Optimal Design of Digital Analysis Filters Based on PSO-BPNN for Aliasing Errors Cancellation in HFB DAC

Yan Wang, Shengjian Liu, Xing Yang, Weiyuan Zhang, Jiansheng Yang, and Jiyao Yang

***Abstract*—To cancel the aliasing errors introduced by the non-ideal characteristics of analog filters and mixers, this paper proposes a method that combines particle swarm optimization (PSO) with back propagation neural networks (BPNN) for designing digital analysis filters in hybrid filter bank digital-to-analog converter (HFB DAC). A mathematical model for the HFB DAC is initially established to derive both its desired and practical transfer functions, facilitating the calculation of the estimation error between them. Next, the approximation error is derived from both the real and imaginary components of the estimation error. A BPNN method is then proposed to minimize the approximation error. To reduce the computational complexity of the traditional BPNN design, we also proposed a PSO algorithm for optimizing all the sub-filter orders, the number of the BPNN hidden layer neurons and the iterations of the BPNN, enabling that the given upper limit errors (upper limits of distortion and aliasing errors) is met. Finally, these optimized parameters are applied to the BPNN method, thereby deriving the optimal coefficients for digital analysis filters. Additionally, this paper derives the computational complexity of PSO-BPNN. Several design examples indicate that, comparing with the other four designs, our proposed PSO-BPNN design not only achieves better aliasing errors cancellation but also reduces all the sub-filter orders, the number of the BPNN hidden layer neurons and the iterations of the BPNN.**

***Index Terms*—HFB DAC, aliasing errors cancellation, digital analysis filters, PSO-BPNN.**

I. Introduction

As modern communication systems continue to evolve, the signal bandwidth required for systems is steadily increasing [1]. Digital sampling technology is usually used to generate the wideband signal required by modern communication systems [2]. The digital-to-analog converter (DAC) is a key element in digital sampling technology, designed to produce wideband signals [3]. However, the wideband signal's bandwidth is limited by the DAC's sampling rate [4].

The DAC's sampling rate can be increased through fabrication processes [5], [6], [7]. Whereas, it can be known from Nyquist theorem [8], the bandwidth of the wideband signal should not exceed half the sampling rate of the DAC. To avoid the limitations imposed by the Nyquist theorem, time-interleaved DAC (TI-DAC) [9], [10]has been proposed to indirectly enhance the sampling rate of the sub-DAC. In spite of this, the bandwidth of the wideband signal remains limited by the zero-order hold (ZOH) characteristic of the sub-DAC, which cannot exceed the sampling rate of the sub-DAC [11].

To overcome the bandwidth limitations caused by the ZOH characteristic of the sub-DAC, a hybrid filter bank DAC is introduced [12], [13], [14]. Fig.1 demonstrates the architecture of an HFB DAC with M sub-DACs. As demonstrate in Fig.1 and based on the principles of the HFB DAC [15], [16], [17], the HFB DAC output bandwidth is M times the sub-DAC output bandwidth. Nonetheless, several system errors are propose in HFB DAC, such as the aliasing errors introduced by the non-ideal characteristics of the analog filters and mixers. These aliasing errors result in distortion within the wideband signal generated by the HFB DAC [18], [19].

To cancel the aliasing errors in HFB DAC system, X. Yang et al. proposed an optimal design of digital analysis filters based on weighted least squares (WLS) design [20]. Similarly, L. Yin et al. proposed an optimal design of digital analysis filters based on WLS+Optimization design [21]. The aim is to determine the optimal coefficients for digital analysis filters by minimizing the estimation error between the desired and practical transfer functions, thereby achieving the aliasing errors cancellation.

Machine learning constitutes a central area of research, primarily aimed at minimizing the error between the desired and practical values in specific applications [22], [23], [24], [25], [26]. For instance, X. Ye et al. proposed a method for analyzing theoretical line losses in distribution networks using back propagation neural networks (BPNN) [27]. This method aims with the aim of minimizing the error between the desired and practical theoretical line loss of the distribution network. L. Jiang et al. proposed a neural network-based method to suppress vibrations in flexible joint space robots, aim of minimizing the error between the desired and practical trajectories [28]. While these aforementioned papers do not discuss how to use machine learning methods for the design of digital analysis filters, they provide valuable references for

Manuscript received July 25, 2024; revised November 18, 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.

Yan Wang is a postgraduate student of the School of Electrical Engineering, Guizhou University, Guiyang 550025, China (e-mail: 18984429437@163.com).

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

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

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: yaoyj@163.com).

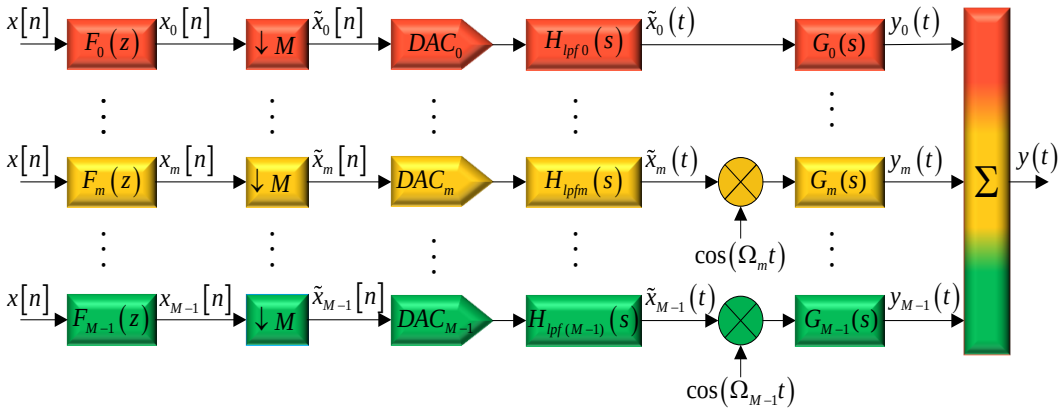


Fig. 1: Architecture of the HFB DAC

research on this topic.

Currently, only a few researchers have investigated the optimal design of digital analysis filters using machine learning methods. Among them, C.-C. Tseng et al. proposed a traditional neural network (TNN) method for the design of digital analysis filters, minimizing the estimation error between the desired and practical amplitude responses [29]. Additionally, J. Yang et al. proposed a BPNN method for the design of digital analysis filters, minimizing the estimation error between the desired and practical amplitude responses [30]. None of the extant papers address how to achieve the aliasing errors cancellation in HFB DAC through the design of digital analysis filters using machine learning methods. However, these papers remain important references for this topic.

This paper seeks to present an optimal design of digital analysis filters based on PSO-BPNN for aliasing errors cancellation in HFB DAC. First, we develop a mathematical model of the HFB DAC to calculate the estimation error between the desired and practical transfer functions. Then, by summing the real and imaginary components of estimation error, we derive the approximation error, BPNN is proposed to minimize the approximation error. To reduce the computational complexity of the traditional BPNN design, we also proposed a PSO algorithm [31] for optimizing all the sub-filter orders, the number of the BPNN hidden layer neurons and the iterations of the BPNN, enabling that the given upper limit errors (upper limits of distortion and aliasing errors) is met. Finally, the optimized all the sub-filter orders, the number of the BPNN hidden layer neurons and the iterations of the BPNN are applied to the BPNN method to derive the optimal coefficients for the digital analysis filters. Additionally, this paper derives the computational complexity of PSO-BPNN.

The key contributions outlined in this paper are as follows:

1) Within the context of HFB DAC applications, we propose an optimal design of digital analysis filters using PSO-BPNN to achieve the aliasing errors cancellation.
2) To reduce the computational complexity of the traditional BPNN design, we also proposed a PSO algorithm for optimizing all the sub-filter orders, the number of the BPNN hidden layer neurons and the iterations of the BPNN, enabling that the given upper limit errors (upper limits of distortion and aliasing errors) is met.
3) This paper compares and analyzes the effectiveness and computational complexity of digital analysis filters designed using various methods, including WLS, WLS-Optimization, TNN, BPNN, and PSO-BPNN.
4) Simulation results indicate that, comparing with the other four designs, our proposed PSO-BPNN design not only achieves better aliasing errors cancellation but also reduces all the sub-filter orders, the number of the BPNN hidden layer neurons and the iterations of the BPNN.

The organization of the subsequent sections of this paper is as follows: Section II outlines the derivation of the estimation error. Section III proposes the optimal design based on a BPNN, focusing on minimizing the approximation error using PSO-BPNN. Section IV derives the computational complexity of PSO-BPNN. In Section V, three experiments are conducted, analyzed in depth, and discussed in detail. Ultimately, conclusions are summarized in Section VI.

II. Estimation Error

Fig.1 demonstrates the system block diagram of an M-channel HFB DAC. The desired transfer function can be written as [20]

$$D_i(j\Omega) = \begin{cases} ce^{-j\Omega T_s d}, i = 0 \\ 0, i \neq 0 \end{cases} \quad \Omega \in (-\frac{\pi}{T_s}, \frac{\pi}{T_s}) \tag{1}$$

where c represents the system delay and d denotes the gain in the HFB DAC system. Furthermore, as demonstrated in [20], when the analog filters and mixers in the HFB DAC are non-ideal, the practical transfer function of the M-channel HFB DAC can be expressed as:

$$\begin{aligned} T_i(j\Omega) = & \frac{1}{2MT_s} \sum_{m=0}^{M-1} F_m(e^{j\Omega T_s - j\frac{2\pi i}{M}}) G_m(j\Omega) \\ & \times [H_{ZOH}(j\Omega - j\Omega_m) H_{lpfm}(j\Omega - j\Omega_m) \\ & + H_{ZOH}(j\Omega + j\Omega_m) H_{lpfm}(j\Omega + j\Omega_m)] \end{aligned} \tag{2}$$

where $\Omega_m = \lceil m/2 \rceil 2\pi f_s/M$ and $\lceil \bullet \rceil$ represents the ceiling function. $H_{ZOH}(j\Omega)$ represents the sample-and-hold character of the DAC, while $H_{lpfm}(j\Omega)$ is the anti-imaging low-pass filter that follows the DAC. $T_0(j\Omega)$ denotes the distortion function, representing the magnitude gain and group delay of the HFB DAC with M sub-DACs. $T_i(j\Omega), i =$

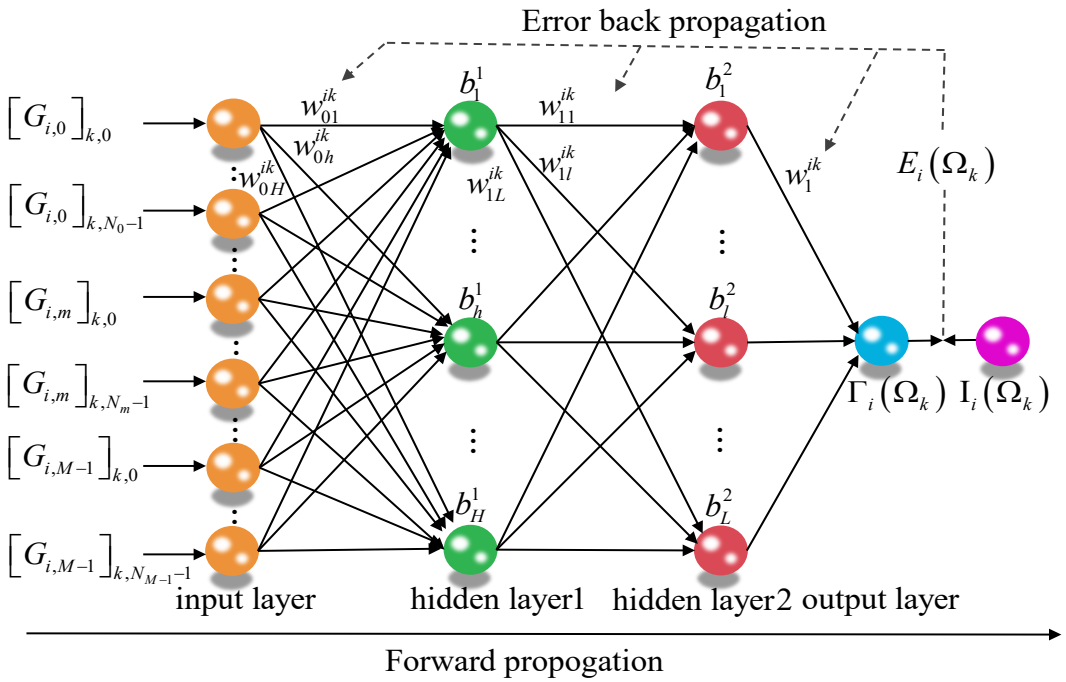


Fig. 2: BPNN model architecture diagram

$1, 2, ..., M-1$ represents the aliasing functions, which are shifted, unwanted versions of the digital wideband input that should ideally be removed. The practical transfer function in Eq. (2) can be redefined as demonstrated in [20]:

$$\begin{aligned} T_i(j\Omega) &= \sum_{m=0}^{M-1} \mathbf{f}_m^T \mathbf{r}_{i,m}(j\Omega) - j \sum_{m=0}^{M-1} \mathbf{f}_m^T \mathbf{i}_{i,m}(j\Omega) \\ &= \mathbf{r}_i^T(\Omega)\mathbf{f} - j\mathbf{i}_i^T(\Omega)\mathbf{f} \\ &= \mathrm{Re}\{T_i(j\Omega)\} - j\mathrm{Im}\{T_i(j\Omega)\} \end{aligned} \tag{3}$$

where $\mathrm{Im}\{\bullet\}$ and $\mathrm{Re}\{\bullet\}$ are the imaginary and real components of a vector or complex number, respectively. $\mathbf{f} = [\mathbf{f}_0^T, \mathbf{f}_1^T, .., \mathbf{f}_{M-1}^T]^T$ with $\mathbf{f}_m = [f_m(0), f_m(1), ..., f_m(N_m-1)]$. Here, N_m represents the order of the m-th digital analysis filter $F_m(z)$. $f_m(\bullet)$ represents the coefficient of $F_m(z)$, which needs to be determined. Further, $\mathbf{r}_i(\Omega) = [\mathbf{r}_{i,0}^T(\Omega), \mathbf{r}_{i,1}^T(\Omega), ..., \mathbf{r}_{i,M-1}^T(\Omega)]^T$ and $\mathbf{i}_i(\Omega) = [\mathbf{i}_{i,0}^T(\Omega), \mathbf{i}_{i,1}^T(\Omega), ..., \mathbf{i}_{i,M-1}^T(\Omega)]^T$ with

$$\begin{aligned} \mathbf{r}_{i,m}(\Omega) = &\frac{1}{2MT_s}\mathbf{c}_i(\Omega)G_m(j\Omega) \\ &\times[H_{ZOH}(j\Omega - j\Omega_m)H_{lpfm}(j\Omega - j\Omega_m) \\ &+H_{ZOH}(j\Omega + j\Omega_m)H_{lpfm}(j\Omega + j\Omega_m)] \end{aligned}$$

$$\begin{aligned} \mathbf{i}_{i,m}(\Omega) = &\frac{1}{2MT_s}\mathbf{s}_i(\Omega)G_m(j\Omega) \\ &\times[H_{ZOH}(j\Omega - j\Omega_m)H_{lpfm}(j\Omega - j\Omega_m) \\ &+H_{ZOH}(j\Omega + j\Omega_m)H_{lpfm}(j\Omega + j\Omega_m)] \end{aligned}$$

where

$$\mathbf{c}_i(\Omega) = [1, \cos(\Omega T_s - \frac{2\pi i}{M}), ..., \cos(N_m - 1)(\Omega T_s - \frac{2\pi i}{M})]^T$$

$$\mathbf{s}_i(\Omega) = [0, \sin(\Omega T_s - \frac{2\pi i}{M}), ..., \sin(N_m - 1)(\Omega T_s - \frac{2\pi i}{M})]^T$$

The estimation error can be calculated using Eq. (1) and Eq. (2) as follows:

$$\begin{aligned} e_i(\Omega) &= T_i(j\Omega) - D_i(j\Omega) \\ &= \left(\mathbf{r}_i^T(\Omega)\mathbf{f} - R_i(\Omega)\right) - j\left(\mathbf{i}_i^T(\Omega)\mathbf{f} - I_i(\Omega)\right) \\ &= \mathrm{Re}\{e_i(\Omega)\} - j\mathrm{Im}\{e_i(\Omega)\} \end{aligned} \tag{4}$$

where, $R_i(\Omega) = \mathrm{Re}\{D_i(j\Omega)\}$, $I_i(\Omega) = \mathrm{Im}\{D_i(j\Omega)\}$.

To cancel the aliasing errors, it is necessary to ensure that the estimation error $e_i(\Omega), i = 0, 1, ..., M-1$ approaches zero.

III. Approximation Error Minimization Based on PSO-BPNN

A. Optimal design of digital analysis filters based on BPNN

Taking into account the output signal $y(t)$ at K discrete frequency points within the frequency range $(-\pi/T_s, \pi/T_s)$, Eq. (4) can also be written as:

$$\begin{aligned} e_i(\Omega_k) &= T_i(j\Omega_k) - D_i(j\Omega_k) \\ &= \left(\mathbf{r}_i^T(\Omega_k)\mathbf{f} - R_i(\Omega_k)\right) - j\left(\mathbf{i}_i^T(\Omega_k)\mathbf{f} - I_i(\Omega_k)\right) \\ &= \mathrm{Re}\{e_i(\Omega_k)\} - j\mathrm{Im}\{e_i(\Omega_k)\} \end{aligned} \tag{5}$$

Since the BPNN model struggles with complex number, Eq. (5) can be rewritten as:

$$\begin{aligned} E_i(\Omega_k) &= \left(\mathbf{r}_i^T(\Omega_k)\mathbf{f} - R_i(\Omega_k)\right) + \left(\mathbf{i}_i^T(\Omega_k)\mathbf{f} - I_i(\Omega_k)\right) \\ &= \mathrm{Re}\{e_i(\Omega_k)\} + \mathrm{Im}\{e_i(\Omega_k)\} \\ &= \Gamma_i(\Omega_k) - \mathrm{I}_i(\Omega_k) \end{aligned} \tag{6}$$

where $E_i(\Omega_k)$ is approximation error, with

$$\Gamma_i(\Omega_k) = \mathrm{Re}\{T_i(j\Omega_k)\} + \mathrm{Im}\{T_i(j\Omega_k)\}$$

$$\mathrm{I}_i(\Omega_k) = \mathrm{Re}\{D_i(j\Omega_k)\} + \mathrm{Im}\{D_i(j\Omega_k)\}$$

Based on Eq. (6), minimizing $E_i(\Omega_k)$ is essential to derive the optimal coefficient vector $\mathbf{f}$. The architecture of the

BPNN model is demonstrated in Fig. 2. The inputs to the BPNN model are:

$$\mathbf{v}_i\left(\Omega_k\right)=\left[\left[\mathbf{G}_{i,0}\right]_k,\left[\mathbf{G}_{i,1}\right]_k,...,\left[\mathbf{G}_{i,M-1}\right]_k\right]^T$$

where

$$[\mathbf{G}_{i,m}]_k=\left[[G_{i,m}]_{k,0},[G_{i,m}]_{k,1},...,[G_{i,m}]_{k,N_m-1}\right]^T$$

$$\begin{aligned}[G_{i,m}]_{k,N_m-1}=&\frac{1}{2MT_s}G_m(j\Omega_k)\\&\times(\cos(N_m-1)(\Omega T_s-\frac{2\pi i}{M})\\&+\sin(N_m-1)(\Omega T_s-\frac{2\pi i}{M}))\\&\times[H_{ZOH}(j\Omega-j\Omega_m)H_{lpfm}(j\Omega-j\Omega_m)\\&+H_{ZOH}(j\Omega+j\Omega_m)H_{lpfm}(j\Omega+j\Omega_m)]\end{aligned}$$

The input and output for the h-th neuron in the first hidden layer are as follows:

$$\mu_h^{N_m-1}=\sum_{m=0}^{M-1}[G_{i,m}]_{k,N_m-1}.w_{(N_m-1)h}^{ik} \tag{7}$$

and

$$b_h^1=f(\mu_h^{N_m-1}-\lambda_h^{N_m-1}) \tag{8}$$

Here, $w_{(N_m-1)h}^{ik}$ denotes the weight linking the input layer to the h-th neuron in the first hidden layer. $\lambda_h^{(Nm-1)}$ denotes the threshold of the first hidden layer, and $f(\bullet)$ denotes the linear activation function. The input and output of the l-th neuron in the second hidden layer are as follows:

$$\zeta_l^{N_m-1}=\sum_{h=1}^{H}b_h^1.w_{(N_m-1)l}^{ik} \tag{9}$$

and

$$b_l^2=f(\zeta_l^{N_m-1}-\alpha_l^{N_m-1}) \tag{10}$$

Here, $w_{(N_m-1)l}^{ik}$ denotes the weight linking the h-th neuron in the first hidden layer to the l-th neuron in the second hidden layer. $\alpha_l^{N_m-1}$ denotes the threshold of the second hidden layer. The inputs and outputs of the output layer $\Gamma_i(\Omega_k)$ are as follows:

$$\theta_k^{N_m-1}=\sum_{l=1}^{L}b_l^2w_l^{ik} \tag{11}$$

and

$$\Gamma_i(\Omega_k)=f(\theta_k^{N_m-1}-\partial_k^{N_m-1}) \tag{12}$$

Here, w_l^{ik} denotes the weight linking the l-th neuron in the second hidden layer to the output layer. $\partial_k^{N_m-1}$ denotes the threshold of the output layer. Thus, the trained BPNN model predicts the actual output as $\Gamma_i(\Omega_k)$ based on the input data.

Using the approximation error as the performance measure the practical and desired output $\mathrm{I}_i(\Omega_k)$, which can be described as:

$$J=\frac{1}{2}\sum_{k=0}^{K-1}\left(E_i(\Omega_k)\right)^2 \tag{13}$$

Algorithm 1 Pseudo-code of PSO-BPNN

1: Set $\boldsymbol{U}$, $\boldsymbol{H}$, $\boldsymbol{L}$, $t_{\text{PSO}}^{\max}$, $\boldsymbol{t}_{\text{BP}}$, and $T_0^{\max}(j\Omega)$, $T_1^{\max}(j\Omega)$, $T_2^{\max}(j\Omega)$,..., $T_{M-1}^{\max}(j\Omega)$.
2: Initialize each particle randomly.
3: Generate $\boldsymbol{N}$, H, L, and t_{BP} from $\boldsymbol{U}$, $\boldsymbol{H}$, $\boldsymbol{L}$, $\boldsymbol{t}_{\text{BP}}$.
4: Calculate particle individual fitness values based on the initial BPNN model.
5: Evaluate each particle to get the global optimal.
6: Update the velocity $v_u(t)$ and position $x_u(t)$ of the particle according to Eq. (15) and Eq. (16).
7: Update group global optimal position $gBest$.
8: Update the individual optimal position of each particle $pBest_u$.
9: $t_{\text{PSO}}=t_{\text{PSO}}+1$
10: **if** $t_{\text{PSO}}<t_{\text{PSO}}^{\max}$ **then**
11: Generate $\boldsymbol{N}$, H, L, and t_{BP} from $\boldsymbol{U}$, $\boldsymbol{H}$, $\boldsymbol{L}$, $\boldsymbol{t}_{\text{BP}}$.
12: **else**
13: Derive the optimal $\boldsymbol{N}$, H, L, and t_{BP}.
14: **end if**
15: Construct the optimized BPNN model using the optimal $\boldsymbol{N}$, H, L, and t_{BP}.
16: Train the optimized BPNN model using the optimal $\boldsymbol{N}$, H, L, and t_{BP}.
17: $t_{\text{BP}}=t_{\text{BP}}+1$
18: **if** $t_{\text{BP}}<t_{\text{BP}}^{\max}$ **then**
19: Train the optimized BPNN model using the optimal $\boldsymbol{N}$, H, L, and t_{BP}.
20: **else**
21: Derive the coefficient vector $\mathbf{f}$.
22: **end if**
23: **if** $T_i(j\Omega)<T_i^{\max}(j\Omega)$ **then**
24: $\mathbf{f}$ is the required coefficient vector.
25: **else**
26: Set $\boldsymbol{U}$, $\boldsymbol{H}$, $\boldsymbol{L}$, $t_{\text{PSO}}^{\max}$, $\boldsymbol{t}_{\text{BP}}$, and $T_i^{\max}(j\Omega)$.
27: **end if**

To minimize the approximation error $E_i(\Omega_k)$, $w_{(N_m-1)h}^{ik}$, $w_{(N_m-1)l}^{ik}$, w_l^{ik}, $\lambda_h^{N_m-1}$, $\alpha_l^{N_m-1}$, $\partial_k^{N_m-1}$ should be updated as

$$\begin{cases}\Delta w_{(N_m-1)h}^{ik}+w_{(N_m-1)h}^{ik}\rightarrow w_{(N_m-1)h}^{ik}\\ \Delta w_{(N_m-1)l}^{ik}+w_{(N_m-1)l}^{ik}\rightarrow w_{(N_m-1)h}^{ik}\\ \Delta w_l^{ik}+w_l^{ik}\rightarrow w_l^{ik}\\ \Delta\lambda_h^{N_m-1}+\lambda_h^{N_m-1}\rightarrow\lambda_h^{N_m-1}\\ \Delta\alpha_l^{N_m-1}+\alpha_l^{N_m-1}\rightarrow\alpha_l^{N_m-1}\\ \Delta\partial_k^{N_m-1}+\partial_k^{N_m-1}\rightarrow\partial_k^{N_m-1}\end{cases} \tag{14}$$

with

$$\Delta w_l^{ik}=-\eta\frac{\partial J}{\partial E_i(\Omega_k)}\frac{\partial E_i(\Omega_k)}{\partial\Gamma_i(\Omega_k)}\frac{\partial\Gamma_i(\Omega_k)}{\partial w_l^{ik}}$$

$$\Delta w_{(N_m-1)l}^{ik}=-\eta\frac{\partial J}{\partial b_l^2}\frac{\partial b_l^2}{\partial w_{(N_m-1)l}^{ik}}$$

$$\Delta w_{(N_m-1)h}^{ik}=-\eta\frac{\partial J}{\partial b_h^1}\frac{\partial b_h^1}{\partial w_{(N_m-1)h}^{ik}}$$

$$\Delta\lambda_h^{N_m-1}=-\eta\frac{\partial J}{\partial\Gamma_i\left(\Omega_k\right)}\frac{\partial\Gamma_i\left(\Omega_k\right)}{\partial\lambda_h^{N_m-1}}$$

$$\Delta\alpha_l^{N_m-1}=-\eta\frac{\partial J}{\partial\Gamma_i\left(\Omega_k\right)}\frac{\partial\Gamma_i\left(\Omega_k\right)}{\partial\alpha_l^{N_m-1}}$$

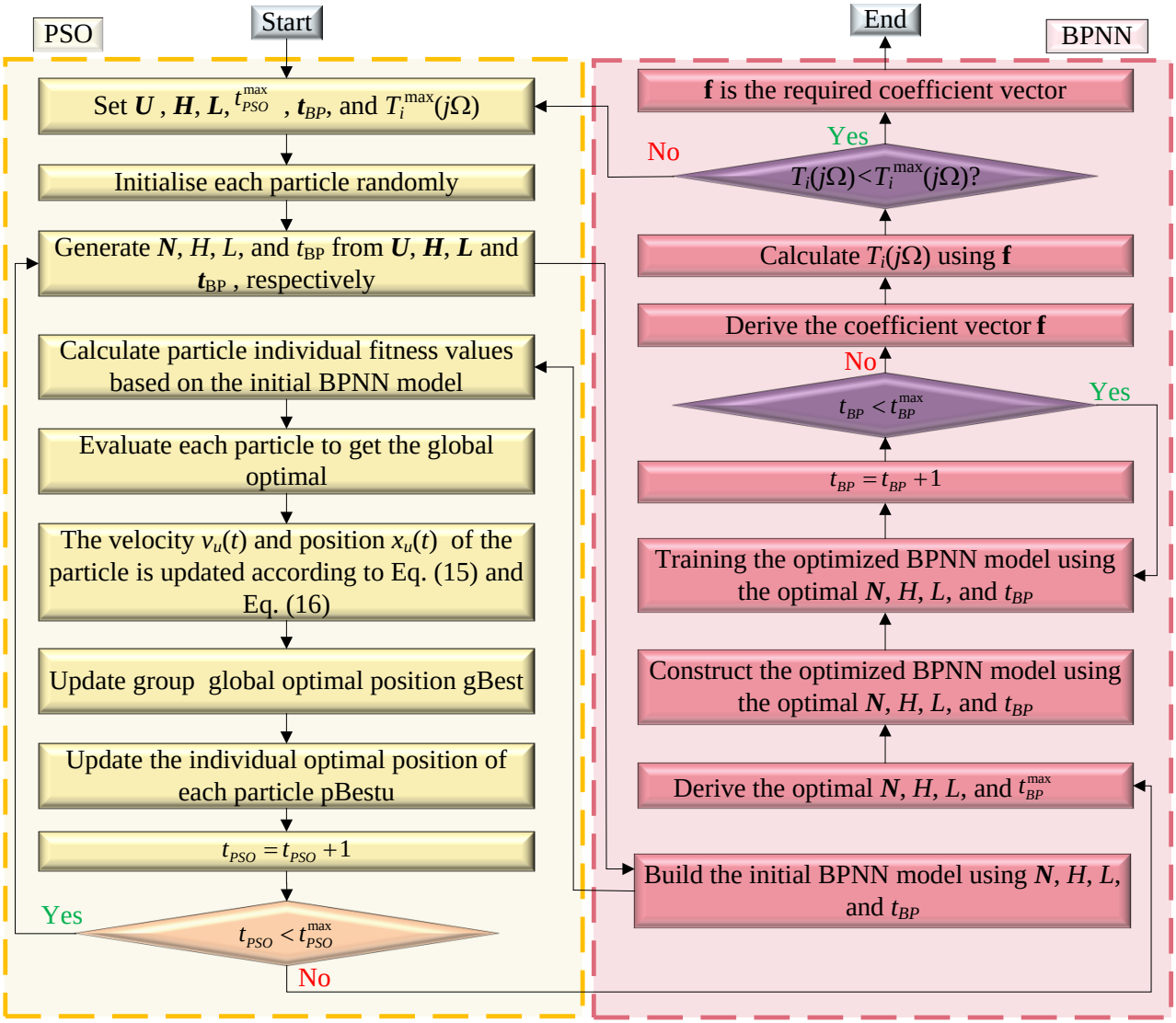


Fig. 3: Parameters optimization flowchart by PSO-BPNN.

$$\Delta\partial_k^{N_m-1} = -\eta \frac{\partial J}{\partial \Gamma_i(\Omega_k)} \frac{\partial \Gamma_i(\Omega_k)}{\partial \partial_k^{N_m-1}}$$

The BPNN iterations are when the maximum number is fulfilled, thus, the coefficients of the digital analysis filters are derived.

B. PSO-BPNN

The PSO algorithm is a population-driven optimization method that leverages swarm intelligence to emulate the foraging dynamics of bird flocks in search of an optimal solution. Each particle represents a potential solution and navigates the search space using both individual and global best experiences. The PSO algorithm, the velocity and position of each particle are updated iteratively he following equations:

$$\begin{aligned} v_u(t+1) =& \omega v_u(t) + c1r1\left(pBest_u - x_u(t)\right) \\ &+ c2r2\left(gBest - x_u(t)\right) \end{aligned} \tag{15}$$

$$x_u(t+1) = x_u(t) + v_u(t+1) \tag{16}$$

Here, $v_u(t)$ denotes the velocity of particle u at time t, $x_u(t)$ denotes its position, $pBest_u$ denotes the individual best position, and $gBest$ denotes the global best position. The parameters ω denotes the inertia weight, c_1 and c_2 are the learning rate, r_1 and r_2 are random values in the range [0, 1]. The PSO method progressively moves toward the global optimum in the search space by applying iterative updates.

Consider the two-dimensional position vector of the particles as the optimal solution. Set the maximum iterations of the PSO as $t_{\mathrm{PSO}}^{\max}$ and the number of particles in the swarm as ϕ. Fig. 3 demonstrates the flowchart of optimizing all the sub-filter orders, the number of the BPNN first hidden layer neurons, the number of the BPNN second hidden layer neurons and the iterations of the BPNN based on the PSO algorithm.

1) Set the range for all the sub-filter orders $\boldsymbol{U}$, the range for the number of the BPNN first hidden layer neurons $\boldsymbol{H}$, the range for the number of the BPNN second hidden layer neurons $\boldsymbol{L}$, the range for the iterations of the BPNN $\boldsymbol{t}_{\mathrm{BP}}$, and the iterations of the PSO algorithm $t_{\mathrm{PSO}}^{\max}$. The given upper limit errors $T_0^{\max}(j\Omega)$, $T_1^{\max}(j\Omega)$, $T_2^{\max}(j\Omega)$,..., $T_{M-1}^{\max}(j\Omega)$ in dB

are the given upper limits for the maximum absolute values of the distortion error $\{T_0(j\Omega)\}$ and the aliasing errors $\{T_1(j\Omega), T_2(j\Omega), ..., T_{M-1}(j\Omega)\}$.

2) Randomly initialize each particle.
3) Generating all the sub-filter orders $\boldsymbol{N} = [N_0, N_1, ..., N_{M-1}]$, the number of the BPNN first hidden layer neurons H, the number of the BPNN second hidden layer neurons L and the iterations of the BPNN t_{BP} in step 1, respectively.
4) Building the initial BPNN model using these parameters generated from step 3.
5) Update the velocity $v_u(t)$ and position $x_u(t)$ of the particle according to Eq. (15) and Eq. (16).
6) Update the group global optimal position $gBest$.
7) Update the individual optimal position $pBest_u$ of each particle.
8) Increment the iterations of the PSO t_{PSO}, if $t_{\text{PSO}} < t_{\text{PSO}}^{\max}$, generate $\boldsymbol{N} = [N_0, N_1, ..., N_{M-1}]$, H, L, and t_{BP} from $\boldsymbol{U}$, $\boldsymbol{H}$, $\boldsymbol{L}$, $\boldsymbol{t}_{\text{BP}}$, otherwise derive the optimal parameters $\boldsymbol{N}$ H, L, t_{BP}.
9) Construct and train the optimized BPNN model using the optimal parameters, increment t_{BP}, if $t_{\text{BP}} < t_{\text{BP}}^{\max}$, train the optimized BPNN model using the optimal $\boldsymbol{N}$, H, L, and t_{BP}, otherwise derive the coefficient vector $\mathbf{f}$.
10) If Eq. (17) is satisfied simultaneously, $\mathbf{f}$ is the required optimal coefficient vector, otherwise, set $\boldsymbol{U}$, $\boldsymbol{H}$, $\boldsymbol{L}$, $t_{\text{PSO}}^{\max}$, $\boldsymbol{t}_{\text{BP}}$, and $T_0^{\max}(j\Omega)$, $T_1^{\max}(j\Omega)$, $T_2^{\max}(j\Omega)$,..., $T_{M-1}^{\max}(j\Omega)$.

$$\begin{cases} T_0(j\Omega) < T_0^{\max}(j\Omega) \\ T_1(j\Omega) < T_1^{\max}(j\Omega) \\ T_2(j\Omega) < T_2^{\max}(j\Omega) \\ \vdots \\ T_{M-1}(j\Omega) < T_{M-1}^{\max}(j\Omega) \end{cases} \tag{17}$$

The effective number of bits (ENOB) is employed to assess the effectiveness of each sub-DAC in the HFB DAC, and the desired ENOB is computed using the following equation:

$$SNR = (6.02ENOB_{bit} + 1.76)dB \tag{18}$$

IV. Derivation of Computational Complexity

This section derives the PSO-BPNN computational complexity. In Part C of Section III, we use PSO to optimize all the sub-filter orders, the number of the BPNN first hidden layer neurons, the number of the BPNN second hidden layer neurons, as well as the iterations of the BPNN, and then apply the optimized parameters to the BPNN model. Thus, the computational complexity of the BPNN model is calculated as follows: During forward propagation, data moves from the input layer to the output layer, undergoing a series of linear transformations and linear activation functions. The steps are as follows:

1) Each input neuron is multiplied by its corresponding weight to feed into the first hidden layer, doing $(K\sum_{m=0}^{M-1} N_m)^2 MH$ multiplications with a computational complexity of $O((K\sum_{m=0}^{M-1} N_m)^2 MH)$.

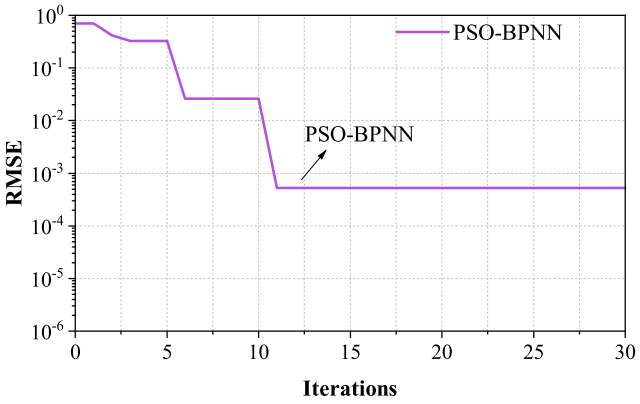


Fig. 4: Loss curves for PSO-BPNN.

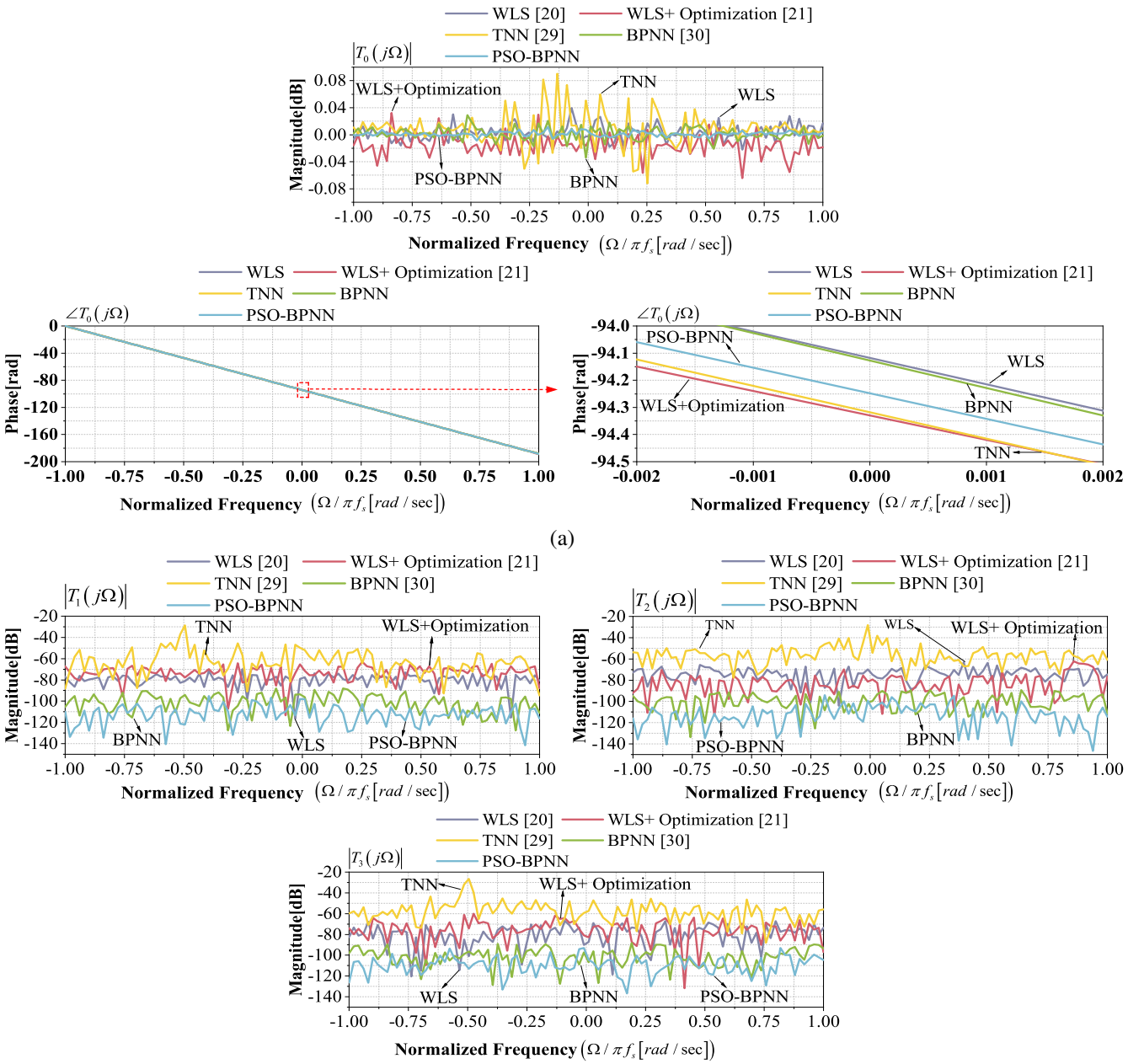
2) The computational complexity of the activation function of all neurons in the first hidden layer is $O(H)$.
3) Each neuron in the first hidden layer is multiplied by its corresponding weight and passed as input to the second hidden layer, doing M^2KL multiplications with a computational complexity of $O(M^2KL)$.
4) The computational complexity of the activation function for all neurons in the second hidden layer is $O(L)$.
5) Each neuron in the second hidden layer is multiplied by its corresponding weight connected to the input of the output layer, doing LMK multiplications with a computational complexity of $O(LMK)$.
6) The computational complexity of the activation function for the output layer is $O(MK)$.

Thus, the overall computational complexity of the forward propagation is:

$$O((K\sum_{m=0}^{M-1} N_m)^2 MH + MKL(1+M) + H + L + MK) \tag{19}$$

During back propagation, the algorithm computes the gradient of the loss function concerning the network parameters, allowing it to update the weights and thresholds. The steps are as follows:

1) The weight update the output layer and the second hidden layer, doing $3MKL$ multiplications with a computational complexity of $O(3MKL)$.
2) The threshold update of the neurons in the output layer, doing $2MKL$ multiplications with a computational complexity of $O(2MKL)$.
3) The weight update the second hidden layer and the first hidden layer, doing $2MKHL$ multiplication with a computational complexity of $O(2MKHL)$.
4) The threshold update of the neurons in the second hidden layer, doing $2MKHL$ multiplications with a computational complexity of $O(2MKHL)$.
5) The weight update the first hidden layer and the input layer, doing $2\sum_{m=0}^{M-1} N_m KH$ multiplication with a computational complexity of $O(2\sum_{m=0}^{M-1} N_m KH)$.
6) The threshold update of the neurons in the first hidden layer, doing $2\sum_{m=0}^{M-1} N_m KH$ multiplications with a



(a)

(b)

Fig. 5: The frequency response results from five different optimal designs. (a) Distortion function $T_0(j\Omega)$ response; (b) Aliasing function $T_i(j\Omega)$ response.

computational complexity of $O(2\sum_{m=0}^{M-1} N_m KH)$.

Thus, the overall computational complexity of back propagation is:

$$O(MK(5L+4HL)+4K\sum_{m=0}^{M-1} N_m H) \tag{20}$$

In summary, the total computational complexity of PSO-BPNN is:

$$O((\sum_{p=1}^{t_{\text{BP}}^{\max}} N_{dp}(K\sum_{m=0}^{M-1} N_m)^2 MH + MKL(1+M) + H + L + MK + MK(5L+4HL) + 4K\sum_{m=0}^{M-1} N_m H) \tag{21}$$

where $t_{\text{BP}}^{\max}$ represents the iterations of the BPNN, respectively. M, K, H and L are the number of channels, the number of discrete frequency points, the number of the BPNN first hidden layer neurons, the number of the BPNN second hidden layer neurons, respectively.

V. Test Validation and Analysis

Several HFB DAC design examples are employed to validate the effectiveness of our proposed an optimal design of digital analysis filters based on PSO-BPNN. The system gain is set to $c = 1$, with a delay parameter of $d = 30$. Additionally, the number of discrete frequency points for the digital analysis filters is set to $K = 100$. The relevant parameters of PSO-BPNN are set as follows: $t_{\text{PSO}}^{\max} = 30$, $\phi = 30$. Further, $\boldsymbol{U}$=[1,59], $\boldsymbol{H}$=[1,63], $\boldsymbol{L}$=[1,31], $\boldsymbol{t}_{\text{BP}}$=[1,99].

TABLE I: SUMMARY OF FIVE DIFFERENT OPTIMAL DESIGN PARAMETERS COMPARISON

Method	Maximum distortion error	Maximum aliasing error	Computational complexity $O(\bullet)$	Sub-filter orders ($\boldsymbol{N}$)	Digital analysis filters length	H	L	Iterations
WLS+Optimization [21]	0.0322dB	-60.16dB	2.30×10^9	$[60, 60, 60, 60]$	240	-	-	-
WLS [20]	0.0396dB	-63.12dB	$\mathbf{2.30 \times 10^7}$	$[60, 60, 60, 60]$	240	-	-	-
TNN [29]	0.0904dB	-26.60dB	5.82×10^8	$[60, 60, 60, 60]$	240	-	-	-
BPNN [30]	0.0290dB	-87.94dB	1.25×10^{13}	$[60, 60, 60, 60]$	240	64	32	100
PSO-BPNN	**0.0118**dB	$\mathbf{-92.91}$dB	9.60×10^{11}	$[\mathbf{18}, \mathbf{25}, \mathbf{13}, \mathbf{9}]$	**65**	**41**	**14**	**31**

TABLE II: SUMMARY OF FIVE DIFFERENT OPTIMAL DESIGN MSE COMPARISON

Method	MSE_0	MSE_1	MSE_2	MSE_3
WLS+Optimization [21]	2.90×10^{-4}	4.93×10^{-6}	4.79×10^{-7}	2.26×10^{-6}
WLS [20]	1.45×10^{-5}	1.29×10^{-6}	4.40×10^{-6}	4.50×10^{-6}
TNN [29]	9.25×10^{-5}	3.02×10^{-4}	1.27×10^{-6}	4.82×10^{-5}
BPNN [30]	3.00×10^{-6}	3.78×10^{-9}	6.86×10^{-9}	6.17×10^{-7}
PSO-BPNN	$\mathbf{5.08 \times 10^{-7}}$	$\mathbf{5.36 \times 10^{-10}}$	$\mathbf{6.52 \times 10^{-11}}$	$\mathbf{1.08 \times 10^{-10}}$

TABLE III: COMPARING WITH THE SUB-FILTER ORDERS OF *M*-CHANNELS FOR FIVE DIFFERENT OPTIMAL DESIGNSf

Method	*M*=2	*M*=4	*M*=6	*M*=8
WLS+Optimization [21]	$[60, 60]$	$[60, 60, 60, 60]$	$[60, 60, 60, 60, 60, 60]$	$[60, 60, 60, 60, 60, 60, 60, 60]$
WLS [20]	$[60, 60]$	$[60, 60, 60, 60]$	$[60, 60, 60, 60, 60, 60]$	$[60, 60, 60, 60, 60, 60, 60, 60]$
TNN [29]	$[60, 60]$	$[60, 60, 60, 60]$	$[60, 60, 60, 60, 60, 60]$	$[60, 60, 60, 60, 60, 60, 60, 60]$
BPNN [30]	$[60, 60]$	$[60, 60, 60, 60]$	$[60, 60, 60, 60, 60, 60]$	$[60, 60, 60, 60, 60, 60, 60, 60]$
PSO-BPNN	$[24, 19]$	$[18, 25, 13, 9]$	$[16, 19, 14, 17, 20, 21]$	$[17, 24, 29, 19, 15, 18, 31, 23]$

We opted to utilize second-order Butterworth filters as the analog LPFs and BPFs.

A. Effectiveness analysis

To validate the effectiveness of our proposed PSO-BPNN, the sampling rate f_s and the number of sub-channels M are set to $f_s = 1/T_s$ and M = 4, respectively. $ENOB_{bit}$ is set to 14 so that the desired SNR of our HFB DAC design examples is 14 × 6.02 +1.76= 86.04dB. Further, the order of each digital analysis filter designed by the design using traditional BPNN is set to 60, i.e., N = 60.

According to the simulation results in [30], the given upper limit errors as $T_0^{\max}(j\Omega)$=0.0290dB, $T_1^{\max}(j\Omega)$=-87.94dB, $T_2^{\max}(j\Omega)$=-89.44dB, $T_3^{\max}(j\Omega)$=-88.29dB. respectively. Fig.4 demonstrates the root-mean-square error (RMSE) loss curve of PSO-BPNN, demonstrating clear convergence.

Fig. 5 and Table I propose the frequency responses of the five different optimal designs in this HFB DAC example. As demonstrated in Table I, the maximum distortion error in the proposed PSO-BPNN optimal design is approximately 0.0118 dB, outperforming the other four optimal designs. As demonstrated in Fig. 5(b) and Table I, the aliasing errors of our proposed PSO-BPNN design is -92.91 dB, respectively. In contrast, the maximum aliasing errors of the optimal designs based on WLS [20], WLS + Optimization [21], TNN [29], and BPNN [30] designs are -63.12 dB, -60.16 dB, -26.60 dB, and -87.94 dB, respectively.

It can also be seen from Table I that the total number of digital analysis filters length, the number of the BPNN first hidden layer neurons, the number of the BPNN second hidden layer neurons and the iterations of the BPNN required for our proposed PSO-BPNN design, comparing with the traditional BPNN design, are 65, 41, 14, and 31. Thus, comparing with the traditional BPNN design, our proposed PSO-BPNN reduces the total number of digital analysis filters length, the number of the BPNN first hidden layer neurons, the number of the BPNN second hidden layer neurons and the iterations of the BPNN by about 72.92%, 35.94%, 56.25% and 69%, respectively. Following optimizing all the sub-filters using the PSO, all the sub-filter orders were optimized to N=$[N_0, N_1, N_2, N_3]$=$[18, 25, 13, 9]$.

In summary, comparing with the traditional BPNN design [30], our proposed PSO-BPNN not only achieves better aliasing errors cancellation but also reduces the computational complexity. Comparing with the WLS [20], WLS + Optimization [21], and TNN [29] designs, our proposed PSO-BPNN achieves better aliasing errors cancellation, at the cost of increased computational complexity.

B. Mean squared error analysis

The effectiveness analysis in this paper could also be an MSE analysis. The MSE equation is as follows:

$$MSE_i = \frac{1}{K} \sum_{k=0}^{K-1} (\Gamma_i(\Omega_k) - \mathrm{I}_i(\Omega_k))^2, i = 0, 1, 2, 3 \quad (22)$$

Here, MSE_0 is the MSE of the distortion function and MSE_1, MSE_2, MSE_3 denotes the MSE of the aliasing function.

As can be seen from Table II, the MSE_0, MSE_1, MSE_2 and MSE_3 of PSO-BPNN are 5.08×10^{-7}, 5.36×10^{-10}, 6.52×10^{-11} and 1.08×10^{-10}, respectively. Compared with the MSE of the WLS [20], WLS+Optimisation [21], TNN

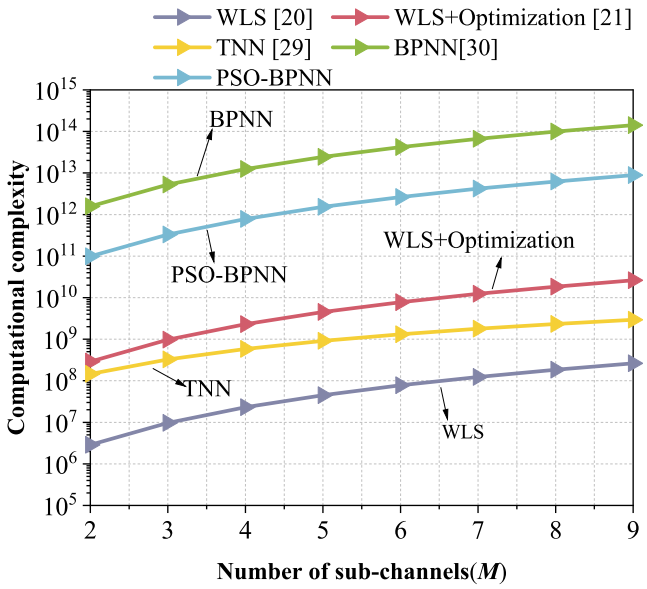


Fig. 6: Computational complexity comparative analysis of different optimal designs with different number of sub-channels when N = 60.

[29], and BPNN [30] based designs, the optimal design MSE based on PSO-BPNN proposed in this paper is significantly lower.

C. Computational complexity analysis

Fig. 6 demonstrates a comparison of computational complexity among the optimal design digital analysis filters based on WLS [20], WLS+Optimization [21], TNN [29], BPNN [30] designs, and we proposed PSO-BPNN in this paper. Table III compares the sub-filter orders of the M-channels for five different optimal designs. Among them, the optimal design proposed in this paper has the order $\boldsymbol{N}$ = [24 19], $\boldsymbol{N}$ = [18 25 13 9], $\boldsymbol{N}$ = [16 19 14 17 20 21], and $\boldsymbol{N}$ = [17 24 29 19 15 18 31 23] for the number of channels M=2, 4, 6, and 8, while the other four optimal designs have the order $\boldsymbol{N}$ = [60 60], $\boldsymbol{N}$ = [60 60 60 60], $\boldsymbol{N}$ = [60 60 60 60 60 60], and $\boldsymbol{N}$ = [60 60 60 60 60 60 60 60] for the number of channels M=2, 4, 6, and 8, respectively. Fig. 6 demonstrates that as the number of channels (M) increases, the computational complexity of the five optimal designs also increases. The computational complexity of these five optimal designs, as demonstrated in Fig. 6, is ranked from highest to lowest as follows:

$$[30] > \text{PSO-BPNN} > [21] > [29] > [20]$$

VI. Conclusion

This paper proposes an optimal design of digital analysis filters based on PSO-BPNN for the aliasing errors cancellation in HFB DAC. The HFB DAC is initially modeled mathematically to calculate the estimation error between the desired and practical transfer functions. The real and imaginary components of the estimation error are then summed to derive the approximation error, BPNN is introduced to minimize the approximation error. To reduce the computational complexity of the traditional BPNN design, we also introduced a PSO algorithm for optimizing all the sub-filter orders, the number of the BPNN hidden layer neurons and the iterations of the BPNN, enabling that the given upper limit errors (upper limits of distortion and aliasing errors) is met. Finally, the optimized parameters are applied to the BPNN model to derive the optimal coefficients for the digital analysis filters. Additionally, this paper derives the computational complexity of PSO-BPNN. We validated the effectiveness of our PSO-BPNN optimal design through several design examples. Simulation results indicate that, comparing with the traditional BPNN design, our proposed PSO-BPNN design not only achieves better aliasing errors cancellation but also reduces all the sub-filter orders, the number of the BPNN hidden layer neurons and the iterations of the BPNN. Comparing with WLS, WLS+Optimization and TNN designs, our proposed PSO-BPNN achieves better aliasing errors cancellation, at the cost of increased computational complexity.

References

[1] S. Achar and D. Jayadevappa, "Optimized recurrent based training accelerator for network-on-chip communication system," IAENG International Journal of Computer Science, vol. 50, no. 4, pp1410-1419, 2023.

[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, pp159-173, 2022.

[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, pp035110, 2017.

[4] 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 Processing, vol. 123, pp103399, 2022.

[5] T. O. Dickson, Z. T. Deniz, M. Cochet, T. J. Beukema, M. Kossel, T. Morf, Y.-H. Choi, P. A. Francese, M. Br¨andli, C. W. Baks et al., "A 72-GS/s, 8-bit DAC-based wireline transmitter in 4-nm FinFET CMOS for 200+ Gb/s serial links," IEEE Journal of Solid-State Circuits, vol. 58, no. 4, pp1074–1086, 2023.

[6] B. Moeneclaey, M. Verplaetse, H. Ramon, N. Singh, H. Li, J. Van Kerrebrouck, X. Yin, and G. Torfs, "A 6-bit 56-GSa/s DAC in 55 nm SiGe BiCMOS," IEEE BiCMOS and Compound Semiconductor Integrated Circuits and Technology Symposium (BCICTS), pp1–4, 2021.

[7] Q. Qi, M. Wang, Y. Yang, H. Hu, Y. Guo, X. Li, Y. Zhang, and Y. Zhang, "A 12-bit 4gs/s dac based on cmos/inp heterogeneous integration," IEEE 6th International Conference on Integrated Circuits and Microsystems (ICICM), pp201–204, 2021.

[8] 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, pp115103, 2018.

[9] W. Zhao, S. Tian, K. Liu, L. Huang, G. Guo, and Q. Wu, "Low spurious waveform synthesis based on digital resampling," IEEE 15th International Conference on Electronic Measurement & Instruments (ICEMI), pp276–280, 2021.

[10] Y. Xiao, L. Huang, W. Zhao, and K. Liu, "The design of a software defined arbitrary waveform generator," IEEE AUTOTESTCON, pp1–6, 2019.

[11] Y. Xiao, G. Xiao, K. Liu, Z. Fu, and H. Wang, "Constraint Models of SDRAM Based Arbitrary Waveform Generator," IEEE Transactions on Instrumentation and Measurement, vol. 72, pp1–10, 2023.

[12] W. Zhao, S. Tian, K. Liu, and Y. Xiao, "A comprehensive estimation method of phase error in the frequency-interleaving digital-to-analog converter," Electronics Letters, vol. 59, no. 2, ppe12707, 2023.

[13] L. Yin, S. Tian, K. Liu, G. Guo, and Y. Xiao, "An estimation method of timing mismatch error in hybrid filter bank DACs," IEICE Electronics Express, vol. 17, no. 9, pp20200126-20200126, 2020.

[14] 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," IEEE AUTOTESTCON, pp1–7, 2022.

[15] W. Liu, K. Liu, Y. Shen, Z. Fu, G. Guo, and W. Zhao, "Support for mimo pre-equalization in BI-DAC system: An accurate frequency response obtaining method," IEEE 16th International Conference on Electronic Measurement & Instruments (ICEMI), pp426–430, 2023.

[16] Y. Shen, K. Liu, W. Liu, G. Guo, W. Zhao, and Z. Fu, "BI-DAC system with guard band: Trade-off between bandwidth and aliasing suppression," IEEE 16th International Conference on Electronic Measurement & Instruments (ICEMI), pp391-396, 2023.
[17] W. Zhao, S. Tian, K. Liu, Y. Shen, G. Guo, and Q. Wu, "A magnituderesponse compensation method in the FI-DAC-based arbitrary waveform generator," IEEE 16th International Conference on Electronic Measurement & Instruments (ICEMI), pp397–403, 2023.
[18] X. Yang, H. Wang, K. Liu, Y. Xiao, Z. Fu, and G. Guo, "Minimax design of digital FIR filters using linear programming in bandwidth interleaving digital-to-analog converter," IEICE Electronics Express, vol. 15, no. 13, pp20180565–20180565, 2018.
[19] X. Chen, S. Chandrasekhar, S. Randel, G. Raybon, A. Adamiecki, P. Pupalaikis, and P. J. Winzer, "All-electronic 100-GHz bandwidth digital-to-analog converter generating PAM signals up to 190 GBaud," Journal of Lightwave Technology, vol. 35, no. 3, pp411–417, 2017.
[20] X. Yang, K. Liu, M. Zhang, and X. Chen, "Noniterative WLS design of digital FIR filters for aliasing errors cancellation in bandwidth interleaving digital-to-analog converter," IEICE Electronics Express, vol. 19, no. 4, pp20210507–20210507, 2022.
[21] L. Yin, S. Tian, K. Liu, G. Guo, and Y. Xiao, "Optimization of synthesis filters for hybrid filter bank DACs," IEICE Electronics Express, vol. 16, no. 6, pp20190053–20190053, 2019.
[22] B. Wang and W. Zhang, "Research on edge network topology optimization based on machine learning," IEEE 5th International Conference on Applied Machine Learning (ICAML), pp41–46, 2023.
[23] Y. Hu, W. Lv, Z. Wang, L. Liu, and H. Liu, "Error prediction of balancing machine calibration based on machine learning method," Mechanical Systems and Signal Processing, vol. 184, pp109736, 2023.
[24] M. Levine and A. Stuart, "A framework for machine learning of model error in dynamical systems," Communications of the American Mathematical Society, vol. 2, no. 07, pp283–344, 2022.
[25] Z. Li, X. Luo, M. Liu, X. Cao, S. Du, and H. Sun, "Wind power prediction based on EEMD-Tent-SSA-LS-SVM," Energy Reports, vol. 8, pp3234–3243, 2022.
[26] W. Tian, Y. Bao, and W. Liu, "Wind power forecasting by the bp neural network with the support of machine learning," Mathematical Problems in Engineering, vol. 2022, no. 1, pp7952860, 2022.
[27] X. Ye and Z. LV, "Distribution network theoretical line loss analysis based on bp neural network algorithm," IEEE 3rd International Conference on Consumer Electronics and Computer Engineering (ICCECE), pp551–554, 2023.
[28] L. Jiang, W. Zhang, S. Jinmiao, Y. Yangfan, and Z. Shuhua, "Vibration suppression of flexible joints space robot based on neural network," IAENG International Journal of Applied Mathematics, vol. 52, no. 4, pp776-783, 2022.
[29] C.-C. Tseng and S.-L. Lee, "A supervised learning method for the design of linear phase fir digital filter using keras," IEEE International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS), pp1–2, 2019.
[30] 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, pp20220491–20220491, 2023.
[31] H. Miyajima, N. Shigei, H. Miyajima, and N. Shiratori, "Securelydistributed computation with divided data for particle swarm optimization," Proceedings of the International MultiConference of Engineersand Computer Scientists (IMECS), pp1–6, 2021.