Low Computational Complexity Design of Digital FIR Filters Based on ABC-LSTM

Jiayin Ye, Yisheng Yan, Fangxin Zhu, Xing Yang, Jiansheng Yang and Ke Liu

***Abstract*—This paper presents a novel approach that combines artificial bee colony (ABC) optimization with long short-term memory (LSTM) neural networks for designing digital finite impulse response (FIR) filters. Initially, a mathematical model of the digital FIR filter is developed to evaluate the actual and desired amplitude responses, and then the error between them is calculated. An LSTM neural network is introduced to minimize the amplitude error. To address the computational complexity of the traditional LSTM design, the ABC optimization method is applied to reduce key parameters, such as the filter order, number of hidden layer neurons, and iterations. This ensures that the specified limits for passband fluctuation and stopband suppression are met simultaneously. Finally, the optimized parameters are applied in the LSTM neural network to obtain the optimal coefficients for the digital FIR filter. Several design examples are employed to assess the performance of the proposed ABC-LSTM model. The simulation results demonstrate that, when compared to four alternative methods, the proposed ABC-LSTM design exhibits reduced passband fluctuation and improved stopband suppression. Moreover, the proposed ABC-LSTM design reduces the filter order, the number of hidden neurons in the LSTM network, and the number of iterations, resulting in lower computational complexity (CPU running time) compared to the traditional LSTM network design.**

***Index Terms*—Artificial bee colony, Long short-term memory, Low computational complexity, Digital FIR filter**

I. Introduction

IN digital signal processing, finite impulse response (FIR) filters are widely implemented due to their strictly linear phase characteristics and inherent stability advantages over infinite impulse response (IIR) filters [1], [2], [3], [4], [5]. FIR filters serve various purposes, including spectral analysis [6], signal shaping [7], band selection [8], and general filtering [9]. Among the four filter types, including Type I, Type II, Type III, and Type IV, only Type I filters can be used to design low-pass, high-pass, band-pass, and band-stop filters [10]. As the demand for advanced signal processing grows, optimizing Type I FIR filters has emerged as a significant challenge.

The window function method is commonly used in digital filter design [11], [12], [13]. Although it is straightforward and easy to implement, it may not always achieve the performance standards required for ideal filters. At present, Type I FIR filters are designed by minimizing the amplitude errors between the actual and desired responses. Well-known optimization techniques include the least squares method [14] and non-iterative weighted least squares algorithms [15].

Machine learning typically aims to reduce estimation errors between actual and desired values in various applications [16], [17]. For example, X. H. Le et al. introduced a runoff prediction method using long short-term memory (LSTM) neural networks, focusing on reducing the gap between actual and desired runoff [18]. In a similar approach, H. Zheng et al. utilized LSTM neural networks to minimize the discrepancy between actual and desired air pollutant concentrations, improving forecasting accuracy [19]. L. Wang et al. proposed a method for short-term power forecasting in photovoltaic systems, based on LSTM neural networks, which effectively minimized the estimation error between actual and desired power generation [20]. While these studies do not specifically focus on machine learning for designing digital FIR filters, they offer valuable insights for related research in this field.

Recent research has explored the use of machine learning techniques to optimize digital FIR filters. One study suggested using a traditional neural network (TNN) to enhance the design of digital FIR filters by minimizing the amplitude error between the actual and desired responses. In contrast to the window function method, the TNN-based design demonstrates improved passband fluctuation and enhanced stopband suppression [21]. Another method employed a backpropagation neural network (BPNN) to enhance the optimization of FIR filter design [22]. Test results demonstrate that it outperforms both the TNN-based design and the rectangular window method in reducing passband fluctuations and enhancing stopband suppression.

This paper introduces a method that combines the artificial bee colony (ABC) algorithm with long short-term memory (LSTM) neural networks to optimize the design of digital finite impulse response (FIR) filters. First, a mathematical model for the digital FIR filter is developed to obtain both the actual and desired amplitude responses, followed by the calculation of the amplitude error between them. Next, the Long Short-Term Memory (LSTM) neural network [23] is utilized to reduce the amplitude error. To minimize the

Manuscript received March 15, 2025; revised June 14, 2025.

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

Jiayin Ye is an undergraduate student in the School of Electrical Engineering, Guizhou University, Guiyang 550025, China (e-mail: jiayinye2@163.com).

Yisheng Yan is a postdoctoral researcher in the School of Information and Communication Engineering, University of Electronic Science and Technology of China, Chengdu, P. R. China (e-mail: yishengyan0308@gmail.com).

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

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

Ke Liu is a professor of the School of Automation Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China (e-mail: liuke@uestc.edu.cn).

computational complexity of LSTM neural network design, the artificial bee colony (ABC) algorithm is used to optimize the filter order, the number of neurons in the LSTM hidden layer, and the number of iterations, while ensuring predefined constraints on passband fluctuation and stopband suppression are met. The artificial bee colony (ABC) algorithm [24], [25] demonstrates robust global search capabilities, effectively avoiding local optima, while the LSTM neural network processes temporal sequences to optimize the filter's dynamic properties. Ultimately, the optimized parameters are used in the LSTM neural network to derive the optimal coefficients for the digital finite impulse response (FIR) filter.

The key contributions of this paper are outlined below:

(1) A novel approach for digital FIR filter design is proposed, utilizing the ABC-LSTM method. The objective is to reduce the amplitude error between the actual and desired responses, thereby generating the optimal filter coefficients.

(2) The ABC algorithm is introduced to optimize the filter order, the number of neurons in the LSTM neural network's hidden layers, and the number of iterations, ensuring that the predefined limits for passband fluctuation and stopband suppression are met simultaneously. This method simplifies the computational demands involved in designing the LSTM neural network.

(3) A thorough comparison is conducted to evaluate the performance, mean square error (MSE), and computational efficiency of filters designed using different methods, including TNN, BPNN, the rectangular window method, LSTM networks, and ABC-LSTM.

(4) The simulation results show that the proposed ABC-LSTM design outperforms the other four methods in passband fluctuation, stopband suppression, and MSE, while also reducing computational complexity.

The structure of this paper is as follows: Section II presents an analysis of amplitude errors in digital FIR filters. Section III outlines the optimization framework based on LSTM neural networks for determining optimal filter coefficients. Section IV presents an analysis of the simulation results, highlighting the performance enhancements attained by the proposed method. Section V then concludes with a summary of the key findings.

II. Amplitude error

To optimize filter performance, it is essential to reduce the amplitude error between the actual and desired responses.

The system function of a digital finite impulse response filter, as derived from the analytical approaches in [21] and [22], is mathematically represented as:

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

If a digital FIR filter satisfies the symmetry condition $H(z)=H(N-n)$ and N is even, the entire sequence will be symmetric about $n=\frac{N}{2}$. The digital FIR filter in question is predominantly recognized as a Type I filter, distinguished by its linear phase properties. If the condition $H(\frac{N}{2}-k)=H(\frac{N}{2}+k)$ is still satisfied and $z=e^{j\omega}$ is given, the frequency response can be expressed as:

$$H(e^{j\omega})=h\left(\frac{N}{2}\right)e^{-j\frac{N}{2}\omega}+\sum_{k=1}^{\frac{N}{2}}h\left(\frac{N}{2}-k\right)e^{-j\left(\frac{N}{2}-k\right)\omega} +\sum_{k=1}^{\frac{N}{2}}h\left(\frac{N}{2}+k\right)e^{-j\left(\frac{N}{2}+k\right)\omega} \tag{2}$$

Using Euler's formula, Eq. (2) can be expressed as:

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

When the filter coefficient is specified as a_k, with:

$$M=\frac{N}{2},a_0=h(M),a_k=2h(M-k),k=1,2,\ldots,M \tag{4}$$

The amplitude response can be expressed as:

$$A(\omega)=a_0+\sum_{k=1}^{M}a_k\cos(k\omega)=\sum_{k=0}^{M}a_k\cos(k\omega) \tag{5}$$

The amplitude error is defined as:

$$e(\omega)=I(\omega)-A(\omega)=I(\omega)-\sum_{k=0}^{M}a_k\cos(k\omega),\omega\in W \tag{6}$$

Here, $I(\omega)$ corresponds to the desired amplitude response of the digital FIR filter. The coefficients $(a_0,\ldots,a_k,\ldots,a_M)$ are therefore determined by minimizing the amplitude error $e(\omega)$.

The continuous frequency variable ω is sampled into uniformly spaced points ω_l within the frequency range W. Here, $l=1,2,\cdots,L$, where L represents the total number of discrete frequency points. Consequently, Eq. (6) can be reformulated as:

$$e(\omega_l)=I(\omega_l)-A(\omega_l)=I(\omega_l)-\sum_{k=0}^{M}a_k\cos(k\omega_l),l=1,2,\ldots,L \tag{7}$$

Thus, the problem of minimizing $e(\omega)$ is transformed into the problem of minimizing $e(\omega_l)$.

III. Design methodology for digital FIR filter using ABC-LSTM

A. Digital FIR filter optimization design via LSTM neural network

As a recurrent neural network with a gating mechanism, the long short-term memory neural network (LSTM) [26], [27], [28], [29] effectively addresses the vanishing gradient problem of traditional recurrent neural networks (RNNs) [30], [31] through memory cells and gating structures. This algorithm offers distinct advantages in handling time-series data, as it dynamically controls the flow of information through the coordinated interaction of the input, forget, and output gates.

The structure of the LSTM neural network is depicted in Fig. 1(a), while its unfolded schematic representation is shown in Fig. 1(b).

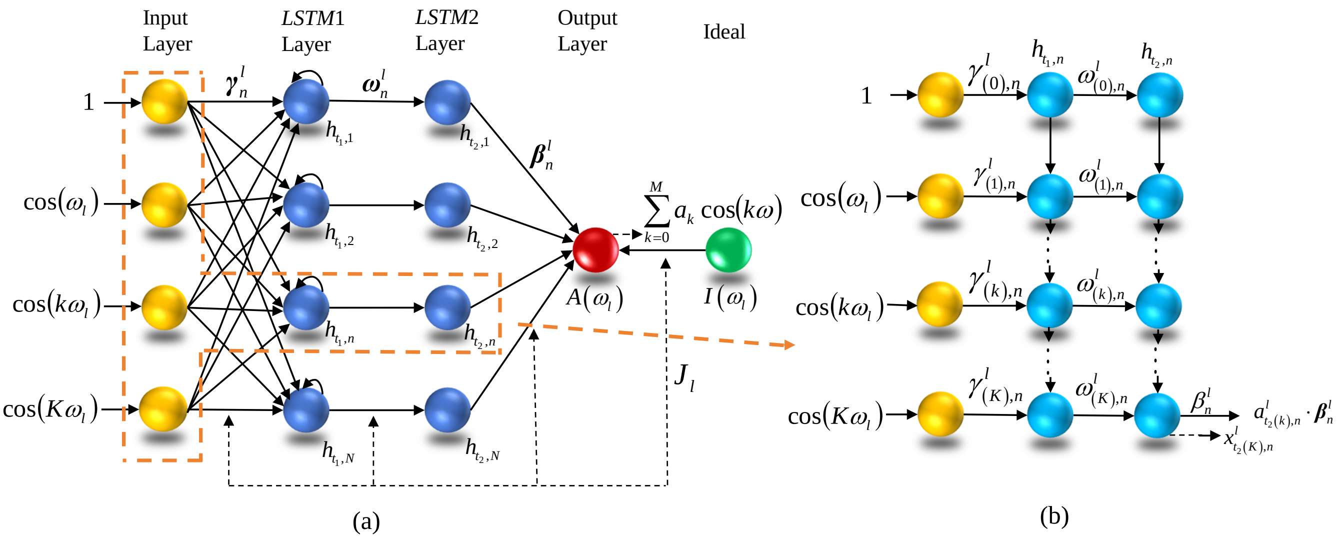

Fig. 1. Structure of the LSTM neural network.

The neural network takes input in the form of a vector, defined as:

$$\begin{aligned}\mathbf{C} &= [1,\cos(\omega_l),\ldots,\cos(k\omega_l),\ldots,\cos(K\omega_l)]^T \\ &= [x_{(0)},x_{(1)},\ldots,x_{(k)},\ldots,x_{(K)}]^T\end{aligned} \tag{8}$$

In the LSTM model, the input and output of the *LSTM*1 layer are:

$$x^l_{t1(k),n} = x_{(k)}\cdot\boldsymbol{\gamma}^l = \cos(k\omega_l)\cdot\boldsymbol{\gamma}^l \tag{9}$$

$$a^l_{t1(k),n} = f(x^l_{t1(k),n}-\lambda^l) \tag{10}$$

$$\boldsymbol{\gamma}^l = [\boldsymbol{\gamma}^l_1,\boldsymbol{\gamma}^l_2,\ldots,\boldsymbol{\gamma}^l_n,\ldots,\boldsymbol{\gamma}^l_N]^T \tag{11}$$

$$\boldsymbol{\gamma}^l_n = [\gamma^l_{(0),n},\gamma^l_{(1),n},\ldots,\gamma^l_{(k),n},\ldots,\gamma^l_{(K),n}]^T \tag{12}$$

Here, $\boldsymbol{\gamma}^l$ denotes the weight matrix that links the input to the *LSTM*1 layer, $\gamma^l_{(k),n}$ represents the weight of the k-th time step. This weight corresponds to the connection between the k-th unit in the initial processing stage and the n-th LSTM unit in the *LSTM*1 layer. λ^l represents the threshold of the *LSTM*1 layer, and $f(\cdot)$ represents a linear activation function.

The input and output of the *LSTM*2 layer are:

$$x^l_{t2(k),n} = a^l_{t1(k),n}\cdot\boldsymbol{\omega}^l \tag{13}$$

$$a^l_{t2(k),n} = f(x^l_{t2(k),n}-\alpha^l) \tag{14}$$

$$\boldsymbol{\omega}^l = [\boldsymbol{\omega}^l_1,\boldsymbol{\omega}^l_2,\ldots,\boldsymbol{\omega}^l_n,\ldots,\boldsymbol{\omega}^l_N]^T \tag{15}$$

$$\boldsymbol{\omega}^l_n = [\omega^l_{(0),n},\omega^l_{(1),n},\ldots,\omega^l_{(k),n},\ldots,\omega^l_{(K),n}]^T \tag{16}$$

Here, $\boldsymbol{\omega}^l_n$ represents the weight matrix that connects the *LSTM*1 layer to the *LSTM*2 layer, $\omega^l_{(k),n}$ denotes the weight for the connection between the n-th LSTM unit's k-th time step in *LSTM*1 and its corresponding one in *LSTM*2, and α^l represents the threshold of the *LSTM*2 layer.

The input and output of the output layer are:

$$x^l_o = a^l_{t2(k),n}\cdot\boldsymbol{\beta}^l \tag{17}$$

$$A(\omega_l) = f(x^l_o-\xi^l) \tag{18}$$

$$\boldsymbol{\beta}^l = [\beta^l_1,\beta^l_2,\ldots,\beta^l_n,\ldots,\beta^l_N]^T \tag{19}$$

Here, $\boldsymbol{\beta}^l$ represents the weight matrix connecting the *LSTM*2 layer to the output layer, β^l_n represents the weight from the n-th LSTM unit of the *LSTM*2 layer to the output layer, and ξ^l represents the threshold of output layer. The amplitude error serves as a performance metric, and the desired amplitude response, $I(\omega_l)$, is defined as:

$$I(\omega_l) = \begin{cases}1, & \omega_l \text{ in the passband} \\ 0, & \omega_l \text{ in the stopband}\end{cases} \quad l = 1,2,\ldots,R \tag{20}$$

Then, the parameter J_l is used to update the weights of the LSTM neural network:

$$J_l = \frac{1}{2}e^2(\omega) \tag{21}$$

To reduce the amplitude error, the weights and threshold must be updated as follows:

$$\begin{cases}\Delta\gamma^l_{(k),n}+\gamma^l_{(k),n}\rightarrow\gamma^l_{(k),n} \\ \Delta\omega^l_{(k),n}+\omega^l_{(k),n}\rightarrow\omega^l_{(k),n} \\ \Delta\beta^l_n+\beta^l_n\rightarrow\beta^l_n \\ \Delta\lambda^l+\lambda^l\rightarrow\lambda^l \\ \Delta\alpha^l+\alpha^l\rightarrow\alpha^l \\ \Delta\xi^l+\xi^l\rightarrow\xi^l\end{cases} \tag{22}$$

with

$$\begin{cases}\Delta\gamma^l_{(k),n} = -\eta\dfrac{\partial J_l}{\partial a^l_{t_1(k),n}}\dfrac{\partial a^l_{t_1(k),n}}{\partial x^l_{t_1(k),n}}\dfrac{\partial x^l_{t_1(k),n}}{\partial\gamma^l_{(k),n}} \\ \Delta\omega^l_{(k),n} = -\eta\dfrac{\partial J_l}{\partial a^l_{t_2(k),n}}\dfrac{\partial a^l_{t_2(k),n}}{\partial x^l_{t_2(k),n}}\dfrac{\partial x^l_{t_2(k),n}}{\partial\omega^l_{(k),n}} \\ \Delta\beta^l_n = -\eta\dfrac{\partial J_l}{\partial A(\omega_l)}\dfrac{\partial A(\omega_l)}{\partial x^l_o}\dfrac{\partial x^l_o}{\partial\beta^l_n} \\ \Delta\lambda^l = -\eta\dfrac{\partial J_l}{\partial a^l_{t_1(k),n}}\dfrac{\partial a^l_{t_1(k),n}}{\partial\lambda^l} \\ \Delta\alpha^l = -\eta\dfrac{\partial J_l}{\partial a^l_{t_2(k),n}}\dfrac{\partial a^l_{t_2(k),n}}{\partial\alpha^l} \\ \Delta\xi^l = -\eta\dfrac{\partial J_l}{\partial A(\omega_l)}\dfrac{\partial A(\omega_l)}{\partial\xi^l}\end{cases} \tag{23}$$

In this context, η represents the learning rate. Once the LSTM model completes the maximum number of iterations, the coefficients of the digital FIR filter are obtained.

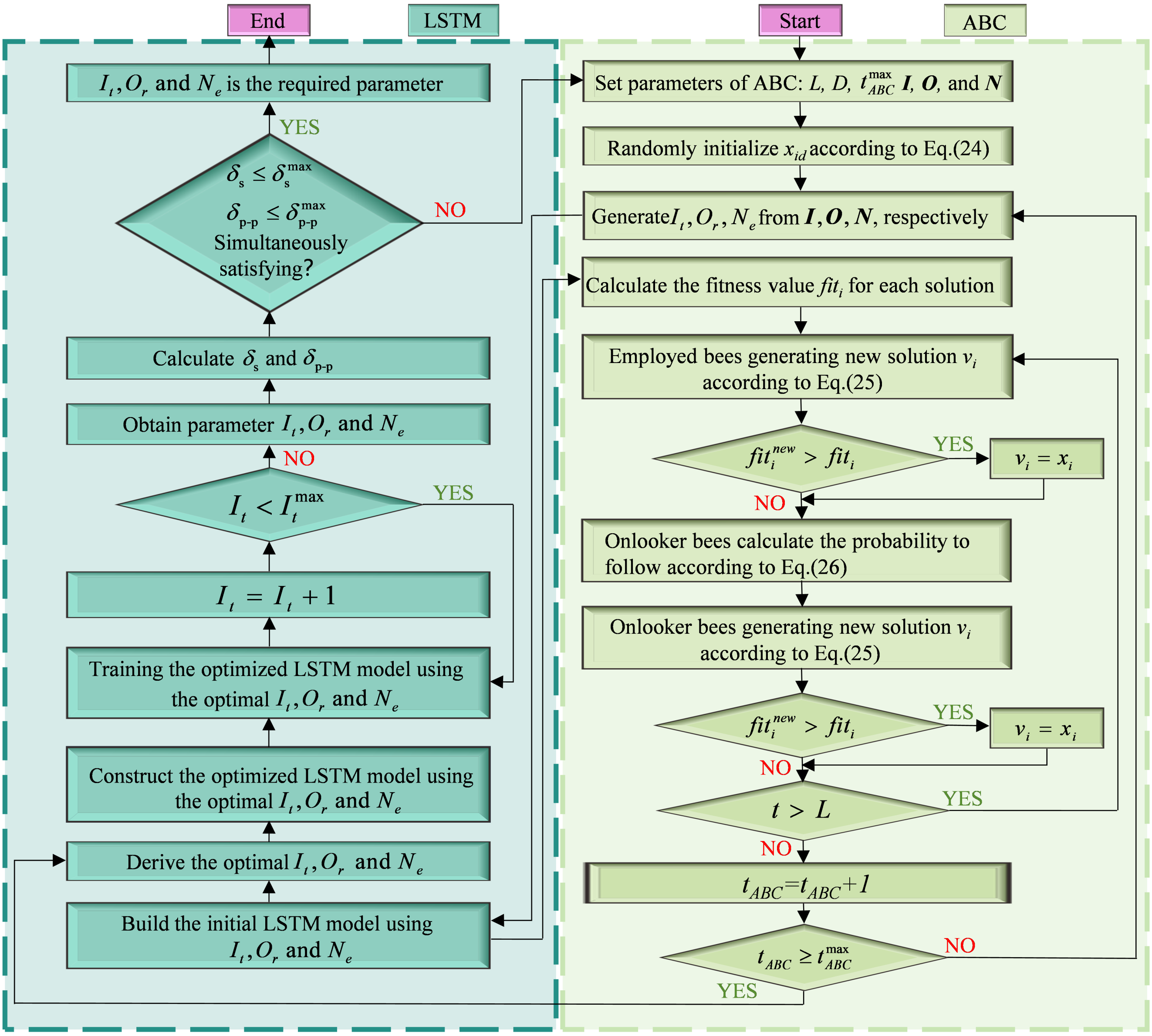
End
LSTM
Start
ABC
I_t, O_r and N_e is the required parameter
YES
$\delta_s \le \delta_s^{max}$
$\delta_{p\text{-}p} \le \delta_{p\text{-}p}^{max}$
Simultaneously satisfying?
NO
Calculate δ_s and $\delta_{p\text{-}p}$
Obtain parameter I_t, O_r and N_e
NO
$I_t < I_t^{max}$
YES
$I_t = I_t + 1$
Training the optimized LSTM model using the optimal I_t, O_r and N_e
Construct the optimized LSTM model using the optimal I_t, O_r and N_e
Derive the optimal I_t, O_r and N_e
Build the initial LSTM model using I_t, O_r and N_e
Set parameters of ABC: L, D, t_{ABC}^{max} $\boldsymbol{I}$, $\boldsymbol{O}$, and $\boldsymbol{N}$
Randomly initialize x_{id} according to Eq.(24)
Generate I_t, O_r, N_e from $\boldsymbol{I}, \boldsymbol{O}, \boldsymbol{N}$, respectively
Calculate the fitness value fit_i for each solution
Employed bees generating new solution v_i according to Eq.(25)
$fit_i^{new} > fit_i$
YES
$v_i = x_i$
NO
Onlooker bees calculate the probability to follow according to Eq.(26)
Onlooker bees generating new solution v_i according to Eq.(25)
$fit_i^{new} > fit_i$
YES
$v_i = x_i$
NO
$t > L$
YES
NO
$t_{ABC}=t_{ABC}+1$
$t_{ABC} \ge t_{ABC}^{max}$
NO
YES

Fig. 2. Parameters optimization flowchart by ABC-LSTM.

B. ABC-LSTM

The artificial bee colony (ABC) algorithm is a metaheuristic optimization method, inspired by the foraging behavior of bees in nature. The algorithm classifies bees into three roles: employed, onlooker, and scout bees. Employed bees search for new solutions near the current best one, while onlooker bees base their search on the locations found by employed bees. Scout bees, on the other hand, randomly select positions within the solution space and generate new solutions under predefined conditions. This mechanism improves the algorithm's capacity to avoid local optima and better explore the global search space.

Randomly initialize i solutions x_i using Eq. (24)

$$x_{id} = L_d + \text{rand}(0,1)(U_d - L_d) \tag{24}$$

Here, L_d and U_d represent the minimum and maximum limits of the search space for each dimension index d, where d ranges over the discrete set $\{1, 2, \ldots, D\}$.

$$v_{id} = x_{id} + \phi_{id}(x_{id} - x_{jd}) \tag{25}$$

Let D be a randomly selected integer from the interval $[1, D]$, denoting the dimension chosen by the employed bee for exploration. The index j corresponds to a honey source selected from NP available sources, ensuring $j \neq i$. The parameter ϕ_{id} is a uniformly distributed random number within the range $[-1, 1]$, governing the magnitude of perturbation.

$$p_i = \frac{fit_i}{\sum_{i=1}^{NP} fit_n} \tag{26}$$

The onlooker bees employ the roulette-wheel selection mechanism to choose among the employed bees. Specifically, a uniformly distributed random number r is generated within the interval $[0, 1]$. If the probability p_i associated with food source i exceeds r, the onlooker bee generates a new food source in the vicinity of food source i based on Eq. (25). The selection process for retaining the food source follows the same methodology as that used by the employed bees.

Fig. 2 illustrates the flowchart outlining the optimization process for selecting the filter order, determining the number of neurons in the LSTM network's hidden layer, and defining the number of iterations in the LSTM network using the ABC algorithm.

1) Set the parameters of the ABC algorithm, including the dimension D of the parameters to be optimized, the iteration limit L for local search, the upper limit t_{ABC}^{max} for the ABC algorithm's iterations, the iteration count t for the local solution, as well as the ranges of LSTM neural network iterations $\boldsymbol{I}$, the number of hidden-layer neurons $\boldsymbol{N}$, and the filter orders $\boldsymbol{O}$.
2) Randomly initialize the food source x_{id} according to Eq.(24).
3) Generate the filter order O_r, the number of LSTM neural network hidden-layer neurons N_e, and iterations I_t as outlined in step 1.
4) Construct the initial LSTM model leveraging the parameters generated in step 3.
5) Compute the fitness value fit_i for each solution.
6) Employed bees generate a new solution v_i using Eq.(25). If $fit_i^{new} > fit_i$, replace x_i with v_i; otherwise, retain x_i.
7) Onlooker bees select a solution to follow based on the probability calculated by Eq.(26).
8) Onlooker bees generate a new solution v_i using Eq.(25). If $fit_i^{new} > fit_i$, replace x_i with v_i; otherwise, retain x_i.
9) Determine whether the current local iteration count t has reached the threshold L. If $t > L$, return to step 6.
10) Increase the iteration count of the ABC algorithm. If $t_{ABC} > t_{ABC}^{max}$, export the optimal parameters I_t, O_r, and N_e; otherwise, return to step 3.
11) Construct and train the optimized LSTM model using the optimal parameters. Increase the iteration count I. If $I_t > I_t^{max}$, train the optimized LSTM model using the optimal parameters; otherwise, export the optimal parameter combination.
12) If all conditions are met $\delta_s \leq \delta_s^{max}$ and $\delta_{p-p} \leq \delta_{p-p}^{max}$ simultaneously, the parameters I_t, O_r, and N_e are the optimal parameters; otherwise, return to step 1.

Algorithm 1 Pseudo-code of ABC-LSTM.

```
Set parameters of ABC: $D$, $L$, $t$, $I$, $O$ and $N$
Randomly initialize $x_{id}$ according to Eq. (1)
Generate $I_t$, $O_r$ and $N_e$ from $\boldsymbol{I}$, $\boldsymbol{O}$, and $\boldsymbol{N}$, respectively
Calculate the fitness $fit_i$ for each solution
Employed bees generating new solution $v_i$ according to Eq. (2)
if $fit_i^{new} > fit_i$ then
    $v_i = x_i$
else
    continue with $x_i$
end if
Onlooker bees calculate the probability to follow according to Eq. (3)
Onlooker bees generating new solution $v_i$ according to Eq. (2)
if $fit_i^{new} < fit_i$ then
    $v_i = x_i$
else
    if $t > L$ then
        Employed bees generating new solution $v_i$ according to Eq. (2)
    else
        $t_{ABC} = t_{ABC} + 1$
    end if
end if
if $t_{ABC} \geq t_{ABC}^{max}$ then
    Derive the optimal It, Or and Ne
else
    Generate $I_t$, $O_r$, $N_e$ from $I$, $O$, $N$, respective
end if
Training the optimized LSTM model using the optimal $I_t$, $O_r$ and $N_e$
$I_t = I_t + 1$
if $I_t < I_t^{max}$ then
    Training the optimized LSTM model using the optimal $I_t$, $O_r$ and $N_e$
else
    Obtain parameter $I_t$, $O_r$, and $N_e$
end if
if $\delta_{\mathrm{s}} \leq \delta_{\mathrm{s}}^{\max}, \delta_{\mathrm{p-p}} \leq \delta_{\mathrm{p-p}}^{\max}$ then
    $I_t$, $O_r$, and $N_e$ is the required parameter
else
    Set parameters of ABC: $D$, $L$, $t$, $I$, $O$ and $N$
end if
```

IV. Simulation results and analysis

The effectiveness of our ABC-LSTM-based digital FIR filter design has been thoroughly evaluated and examined using multiple unique design examples. In these instances, the order M of the digital FIR filter was set to 40, the aggregate quantity of sampled frequency values L was set to 2837, and $\eta = 1$.

A. Assessment of effectiveness

Example 1: Design of low-pass finite impulse response filters

In the first design example, the passband and stopband threshold frequencies were set to 0.45π and 0.55π, respectively.

Fig. 3 displays the frequency responses of low-pass FIR filters designed through various methods. Table I presents the key performance metrics for each approach, including passband fluctuation (δ_{p-p}), maximal stopband suppression ($\delta_{s_{max}}$), and minimal stopband suppression ($\delta_{s_{min}}$). Analysis of the results shows that the LSTM neural network achieves a δ_{p-p} of 0.1065 dB, while the BPNN and TNN methods yield 1.5928 dB and 1.2441 dB, respectively. The rectangular window approach demonstrates a value of 1.2104 dB. In comparison, our ABC-LSTM-based design achieves a lower δ_{p-p} of 0.0674 dB, outperforming these four methods.

Moreover, the maximal stopband suppression ($\delta_{s_{\max}}$) achieved by the ABC-LSTM, LSTM neural network, BPNN, TNN, and rectangular window methods are -43.1007 dB, -41.3632 dB, -28.6924 dB, -26.3144 dB, and -26.5044 dB, correspondingly. Similarly, the minimal

TABLE I
SUMMARY OF PASSBAND FLUCTUATION AMPLITUDE, AND THE RANGE OF STOPBAND SUPPRESSION

Design Example	Method	δ_{P-P}	$\delta_{s\ \max}$	$\delta_{s\ \min}$
Low-pass digital FIR filter	ABC-LSTM	**0.0674 dB**	**-43.1007 dB**	**-117.0269 dB**
	LSTM neural network [23]	0.1065 dB	-41.3632 dB	-106.5557 dB
	BPNN [22]	1.5928 dB	-28.6924 dB	-91.2463 dB
	TNN [21]	1.2441 dB	-26.3144 dB	-88.5629 dB
	Rectangular window [12]	1.2104 dB	-26.5044 dB	-88.6183 dB
High-pass digital FIR filter	ABC-LSTM	**0.4468 dB**	**-41.6526 dB**	**-123.6674 dB**
	LSTM neural network [23]	0.7133 dB	-41.1869 dB	-114.2902 dB
	BPNN [22]	1.6374 dB	-26.2171 dB	-93.6799 dB
	TNN [21]	1.0192 dB	-29.3275 dB	-106.4870 dB
	Rectangular window [12]	1.2104 dB	-29.7825 dB	-88.6179 dB
Band-pass digital FIR filter	ABC-LSTM	**0.3795 dB**	**-41.3415 dB**	**-122.4697 dB**
	LSTM neural network [23]	0.7990 dB	-30.0651 dB	-118.6382 dB
	BPNN [22]	1.6046 dB	-23.3088 dB	-103.9448 dB
	TNN [21]	1.4286 dB	-21.0974 dB	-92.0916 dB
	Rectangular window [12]	1.3244 dB	-20.9115 dB	-82.2671 dB
Band-stop digital FIR filter	ABC-LSTM	**0.7990 dB**	**-38.5022 dB**	**-114.0830 dB**
	LSTM neural network [23]	0.7998 dB	-37.8101 dB	-109.7864 dB
	BPNN [22]	0.7996 dB	-18.9314 dB	-93.8306 dB
	TNN [21]	0.7996 dB	-21.4342 dB	-80.6333 dB
	Rectangular window [12]	0.7996 dB	-20.0883 dB	-81.9966 dB

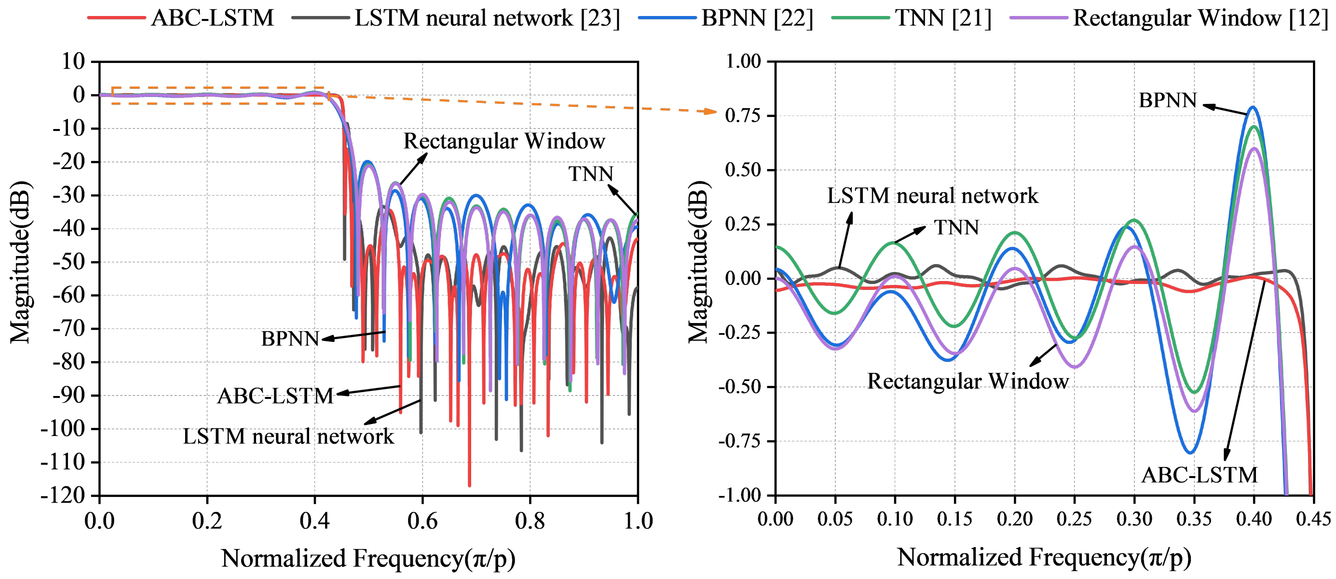

Fig. 3. Frequency response comparison of low-pass digital FIR filters designed using different methods.

stopband suppression ($\delta_{s_{\min}}$) values for these methods are -117.0269 dB, -106.5557 dB, -91.2463 dB, -88.5629 dB, and -88.6183 dB, respectively. The results clearly show that the ABC-LSTM-based optimization design outperforms the other four methods, achieving notably lower maximal and minimal stopband suppression levels.

Example 2: Design of high-pass finite impulse response filters

In design example 2, the cutoff frequencies for the passband and stopband were set to 0.55π and 0.45π, correspondingly.

Fig. 4 presents the frequency responses of high-pass FIR filters designed using different techniques. Table I summarizes the key performance metrics, including passband fluctuation (δ_{p-p}), maximal stopband suppression ($\delta_{s_{max}}$), and minimal stopband suppression ($\delta_{s_{min}}$) for each approach. Analysis of the results shows that the LSTM neural network achieves a δ_{p-p} of 0.7133 dB, while the BPNN and TNN methods yield 1.6374 dB and 1.0192 dB, respectively. The rectangular window approach demonstrates a value of 1.2104 dB. In comparison, our ABC-LSTM-based design achieves a lower δ_{p-p} of 0.4468 dB, outperforming these four methods.

Additionally, the maximal stopband suppression ($\delta_{s_{max}}$) values achieved by the ABC-LSTM, LSTM neural network, BPNN, TNN, and rectangular window methods are -41.6526 dB, -41.1869 dB, -26.2171 dB, -29.3275 dB, and -29.7825 dB, respectively. The corresponding minimal stopband

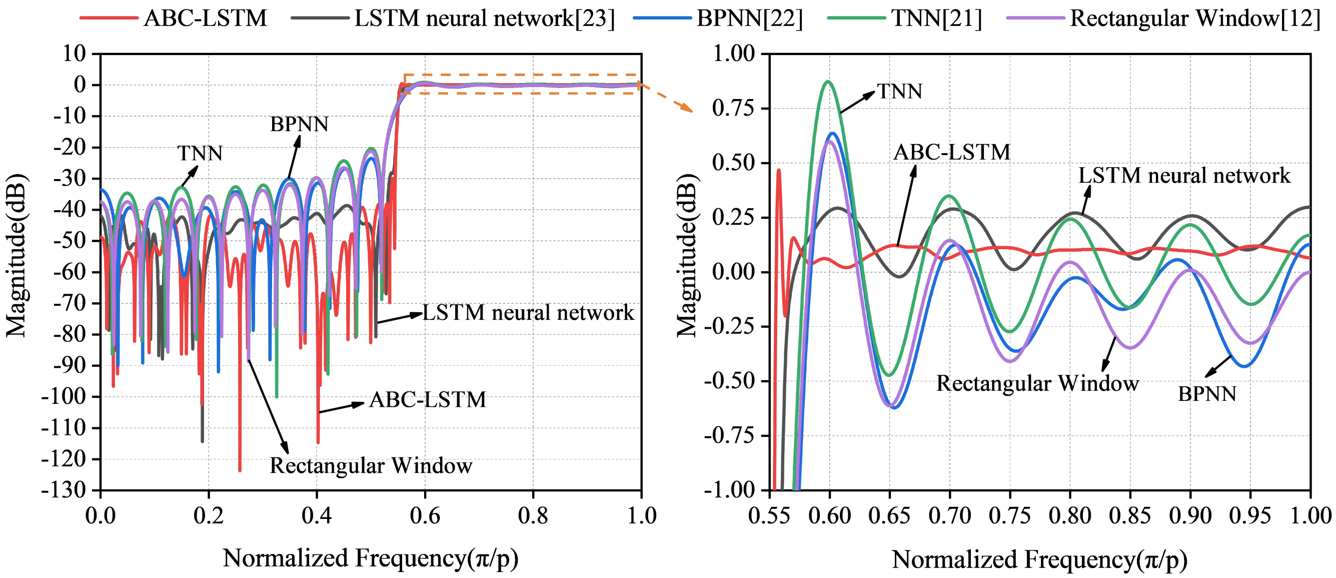

Fig. 4. Frequency response comparison of high-pass digital FIR filters designed using different methods.

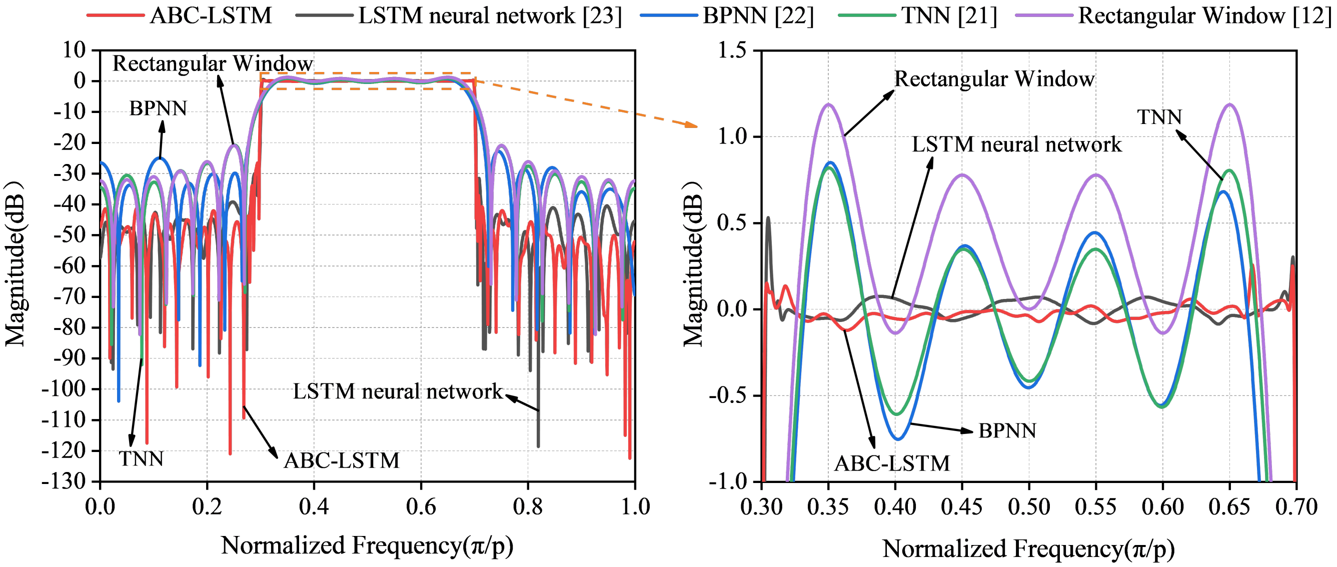

Fig. 5. Frequency response comparison of band-pass digital FIR filters designed using different methods.

suppression ($\delta_{s_{min}}$) values are -123.6674 dB, -114.2902 dB, -93.6799 dB, -106.4807 dB, and -88.6179 dB. These results clearly indicate that the ABC-LSTM-based optimization outperforms the other methods, achieving significantly lower maximal and minimal stopband suppression levels.

Example 3: Design of band-pass finite impulse response filters

Regarding the third design case, the threshold frequencies for the passband are arranged at 0.3π and 0.7π, and the stopband threshold frequencies are determined to be 0.25π and 0.75π.

Fig. 5 illustrates the frequency responses of band-pass FIR filters designed using various techniques. Table I summarizes the key performance metrics, including passband fluctuation (δ_{p-p}), maximal stopband suppression ($\delta_{s_{max}}$), and minimal stopband suppression ($\delta_{s_{min}}$) for each approach. Analysis of the results shows that the LSTM neural network achieves a δ_{p-p} of 0.7990 dB, while the BPNN and TNN methods yield 1.6046 dB and 1.4286 dB, respectively. The rectangular window approach demonstrates a value of 1.3244 dB. In comparison, our ABC-LSTM-based design achieves a lower δ_{p-p} of 0.3795 dB, outperforming these four methods.

The maximal stopband suppression ($\delta_{s_{max}}$) achieved by methods based on ABC-LSTM, LSTM neural network, BPNN, TNN, and the rectangular window method are -41.3415 dB, -30.0651 dB, -23.3088 dB, -21.0974 dB, and -20.9115 dB, respectively. Meanwhile, the minimal stopband suppression ($\delta_{s_{min}}$) values are -122.4697 dB, -118.6382 dB, -103.9448 dB, -92.0916 dB, and -82.2671 dB, respectively. In comparison to the other four methods, our proposed ABC-LSTM-based optimization design achieves lower values for both maximal and minimal stopband suppression.

Example 4: Design of band-stop finite impulse response filters

In the fourth design example, the threshold frequencies of the passband are set to 0.25π and 0.75π, while the stopband is defined by frequencies of 0.3π and 0.7π.

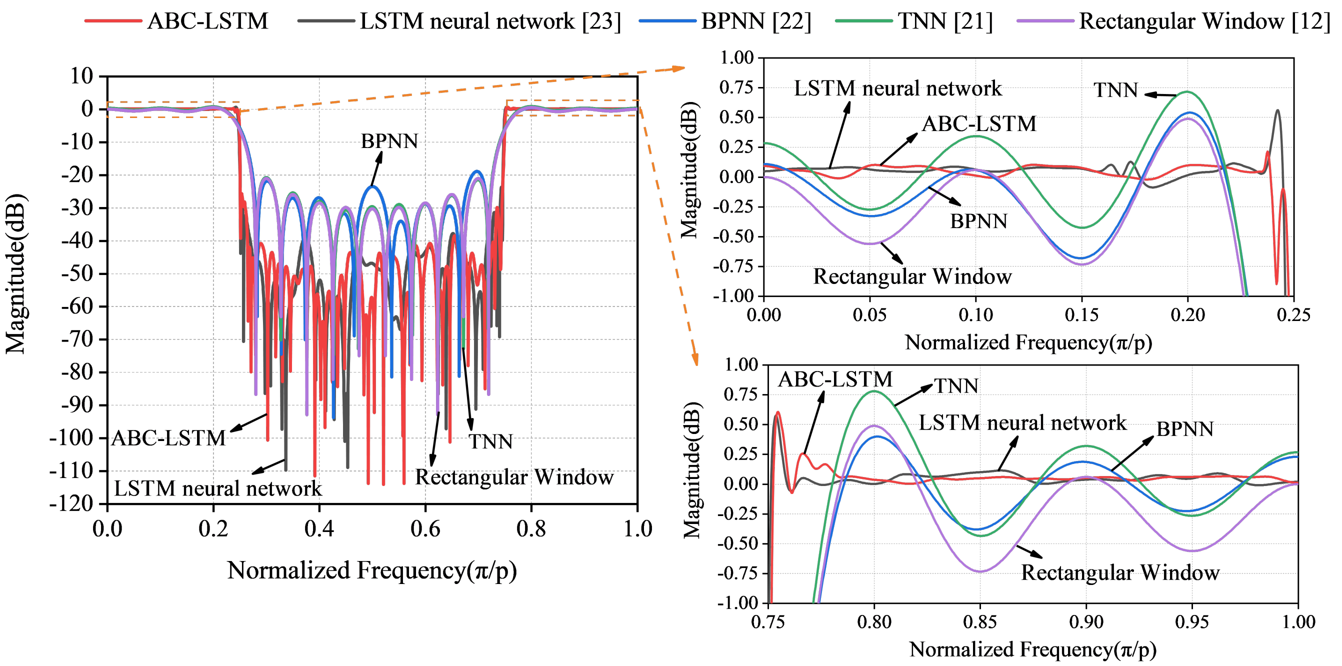

Fig. 6. Frequency response comparison of band-stop digital FIR filters designed using different methods.

Figure 6 displays the frequency responses of band-stop FIR filters designed using various methods. Table I summarizes the key performance metrics, including passband fluctuation (δ_{p-p}), maximal stopband suppression ($\delta_{s_{max}}$), and minimal stopband suppression ($\delta_{s_{min}}$) for each approach. Analysis of the results shows that the LSTM neural network achieves a δ_{p-p} of 0.7998 dB, while the BPNN and TNN methods both yield 0.7996 dB. The rectangular window approach also demonstrates a similar value of 0.7996 dB. In comparison, our ABC-LSTM-based design achieves a lower δ_{p-p} of 0.7990 dB, outperforming these four methods.

The maximal stopband suppression values $\delta_{s_{max}}$ for the ABC-LSTM, LSTM neural network, BPNN, TNN, and rectangular window methods are -38.5022 dB, -37.8101 dB, -18.9314 dB, -21.4342 dB, and -20.0883 dB, respectively. In contrast, the minimal stopband suppression values $\delta_{s_{min}}$ for these methods are -114.0830 dB, -109.7864 dB, -93.8306 dB, -80.6333 dB, and -81.9966 dB, respectively. The ABC-LSTM method outperforms the other four approaches by achieving lower maximal and minimal stopband suppression, demonstrating its superior effectiveness.

The analysis of Examples 1 to 4 shows that the ABC-LSTM implementation outperforms other methods, including LSTM neural network, BPNN, TNN, and rectangular window techniques, in terms of improved performance metrics such as reduced passband fluctuation and better stopband suppression.

Additionally, the performance of these four digital FIR filter designs can be ranked quantitatively, from highest to lowest performance:

$$\text{ABC-LSTM} > [23] > [22] > [21] > [12] \tag{27}$$

B. Mean square error analysis and complexity analysis

The operational frequency range for digital FIR filters, including low-pass, high-pass, and band-pass types, is defined as $\Omega_p = [\omega_{p1}, \omega_{p2}]$. Passband accuracy is evaluated using mean square error analysis:

$$\mathrm{MSE_p} = \frac{\sum_{l=\frac{\omega_{p1}}{\pi}L}^{\frac{\omega_{p2}}{\pi}L} (I(\omega_l) - A(\omega_l))^2}{\frac{\omega_{p2} - \omega_{p1}}{\pi} L} \tag{28}$$

For the band-stop digital FIR filter implementation, the operational frequency range is defined as $\Omega_p = [\omega_{p1}, \omega_{p2}] \cup [\omega_{p3}, \omega_{p4}]$. The passband accuracy is then determined using mean square error analysis:

$$\mathrm{MSE_p} = \frac{\sum_{l=\frac{\omega_{p1}}{\pi}L}^{\frac{\omega_{p2}}{\pi}L} (I(\omega_l) - A(\omega_l))^2 + \sum_{l=\frac{\omega_{p3}}{\pi}L}^{\frac{\omega_{p4}}{\pi}L} (I(\omega_l) - A(\omega_l))^2}{\frac{(\omega_{p2} - \omega_{p1}) + (\omega_{p4} - \omega_{p3})}{\pi} L} \tag{29}$$

The stopband region for the low-pass and band-pass digital FIR filters is defined as $\Omega_s = [\omega_{s1}, \omega_{s2}]$. The mean square error of the stopband is expressed as:

$$\mathrm{MSE_s} = \frac{\sum_{l=\frac{\omega_{s1}}{\pi}L}^{\frac{\omega_{s2}}{\pi}L} (I(\omega_l) - A(\omega_l))^2}{\frac{\omega_{s2} - \omega_{s1}}{\pi} L} \tag{30}$$

The stopband bandwidth of a stopband digital FIR filter can be set to $\Omega_s = [\omega_{s1}, \omega_{s2}] \cup [\omega_{s3}, \omega_{s4}]$. The mean square error of the stopband is expressed as:

$$\mathrm{MSE_s} = \frac{\sum_{l=\frac{\omega_{s1}}{\pi}L}^{\frac{\omega_{s2}}{\pi}L} (I(\omega_l) - A(\omega_l))^2 + \sum_{l=\frac{\omega_{s3}}{\pi}L}^{\frac{\omega_{s4}}{\pi}L} (I(\omega_l) - A(\omega_l))^2}{\frac{(\omega_{s2} - \omega_{s1}) + (\omega_{s4} - \omega_{s3})}{\pi} L} \tag{31}$$

In terms of computational efficiency, the implementation of ABC-LSTM-optimized digital FIR filters (low-pass,

TABLE II
COMPARATIVE ANALYSIS OF MEAN SQUARE ERROR AND ALOGORITHMIC

Design Example	Method	MSE$_P$	MSE$_S$	CPU running time	Order	Neurons	Ierations
Low-pass digital FIR filter	ABC-LSTM	**0.0002**	**0.8105×10^{-5}**	71.2888s	**32**	**69**	**77**
	LSTM neural network [23]	0.00023	1.1159×10^{-5}	185.7794s	62	128	100
	BPNN [22]	0.0078	0.0002	90.5965s	-	-	-
	TNN [21]	0.0060	0.0002	37.7183s	-	-	-
	Rectangular window [12]	0.9418	0.0001	**0.0013s**	-	-	-
High-pass digital FIR filter	ABC-LSTM	**0.0015**	**1.1003×10^{-5}**	118.6276s	**54**	**76**	**81**
	LSTM neural network [23]	0.0019	3.1413×10^{-5}	179.9818s	62	128	100
	BPNN [22]	0.0060	0.0007	95.9772s	-	-	-
	TNN [21]	0.0054	0.0003	19.8563s	-	-	-
	Rectangular window [12]	0.9459	0.0001	**0.0018s**	-	-	-
Band-pass digital FIR filter	ABC-LSTM	**0.0005**	**1.7226×10^{-5}**	106.5470s	**52**	**82**	**88**
	LSTM neural network [23]	0.0013	2.0276×10^{-5}	110.3697s	62	128	100
	BPNN [22]	0.0193	0.0007	79.2226s	-	-	-
	TNN [21]	0.0138	0.0008	34.3453s	-	-	-
	Rectangular window [12]	1.1049	0.0004	**0.0017s**	-	-	-
Band-stop digital FIR filter	ABC-LSTM	**0.0013**	**2.0837×10^{-5}**	90.5903s	**40**	**27**	**92**
	LSTM neural network [23]	0.0025	2.1359×10^{-5}	144.9199s	62	128	100
	BPNN [22]	0.0098	0.0015	93.8691s	-	-	-
	TNN [21]	0.0101	0.0011	14.4203s	-	-	-
	Rectangular window [12]	1.0400	0.0073	**0.0019s**	-	-	-

high-pass, band-pass, and band-stop) on systems with a 2.8 GHz CPU results in processing times of 71.29 s, 118.63 s, 106.55 s, and 90.59 s, respectively. The optimization design using the LSTM neural network takes 185.7794 s, 179.9818 s, 110.3697 s, and 144.9199 s, respectively. The optimization design using BPNN takes 90.5965 s, 95.9772 s, 79.2226 s, and 93.8691 s, respectively. In contrast, the optimization design using TNN takes 37.7183 s, 19.8563 s, 34.3453 s, and 14.4203 s, respectively. Finally, the optimization design using the rectangular window method takes 0.0013 s, 0.0018 s, 0.0017 s, and 0.0019 s, respectively.

Table II provides a summary of the passband mean square error (MSE$_p$), stopband mean square error (MSE$_s$), and computational complexity (CPU execution time), comparing designs optimized using ABC-LSTM, LSTM neural network, BPNN, TNN, and the rectangular window method. Additionally, we can analyze the filter order, the number of neurons in the hidden layers of the LSTM neural network, and the number of iterations. In the traditional LSTM neural network design, the parameters for low-pass, high-pass, band-pass, and band-stop filters are all the same and are 62, 128, and 100, respectively. In contrast, our proposed ABC-LSTM design achieves parameters of 32, 69, and 77 for the low-pass filter, representing reductions of 48%, 46%, and 23%compared to the traditional LSTM neural network design. For the high-pass filter, the ABC-LSTM design yields parameters of 54, 76, and 81, corresponding to reductions of 13%, 40%, and 19%compared to the traditional LSTM neural network design. The band-pass filter parameters in the ABC-LSTM design are 52, 82, and 88, showing reductions of 17%, 36%, and 12%compared to the traditional LSTM neural network design. Finally, the band-stop filter parameters in the ABC-LSTM design are 40, 27, and 92, indicating reductions of 35%, 78%, and 8%compared to the traditional LSTM neural network design.

These results demonstrate that the proposed ABC-LSTM-based design achieves lower passband and stopband MSE compared to the other four methods. It not only reduces the relevant key parameters, including the filter order, the number of the LSTM neural network hidden layer neurons and the iterations, but also minimizes the CPU running time.

V. Conclusion

This paper presents an algorithm that combines the artificial bee colony (ABC) algorithm with long short-term memory (LSTM) networks for the design of digital finite impulse response (FIR) filters. The objective is to improve filter performance by minimizing the amplitude error between the actual and desired frequency responses. First, a digital FIR filter model is constructed to obtain the actual amplitude response and the desired amplitude response, and then the amplitude error between them is calculated. Following this, we proposed a long short-term memory (LSTM) neural network to minimize the amplitude error. To reduce the computational complexity associated with the traditional LSTM neural network design, the ABC algorithm is employed to optimize the filter order, the number of LSTM neural network hidden layer neurons, and the number of LSTM neural network iterations, ensuring that the given upper limits of the fluctuation in the passband and the suppression in the stopband can be satisfied simultaneously. Then, these optimized parameters are applied to the LSTM model to generate the optimal digital finite impulse response (FIR) filter coefficients.

The simulation results demonstrate that the proposed ABC-LSTM-based filter design exhibits superior performance across all four types of filters. In terms of passband characteristics, the passband fluctuations are measured to be 0.0674 dB, 0.4468 dB, 0.3795 dB, and 0.7990 dB, respectively. Regarding the stopband performance, the corresponding ranges of stopband suppression levels are [-43.1007 dB, -117.0269 dB], [-41.6526 dB, -123.6674 dB], [-41.3415 dB, -122.4697 dB], and [-38.5022 dB, -114.0830 dB], respectively. Regarding the error metrics, the passband mean square errors are quantified as 0.00020, 0.0015, 0.0005, and 0.0013, while the stopband mean square error values are measured to be 0.8105×10^{-5}, 1.1003×10^{-5}, 1.7226×10^{-5}, and 2.0837×10^{-5}, respectively.

The ABC-LSTM-based approach outperforms traditional digital finite impulse response (FIR) filter design methods, including LSTM, backpropagation neural networks (BPNN), traditional neural networks (TNN), and rectangular window techniques, in terms of performance metrics. These performance improvements are achieved while maintaining a low computational complexity by not only reducing the correlation coefficients, including the filter order, the number of LSTM neural network hidden layer neurons and the iterations, but also by satisfying the given upper bounds on the passband fluctuations and the stopband suppression.

REFERENCES

[1] E. Chitra, et al., “Analysis and Implementation of High Performance Reconfigurable Finite Impulse Response Filter Using Distributed Arithmetic,” *Wireless Personal Communications*, pp. 1–4, 2018.

[2] R. S. Roach, N. N. Singh and T. S. A. Samuel, “Resource minimization and power reduction of ESPFFIR filter using unified adder/subtractor,” *Analog Integrated Circuits and Signal Processing*, vol. 98, no. 1, pp. 169-179, 2019.

[3] Rodjarin, et al., “Algorithmic Scheme-Integrated Bandwidth Compensatory Reconstruction Filter of Digital Signal Processing System,” *Journal of Signal Processing*, vol. 20, no. 3, pp. 91-103, 2016.

[4] M. L., et al., “RDO-WT: optimised Wallace Tree multiplier based FIR filter for signal processing applications,” *International Journal of Electronics*, vol. 109, no. 10, pp. 1759-1780, 2022.

[5] K. TAKATO, et al., “A Serial Processing FIR Inverse Filter Circuit,” *Electronics and Communications in Japan*, vol. 98, no. 12, pp. 37-45, 2015.

[6] K. Andrade, et al., “A Spectrum Analyzer in the 470 to 698 MHz Band Using Software Defined Radio for the Analysis of Digital Terrestrial Television Signals (DTTs),” *Engineering Proceedings*, vol. 47, pp. 3129–3136, 2023.

[7] L. Guanhua, et al., “Design of Experimental Circuit Board of Spectrometer Amplifier in Nuclear Electronics,” *Journal of Physics: Conference Series*, vol. 2440, no. 1, pp. 1–7, 2023.

[8] S. Udhayasuriyan, et al., “A Compact Metamaterial-Based Open-Ended CPW Band-Pass Filter for Wireless Applications,” *Arabian journal for science and engineering*, no. 3, pp. 47, 2022.

[9] W. F., et al., “Research on Kalman Filter for One-dimensional Discrete Data,” *Journal of Physics: Conference Series*, vol. 2005, no. 1, pp. 1–7, 2021.

[10] S. N. Mirebrahimi, et al., “Programmable discrete-time type I and type II FIR filter design on the memristor crossbar structure,” *Analog Integrated Circuits and Signal Processing*, vol. 79, no. 3, pp. 529-541, 2014.

[11] M. Bayasgalan, et al., “Graphical User Interface Design of FIR Filter (),” *Open Access Library Journal*, vol. 5, no. 12, pp. 1–10, 2018.

[12] F. Serbet, et al., “Optimization Approach in Window Function Design for Real-Time Filter Applications,” *Journal of Circuits, Systems & Computers*, vol. 32, no. 13, pp. 2350215-2350226, 2023.

[13] A. A. S. Hannah, et al., “A Design of a low-pass FIR filter using Hamming Window Functions in Matlab,” *Analog Integrated Circuits and Signal Processing*, 2020.

[14] Y. Sarucha, et al., “A Novel Robust Adaptive Control for PMDC Servo Motor Incorporating Recursive Least Square and Particle Swarm Optimization,” *Proceedings of The International MultiConference of Engineers and Computer Scientists 2023*, pp. 118–123, July. 2023.

[15] J. Chen, et al., “Novel Non-iterative Least Square Method with Adjustable Weighting Vector for Filter Design,” *Analog Integrated Circuits and Signal Processing*, vol. 466, no. 1, pp. 012063 (6pp), 2014.

[16] S. Mahata, et al., “Design of fractional-order transitional filters of the Butterworth-Sync-Tuned, Butterworth-Chebyshev, and Chebyshev-Sync-Tuned types: optimization, simulation, and experimental verification,” *Computers and Electrical Engineering*, vol. 116, pp. 109200, 2024.

[17] G. Ming-Xia, et al., “Selecting New Construction Road Sections for Urban Area with Hazard Source Based on Resilience,” *IAENG International Journal of Applied Mathematics*, vol. 54, no. 10, pp. 1234-1245, 2024.

[18] X. H. Le, et al., “Comparison of Deep Learning Techniques for River Streamflow Forecasting,” *IEEE Access*, vol. 9, pp. 71805-71820 ,2021.

[19] H. Zheng, et al., “Long-term urban air quality prediction with hierarchical attention loop network,” *Sustainable Cities and Society*, vol. 118, pp. 106215, 2025.

[20] L. Wang, et al., “Short-Term PV Power Prediction Based on Optimized VMD and LSTM,” *IEEE Access*, vol. 8, pp. 165849-165862, 2020.

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

[22] J. Yang, et al., “Optimal design of digital FIR filters based on back propagation neural network,” *IEICE Electron. Express*, vol. 20, no. 1, pp. 20220491-20220491, 2023.

[23] F. Zhu, et al., “Design of Digital FIR Filter Based on Long Short-Term Memory Neural Network,” *Engineering Letters*, vol. 32, no. 10, pp. 1993-2001, 2024.

[24] A. G. Abro, et al., “Enhanced Global-Best Artificial Bee Colony Optimization Algorithm,” *Computer Modeling and Simulation (EMS), 2012 Sixth UKSim/AMSS European Symposium on*, pp. 1-6, 2012.

[25] M. Liu, et al., “A learning-based artificial bee colony algorithm for operation optimization in gas pipelines,” *Information Sciences*, vol. 690, pp. 121593-121593, 2025.

[26] T. N. Sainath, et al., “Convolutional, Long Short-Term Memory, fully connected Deep Neural Networks,” *ICASSP 2015 - 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4580-4584, 2015.

[27] H. M. Safayet, et al., “Short-Term Photovoltaic Power Forecasting Using an LSTM Neural Network and Synthetic Weather Forecast,” *IEEE ACCESS*, vol. 8, pp. 172524-172533, 2020.

[28] A. R. H. and F.Sara, “Predicting the Price of Crude Oil and its Fluctuations Using Computational Econometrics: Deep Learning, LSTM, and Convolutional Neural Networks,” *Econometric Research in Finance*, vol. 6, no. 2, pp. 119-137, 2021.

[29] H. Feng, et al., “A new wind power forecasting algorithm based on long short-term memory neural network,” *International Transactions on Electrical Energy Systems*, vol. 31, no. 12, pp. 1-11, 2021.

[30] H. Wang, et al., “Dynamic optimization of a pilot-scale entrained-flow gasifier using artificial recurrent neural networks,” *Fuel*, vol. 272, pp.117735, 2020.

[31] S. Hashida, et al., “Classifying Tweets using Convolutional Neural Networks with Multi-Channel Distributed Representation,” *IAENG Internaitonal journal of computer science*, vol. 46, no. 1, pp. 68-75, 2019.