window are -114.7023 dB, -100.0401 dB, -92.1125 dB, and -92.9414 dB, respectively. And $\delta_{s\min}$ of these four optimal designs are individually -37.0812 dB, -22.3030 dB, -21.7347 dB, and -21.0732 dB. Comparing with the other three designs, our proposed design based on LSTM neural network achieves lower maximum and minimum stopband attenuations.

It is evident from Design example 1, 2, 3, and 4 that our proposed design based on LSTM neural network reduces passband peak-to-peak deviation, decreases maximum and minimum stopband attenuations compared to BPNN-based, TNN-based, and rectangular window designs. Additionally, the effectiveness of these four digital FIR filter designs is ranked from best to worst as follows:

$$\begin{aligned}&\text{LSTM neural network} > \text{BPNN}\ [23]\\ &\qquad\qquad > \text{TNN}\ [22]\\ &\qquad\qquad > \text{Rectangular Window}\ [11]\end{aligned} \tag{32}$$

B. Analysis of MSE and computational complexity

The passband bandwidth for low-pass, high-pass, and band-pass digital FIR filters are assigned to $\Omega_p = \left[\omega_{p1}, \omega_{p2}\right]$, the MSE($\text{MSE}_\text{P}$) of the passband is expressed as:

$$\text{MSE}_\text{P} = \frac{\sum_{r=\frac{\omega_{p1}}{\pi}R}^{\frac{\omega_{p2}}{\pi}R}\left(L(\omega_r) - A(\omega_r)\right)^2}{\frac{\omega_{p2}-\omega_{p1}}{\pi}R} \tag{33}$$

The passband bandwidth for the band-stop digital FIR filter is assigned to $\Omega_p = \left[\omega_{p1}, \omega_{p2}\right] \cup \left[\omega_{p3}, \omega_{p4}\right]$, the MSE($\text{MSE}_\text{P}$) of the passband can be expressed as:

TABLE III
SUMMARY OF MSE AND COMPUTATIONAL COMPLEXITY

Design Example	Method	MSE_P	MSE_S	CPU running time
Low-pass digital FIR filter	**LSTM neural network**	**0.0041**	**1.2543×10^{-5}**	69.7652s
	BPNN [23]	0.0048	0.0001	6.5374s
	TNN [22]	0.0061	0.0005	5.2213s
	Rectangular window [11]	0.9419	0.0001	**0.0010s**
High-pass digital FIR filter	**LSTM neural network**	**0.0003**	**1.4490×10^{-6}**	63.1727s
	BPNN [23]	0.0042	0.0001	6.1184s
	TNN [22]	0.0058	0.0002	5.5392s
	Rectangular window [11]	0.9426	0.0001	**0.0010s**
Band-pass digital FIR filter	**LSTM neural network**	**0.0026**	**2.6413×10^{-5}**	62.3076s
	BPNN [23]	0.0108	0.0004	6.0137s
	TNN [22]	0.0144	0.0009	5.7982s
	Rectangular window [11]	1.1049	0.0005	**0.0010s**
Band-stop digital FIR filter	**LSTM neural network**	**0.0032**	**1.6057×10^{-5}**	67.6546s
	BPNN [23]	0.0080	0.0007	7.5003s
	TNN [22]	0.0119	0.0009	5.2304s
	Rectangular window [11]	0.9114	0.0006	**0.0010s**

$$\mathrm{MSE_P}=\frac{\sum_{r=\frac{\omega_{p1}}{\pi}R}^{\frac{\omega_{p2}}{\pi}R}\left(L(\omega_r)-A(\omega_r)\right)^2+\sum_{r=\frac{\omega_{p3}}{\pi}R}^{\frac{\omega_{p4}}{\pi}R}\left(L(\omega_r)-A(\omega_r)\right)^2}{\frac{\left(\omega_{p2}-\omega_{p1}\right)+\left(\omega_{p4}-\omega_{p3}\right)}{\pi}R} \quad (34)$$

The stopband bandwidth for low-pass, high-pass, and band-stop digital FIR filters are assigned to $\Omega_s=[\omega_{s1},\omega_{s2}]$, the MSE($MSE_S$) of the stopband can be expressed as:

$$\mathrm{MSE_S}=\frac{\sum_{r=\frac{\omega_{s1}}{\pi}R}^{\frac{\omega_{s2}}{\pi}R}\left(L(\omega_r)-A(\omega_r)\right)^2}{\frac{\omega_{s2}-\omega_{s1}}{\pi}R} \quad (35)$$

The stopband bandwidth for the band-pass digital FIR filter is assigned to $\Omega_s=[\omega_{s1},\omega_{s2}]\cup[\omega_{s3},\omega_{s4}]$, the MSE($MSE_S$) of the stopband can be expressed as:

$$\mathrm{MSE_S}=\frac{\sum_{r=\frac{\omega_{s1}}{\pi}R}^{\frac{\omega_{s2}}{\pi}R}\left(L(\omega_r)-A(\omega_r)\right)^2+\sum_{r=\frac{\omega_{s3}}{\pi}R}^{\frac{\omega_{s4}}{\pi}R}\left(L(\omega_r)-A(\omega_r)\right)^2}{\frac{\left(\omega_{s2}-\omega_{s1}\right)+\left(\omega_{s4}-\omega_{s3}\right)}{\pi}R} \quad (36)$$

In terms of computational efficiency, on a computer with 2.8-GHz computer process unit (CPU), the optimal designs of low-pass, high-pass, band-pass, and band-stop digital FIR filters based on LSTM neural network are taking 69.7652s, 63.1727s, 62.3076s, and 67.6546s, respectively. The times consumed for the BPNN-based optimal design are 6.5374s, 6.1184s, 6.0137s, and 7.5003s, respectively. For the TNN-based optimal design, the times are 5.2213s, 5.5392s, 5.7982s and 5.2304s, respectively. The rectangular window design consistently takes only 0.0010s.

Table III presents the parameters of passband MSE(MSE_P), stopband MSE(MSE_S), and computational complexity (i.e. CPU running time) for the optimal designs based on LSTM neural network, BPNN, TNN, and rectangular window. Table III shows that the passband and stopband MSE of our proposed design based on LSTM neural network are smaller than those of the other three methods. However, this improved performance comes at the cost of increased CPU running time.

V. Conclusion

An LSTM neural network is proposed to design digital FIR filter in this paper. In the LSTM model, the LSTM neurons utilize *sigmoid* and *tanh* activation functions to capture complex relationship between the input data and expected amplitude response, thereby enhancing learning capabilities. It can also enhance the expressiveness of the network by stacking multiple LSTM layers. In addition, adaptive parameter adjustment using optimization algorithms. In our proposed design, Adam is used for minimizing the amplitude error between the expected and practical amplitude responses. Compared to traditional digital FIR filter design methods, this continuous optimization improves filter performance and provides greater flexibility and adaptability. The main advantages of this method include memory cells, gating mechanisms, end-to-end learning, gradient change stabilization, global optimization, generalization, and iterative optimization. In our proposed design based on LSTM neural network, these features can effectively minimize the amplitude error between the expected and practical amplitude responses, thereby improving filter performance.

The simulation results indicate that our proposed design based on LSTM neural network achieves peak-to-peak deviations of 0.9132 dB for the low-pass filter, 0.3280 dB for the high-pass filter, 0.1004 dB for the band-pass filter, and 0.2524 dB for the band-stop filter. Additionally, the stopband attenuations for these four digital FIR filters are as follows: [-123.9678 dB, -40.8170 dB], [-123.6654 dB, -50.2704 dB], [-117.5967 dB, -36.0197 dB], and [-114.7023 dB, -37.0812 dB]. The passband MSE of low-pass, high-pass, band-pass, and band-stop digital FIR filters are 0.0041, 0.0003, 0.0026, and 0.0032, respectively. The stopband MSE of these digital FIR filters are 1.2543×10^{-5}, 1.4490×10^{-6}, 2.6413×10^{-5}, and 1.6057×10^{-5}, respectively. Compared to the other three digital FIR filter designs, our proposed design based on LSTM neural network exhibits superior performance. However, these improvements result in higher computational complexity. Thus, our proposed digital FIR filter design based on LSTM neural network is suitable for applications where increased computational complexity is acceptable in exchange for improved effectiveness.

In future research, we plan to achieve the hardware implementation of digital FIR filter using Intel's Agilex™ 7

FPGA. This will enable a deeper analysis of the performance of our proposed digital FIR filter design based on long short-term memory neural network.

REFERENCES

[1] Y. Yan, W. Wang, S. Zhang, and J. Cai, "Range-ambiguous clutter characteristics in airborne FDA radar," *Signal Processing*, vol.170,2020. pp.107407.

[2] W. Zhao, S. Tian, G. Guo, J. You, Q. Wu, and K. Liu, "An arbitrary waveform synthesis structure with high sampling rate and low spurious," *Metrology and Measurement Systems*, vol.29, no.1, 2022. pp.159-173.

[3] K. Liu, S. Tian, G. Guo, and Y. Xiao, "Precisely synchronous and cascadable multi-channel arbitrary waveform generator," *Review of Scientific Instruments*, vol.88, no.3, 2017. pp.035110.

[4] O. V. Chernoyarov, V. A. Ivanov, A. V. Salnikova, and M. A. Slepneva, "Intra-Period Signal Processing in a Synthetic Aperture Radar," *Engineering Letters*, vol.30, no.1, 2022. pp.67-72.

[5] L. Huang, S. Tian, K. Liu, G. Guo, Y. Xiao, W. Zhao, and X. Yang, "The design of a wide bandwidth time marker generator," *Review of Scientific Instruments*, vol.89, no.11, 2018. pp.115103.

[6] W. Zhao, S. Tian, H. Chen, Y. Xiao, Q. Wu, and K. Liu, "Design of An All-Pass Phase Compensation Filter Based on Modified Genetic algorithm in FI-DAC," in *2022 IEEE AUTOTESTCON*, 2022. pp.1-7.

[7] Y. Xiao, W. Mo, K. Liu, W. Zhao, and C. Hu, "Selecting the optimal sampling rate for the waveform generator with a variable clock," *Digital Signal Process*, vol.123, 2022. pp.103399.

[8] S. N. Mirebrahimi, and F. Merrikh-Bayat, "Programmable discrete-time type I and type II FIR filter design on the memristor crossbar structure," *Analog Integrated Circuits and Signal Processing*, vol.79, 2014. pp.529-541

[9] P. H. Wang, B. Y. Yu, and P. N. Chen, "Type II, III, and IV linear-phase FIR structures based on cardinal filters," in *2019 IEEE Transactions on Circuits and Systems II: Express Briefs*, vol.66, no.11, 2019. pp.1920-1924.

[10] I. A. Sulaiman *et al*, "Design, comparison and analysis of low pass FIR filter using window techniques method," *Materials Today: Proceedings*, vol.49, no.8, 2022. pp.3117-3121.

[11] M. G. Chen, "Design a FIR filter using window function," *Applied Mechanics and Materials*, vol.65, 2011. pp.435-438.

[12] V. Kumar, S. Bangar, S. N. Kumar, and S. Jit, "Design of effective window function for FIR filters." in *2014 IEEE International Conference on Advances in Engineering & Technology Research (ICAETR-2014)*, 2014. pp.1-5.

[13] S. K. Mitra, "Digital Signal Processing: A Computer-Based Approach," Fourth Edition, McGRAW-Hill 2011.

[14] M. Khan, and S. Agha, "Least squares linear phase FIR filter design and its VLSI implementation," *Analog Integrated Circuits and Signal Processing*, vol.105, 2020. pp.99-109.

[15] X. Yang, H. Wang, and K. Liu, "Estimation and compensation methods of time delay and phase offassigned in hybrid filter bank DACs," *Electronics Letters*, vol.54, Issue 13, 2018. pp.806-808.

[16] Q. Huang, "Energy-Efficient Smart Building Driven by Emerging Sensing, Communication, and Machine Learning Technologies," *Engineering Letters*, vol.26. no.3, 2018. pp.320-322.

[17] M. A. A. R. Asif et al, "Performance Evaluation and Comparative Analysis of Different Machine Learning Algorithms in Predicting Cardiovascular Disease," *Engineering Letters*, vol.29, no.2, 2021. pp.731-741.

[18] S. Verma, S. P. Sahu, and T. P. Sahu, "Stock Market Forecasting with Different Input Indicators using Machine Learning and Deep Learning Techniques: A Review," *Engineering Letters*, vol.31, no.1, 2023. pp.213-229.

[19] Y. Chen, "Voltages prediction algorithm based on LSTM recurrent neural network," *Optik*, vol.220, 2020. pp.164869.

[20] A. Pranolo, Y. Mao, Y. Tang, and A. P. Wibawa, "A long short term memory implemented for rainfall forecasting," in *2020 IEEE 6th International Conference on Science in Information Technology (ICSITech)*, 2020. pp.194-197.

[21] X. Qing, and Y. Niu, "Hourly day-ahead solar irradiance prediction using weather forecasts by LSTM," *Energy*, vol.148, 2018. pp.461-468.

[22] C. C. Tseng, and S. L. Lee, "A Supervised Learning Method for the Design of Linear Phase FIR Digital Filter Using Keras," in *2019 IEEE International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS)*, 2019. pp.1-2.

[23] J. Yang, H. Yang, X. Yang, and J. Yang, "Optimal design of digital FIR filters based on back propagation neural network," *IEICE Electronics Express*, vol.20, no.1, 2023. pp.20220491-20220491.

[24] Q. Zhu, X. Zhou, J. Tan, and L. Guo, "Knowledge base reasoning with convolutional-based recurrent neural networks," *IEEE Transactions on Knowledge and Data Engineering*, vol.33, Issue 5, 2019. pp.2015-2028.

[25] R. DiPietro, and G. D. Hager, "Deep learning: RNNs and LSTM," in *Handbook of medical image computing and computer assisted intervention*. Academic Press, 2020. pp.503-519.

[26] S. A. Khaliq, and M. A. Basiri M, "An Efficient VLSI Architecture of Recurrent Neural Network," in *2022 IEEE 6th Conference on Information and Communication Technology (CICT)*, 2020. pp.1-6.

[27] M. W. Nadeem, H. G. Goh, Y. Aun, and V. Ponnusamy, "A Recurrent Neural Network based Method for Low-Rate DDoS Attack Detection in SDN," in *2022 IEEE 3rd International Conference on Artificial Intelligence and Data Sciences (AiDAS)*, 2022. pp.13-18.

[28] T. N. Sainath, O. Vinyals, A. Senior, and H. Sak, "Convolutional, Long Short-Term Memory, fully connected Deep Neural Networks," in *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, 2015. pp.4580-4584.

[29] M. Bkassiny, "A Deep Learning-based Signal Classification Approach for Spectrum Sensing using Long Short-Term Memory (LSTM) Networks," in *2022 IEEE 6th International Conference on Information Technology, Information Systems and Electrical Engineering (ICITISEE)*, 2022. pp.667-672.

[30] K. Nitesh, Y. Abhiram, R. K. Teja, and S. Kavitha, "Weather Prediction using Long Short Term Memory (LSTM) model," in *2023 IEEE 5th International Conference on Smart Systems and Inventive Technology (ICSSIT)*, 2023. pp.1-6.

[31] H. Abbasimehr, M. Shabani, and M. Yousefi, "An optimal model using LSTM network for demand forecasting," *Computers & industrial engineering*, vol.143, 2020. pp.106435.