

Uma Arquitetura Híbrida Baseada em Janelamento e Redes Neurais para Projeto de Filtros FIR

Wheidima Carneiro de Melo¹, Diego Alves Amoedo², Waldir Sabino da Silva Júnior², Eddie Batista de Lima Filho^{2,3}

¹Universidade do Estado do Amazonas (UEA), AM, Brasil

²Universidade Federal do Amazonas (UFAM), AM, Brasil

³TPV Technology, AM, Brasil

Emails: wmelo@uea.edu.br, {diegoamoedo, waldirjr}@ufam.edu.br, eddie.filho@tpv-tech.com

Resumo— Este artigo propõe uma arquitetura híbrida para o projeto de filtros FIR, combinando janelamento, redes neurais e otimização. Ela gera uma envoltória de ponderação digital, ajustada iterativamente por uma função de perda, que penaliza violações de *ripple* e de atenuação. O filtro resultante é gerado pelo produto entre essa envoltória e o filtro ideal. Resultados experimentais demonstram a superioridade da abordagem proposta frente aos métodos clássicos de Parks-McClellan e janelamento de Kaiser, indicando o potencial para implementações em hardware mais eficientes, com menor latência e consumo energético.

Palavras-Chave— Projeto de filtros digitais, filtros FIR, redes neurais, otimização.

Abstract— This paper proposes a hybrid architecture for FIR filter design, combining windowing, neural networks, and optimization. It generates a digital weighting envelope, iteratively adjusted by a loss function, that penalizes ripple and attenuation violations. The resulting filter is generated by the product of this envelope and the ideal filter. Experimental results demonstrate the superiority of the proposed approach over the classical Parks-McClellan and Kaiser windowing methods, indicating the potential for more efficient hardware implementations with lower latency and energy consumption.

Keywords— Digital filter design, FIR filters, neural networks, optimization.

I. INTRODUÇÃO

Técnicas de processamento digital de sinais (PDS) são amplamente aplicadas a diferentes áreas, incluindo medicina [1], telecomunicações [2] e processamento de áudio [3] e vídeo [4], entre outras [5]. Isso ocorre porque tais domínios demandam tarefas complexas, exigindo precisão e robustez. Além disso, a evolução e o aumento significativo da capacidade computacional de dispositivos eletrônicos favoreceram a implementação e a disseminação dessas técnicas, mesmo as mais sofisticadas, bem como o seu emprego em aplicações de tempo real.

Entre as técnicas de PDS, a filtragem digital desempenha um papel fundamental, sendo responsável por manipular ou extrair informações relevantes de diversos tipos de sinais. Isso permite, por exemplo, a remoção de ruído [6], a atenuação de faixas de frequência [7], a separação de componentes [8] e o realce de características [9]. Além disso, técnicas de filtragem também são utilizadas como etapas de pré-processamento em sistemas de tratamento de dados, com o objetivo de remover

informações irrelevantes e facilitar a detecção e a extração de características discriminativas [10], [11], [12], [13].

Filtros digitais podem ser categorizados em dois tipos: resposta ao impulso de duração finita (*Finite-duration Impulse Response* – FIR) e resposta ao impulso de duração infinita (*Infinite-duration Impulse Response* – IIR). Além disso, filtros FIR possuem algumas importantes vantagens sobre filtros IIR, tais como estabilidade inerente, possibilidade de resposta de fase linear e baixa sensibilidade de coeficiente, tornando o seu emprego particularmente atrativo em diversas aplicações [14], [15], [16]. Por outro lado, filtros FIR geralmente requerem um número maior de coeficientes, o que implica maior demanda de recursos computacionais e consumo de energia.

Diante desse cenário, o grande desafio no projeto de filtros FIR reside em satisfazer especificações de resposta em frequência com um número reduzido de coeficientes. Em geral, esse objetivo é perseguido com o uso de métodos clássicos de projeto, como as técnicas de janelamento, entre elas a janela de Kaiser [17], as quais empregam janelas fixas, e os consagrados algoritmos de otimização, entre eles o de Parks-McClellan [18]. Tais abordagens proporcionam simplicidade, muitas vezes com opções de manutenção e controle de aspectos, no primeiro caso, e otimização do erro máximo, no segundo.

No entanto, para atenderem a critérios de projeto, especialmente os mais estritos, as abordagens mencionadas frequentemente resultam em filtros de ordem elevada, exacerbando o custo computacional e o atraso de processamento resultante (latência). De fato, métodos de janelamento espalham o erro obtido, geralmente utilizando muito espaço para suavizá-lo, ao passo que algoritmos de otimização acabam precisando de muitos graus de liberdade para convergência. Por outro lado, uma rede neural vai além de uma formulação fechada e pode ser treinada para se adaptar às necessidades de sistemas reais, gerando filtros aceitáveis, realizando otimização não uniforme e criando matrizes de filtros esparsas, com um melhor compromisso entre tempo e frequência.

Como consequência, o presente artigo endereça os pontos levantados e propõe uma arquitetura híbrida que combina janelamento, redes neurais artificiais e otimização para o projeto de filtros FIR, empregando um mapeamento paramétrico que alcance o resultado desejado. A resposta ao impulso efetiva

e de duração finita de um filtro, $h(n)$, é obtida através do produto de Hadamard entre a resposta de filtro ideal e o vetor de ativação de uma rede neural, que exprime uma envoltória de ponderação digital $w_\theta(n)$. Além disso, um algoritmo de otimização realiza a atualização iterativa de pesos e vieses, forçando a rede neural mencionada a ajustar $w_\theta(n)$ até que as restrições de projeto, expressas como uma função de perda, sejam devidamente satisfeitas.

As principais contribuições deste artigo podem ser resumidas da seguinte forma:

- uma metodologia baseada em redes neurais com estruturas adaptativas para a ponderação de respostas ideais de filtros digitais, de forma otimizada;
- uma função de perda formulada especificamente para penalizar violações em restrições de projetos de filtros, de forma analítica e exata;
- resultados experimentais que demonstram a efetividade do método proposto e delineiam futuras comparações.

II. MÉTODO PROPOSTO

Redes neurais artificiais [19] são modelos computacionais fundamentados na interconexão de unidades de processamento simples, denominadas neurônios, e organizadas em camadas. No contexto de PDS e aproximação de funções, as redes neurais destacam-se pela sua capacidade intrínseca de mapear relações complexas entre entradas e saídas, funcionando como aproximadores universais. O seu aprendizado ocorre por meio do ajuste iterativo de seus parâmetros internos, com o objetivo de minimizar uma função de custo predefinida. Tal processo é tipicamente realizado com algoritmos de otimização (como o Gradiente Descendente [20] ou Adam [21]), que atualizam parâmetros com base em gradientes calculados de forma eficiente pelo algoritmo de *backpropagation* [22].

Na Fig. 1, a metodologia proposta, baseada em redes neurais, é apresentada. De forma simplificada, ela pode ser dividida em três estágios: **Geração de Coeficientes**, **Geração de Características** e **Otimização**. Ela utiliza como entrada as seguintes especificações desejadas: frequência de amostragem (f_s), número de coeficientes (M), frequência analógica de corte (f_c), faixa de transição analógica (T) e erro máximo de aproximação ou *ripple* (δ) aceitável. No geral, o esquema resultante opera em malha fechada, onde as especificações de projeto são processadas para a geração de um filtro inicial, cuja resposta é iterativamente refinada.

A. Geração de Coeficientes

O estágio de **Geração de Coeficientes** inicia pela obtenção da frequência digital de corte (ω_c) e da faixa de transição digital (T_r), nomeada **Mapeamento de Frequência**, a partir das especificações f_s , f_c e T , utilizando:

$$\omega_c = \frac{2\pi f_c}{f_s}, \quad T_r = \frac{2\pi T}{f_s}. \quad (1)$$

Em resumo, ω_c atua como um vetor contendo as frequências de corte mapeadas, possuindo um único elemento, para filtros passa-baixa e passa-alta, ou dois, ω_{c1} e ω_{c2} , para filtros passa-faixa e rejeita-faixa.

Com base nos parâmetros obtidos e no comprimento de filtro M , a etapa **Síntese de Filtro Ideal** calcula a resposta ao impulso $h_{id}(n)$. Para unificar o projeto dos quatro tipos clássicos de filtros (passa-baixa, passa-alta, passa-faixa e rejeita-faixa) em uma única equação, a resposta ao impulso ideal é formulada analiticamente como a soma de K bandas de passagem: $K = 1$ para as seletividades passa-baixa, passa-alta e passa-faixa, ao passo que $K = 2$ para a rejeita-faixa. Assumindo que a i -ésima banda de passagem seja delimitada no intervalo de frequências $[\omega_{a,i}, \omega_{b,i}]$, cujos limites são extraídos diretamente das componentes do vetor ω_c e dos extremos do espectro digital (0 e π), a resposta ao impulso unificada, centrada na origem, é dada por:

$$h_{id}(n) = \begin{cases} \sum_{i=1}^K \frac{\omega_{b,i} - \omega_{a,i}}{\pi}, & \text{se } n = 0 \\ \sum_{i=1}^K \frac{\sin(\omega_{b,i}n) - \sin(\omega_{a,i}n)}{\pi n}, & \text{se } n \neq 0. \end{cases} \quad (2)$$

Nos métodos de aproximação de filtros digitais baseados em janelamento, $h_{id}(n)$ é truncada por uma função fixa de suporte finito $w(n)$, gerando $h(n)$. Em contraste, a metodologia proposta substitui esta última por um mapeamento paramétrico conduzido por uma rede neural artificial de camada única e definida pelos parâmetros treináveis θ : uma matriz de pesos sinápticos e um vetor de vies. Em **Ponderação Digital Baseada em Redes Neurais**, a rede neural de camada única recebe como entrada a resposta ao impulso ideal completa $h_{id}(n)$. Para um projeto de número de coeficientes M (onde M é ímpar), a arquitetura é projetada para produzir uma envoltória de ponderação w_θ contendo apenas $L = \lfloor M/2 \rfloor + 1$ coeficientes treináveis. O filtro $h(n)$ é obtido pelo produto de Hadamard entre a envoltória w_θ e a metade causal da resposta ideal. Para compor a resposta ao impulso completa de comprimento M (ímpar), impõe-se a simetria par em torno do coeficiente central, garantindo a fase linear do filtro, conforme a relação:

$$h(n) = h(-n), \quad |n| \leq \lfloor \frac{M}{2} \rfloor. \quad (3)$$

Dessa forma, a rede neural otimiza a forma da janela para satisfazer as restrições espectrais, enquanto a estrutura de reflexão impõe a linearidade de fase de forma analítica e exata. Uma vez alcançada a convergência, o conjunto otimizado de parâmetros consolida a resposta ao impulso $h(n)$, garantindo o pleno atendimento às restrições de projeto.

B. Geração de Características

Para o cumprimento das especificações de projeto, é imprescindível avaliar quantitativamente o comportamento de um filtro que está sendo gerado. Com essa finalidade, introduzimos o estágio de **Geração de Características**. Essa etapa calcula a resposta ao impulso $h(n)$ no domínio da frequência, por meio da Transformada Rápida de Fourier (*Fast Fourier Transform* - FFT), e, posteriormente, a magnitude de sua transformada. Em seguida, utilizando ω_c e T_r , ela segmenta o espectro resultante. Esse processo isola os vetores correspondentes exclusivamente à faixa de passagem e à faixa de rejeição, fornecendo os

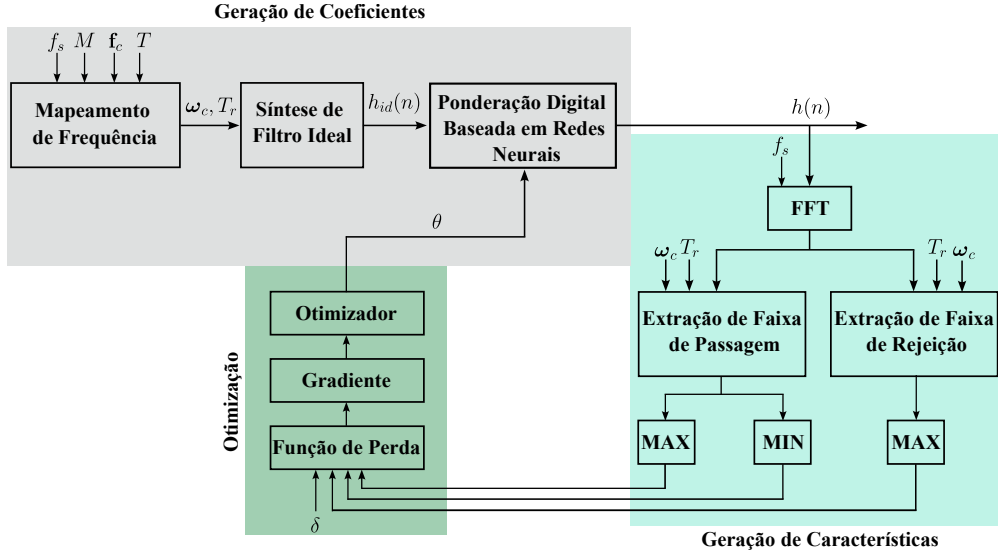


Fig. 1. A metodologia proposta para geração de filtros FIR. As fases de **Geração de Coeficientes**, **Geração de Características** e **Otimização** são destacadas em cinza, verde claro e verde escuro, respectivamente.

subsídios exatos para que **Função de Perda** penalize eventuais violações de *ripple* e atenuação.

Com as informações espectrais devidamente segmentadas, a metodologia proposta avalia o desempenho do pior caso de um filtro em cada região, através dos operadores de máximo (MAX) e mínimo (MIN). Especificamente, em **Extração de Faixa de Passagem**, tais funções são aplicadas sobre o vetor de magnitude isolado para capturar os picos superiores e os vales inferiores do *ripple*, permitindo quantificar, com precisão, a máxima excursão da amplitude em relação ao ganho unitário ideal. Paralelamente, em **Extração de Faixa de Rejeição**, a operação MAX é utilizada na identificação da componente espectral de maior amplitude, a qual dita a atenuação mínima efetiva do filtro (o limite de vazamento de sinal). Esses escalares tornam-se as métricas definitivas que guiarão **Função de Perda**, forçando os parâmetros da rede neural a convergirem no sentido de mitigar as violações espectrais mais críticas do projeto.

C. Otimização

Para conduzir a convergência da rede neural, **Função de Perda**, chamada aqui de função de perda total $\mathcal{L}_{\text{total}}$, foi formulada como a soma de três parcelas de erro quadrático. Cada uma penaliza especificamente as violações dos limites extraídos no estágio anterior, em relação aos parâmetros de *ripple* permitido na banda de passagem e de atenuação na banda de rejeição. O cálculo é decomposto da seguinte forma:

- 1) A primeira parcela avalia o limite inferior do *ripple* na faixa de passagem, onde o erro é calculado através da diferença quadrática entre o valor mínimo extraído ($\min(|H_{\text{pass}}|)$) e o limite de tolerância inferior desejado, fixado em $(1 - \delta)$, ou seja,

$$\mathcal{L}_{p,\min} = (\min(|H_{\text{pass}}|) - (1 - \delta))^2; \quad (4)$$

- 2) A segunda parcela foca no limite superior do *ripple* na

faixa de passagem, garantindo que o pico de amplitude não ultrapasse a tolerância de $(1 + \delta)$, dada por

$$\mathcal{L}_{p,\max} = (\max(|H_{\text{pass}}|) - (1 + \delta))^2; \quad (5)$$

- 3) A terceira parcela quantifica a eficiência de atenuação do filtro, onde o erro é mensurado pela diferença quadrática entre o maior lóbulo da faixa de rejeição (pior caso de vazamento) e o limite máximo permitido δ , usando

$$\mathcal{L}_{r,\max} = (\max(|H_{\text{rej}}|) - \delta)^2; \quad (6)$$

- 4) Por fim, a função de perda total, que atua como o critério de custo para o algoritmo de *backpropagation* e atualização dos pesos θ , é dada pela sobreposição direta dessas três penalidades, ou seja,

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{p,\min} + \mathcal{L}_{p,\max} + \mathcal{L}_{r,\max}. \quad (7)$$

Cabe ressaltar que a formulação descrita acima atende diretamente aos casos de filtros com apenas uma faixa de passagem e uma de rejeição, ou seja, filtros passa-baixa e passa-alta. Para acomodar outras seletividades, a métrica é expandida. No projeto de um filtro passa-faixa, que possui duas regiões de rejeição, adicionamos um termo extra à $\mathcal{L}_{\text{total}}$: o erro máximo da segunda faixa de rejeição $\mathcal{L}_{r2,\max}$. De forma análoga, no projeto de um filtro rejeita-faixa, que apresenta duas bandas de passagem, incorporamos dois termos adicionais para avaliar os limites inferior e superior da segunda faixa de passagem: $\mathcal{L}_{p2,\min}$ e $\mathcal{L}_{p2,\max}$, garantindo a otimização simultânea de todo o espectro desejado.

Conforme ilustrado em **Otimização**, na arquitetura proposta (ver Fig. 1), $\mathcal{L}_{\text{total}}$ atua como critério de custo. A partir dessa perda, **Gradiente** emprega o algoritmo de *backpropagation* para computar as derivadas parciais de erro em relação a cada parâmetro da rede. Por fim, **Otimizador** emprega o algoritmo Adam para realizar a atualização iterativa dos pesos e vieses (θ), com base nesses gradientes, forçando a rede neural a

ajustar a envoltória $w_\theta(n)$ até que as restrições de projeto sejam devidamente satisfeitas.

D. Implementação

A implementação prática da arquitetura descrita foi desenvolvida em linguagem Python, fazendo uso do framework de aprendizado profundo PyTorch para a computação dos gradientes e a execução do otimizador. O treinamento da rede neural é realizado de forma iterativa até que os critérios de parada sejam atingidos, ou seja, quando o número de épocas é suficiente para que a envoltória $w_\theta(n)$ resulte em um filtro que satisfaça integralmente as especificações de projeto fornecidas.

III. RESULTADOS E DISCUSSÃO

Para validar a eficácia do método proposto, realizamos diversos experimentos abrangendo os quatro tipos clássicos de filtros: passa-baixa (PB), passa-alta (PA), passa-faixa (PF) e rejeita-faixa (RF). Comparamos o método proposto ao de janelamento de Kaiser e ao algoritmo de Parks-McClellan, de modo que todas as restrições de projeto fossem atendidas pelos métodos de projeto testados. Em todas as simulações, utilizamos $f_s = 1000$ Hz e uma faixa de transição de 4 Hz. O objetivo principal foi determinar o número de coeficientes mínimo M para o atingimento de atenuações de -40 dB, -50 dB e -60 dB, mantendo o *ripple* na banda de passagem dentro da tolerância desejada (entre $1 - \delta$ a $1 + \delta$). Os resultados obtidos em todos os experimentos estão apresentados na Tabela I.

TABELA I

COMPARAÇÃO DE NÚMERO DE COEFICIENTES DO FILTRO (M) PARA DIFERENTES SELETIVIDADES E ATENUAÇÕES.

Tipo	Atenuação	Kaiser	Parks-McClellan	Proposto
PB	-40 dB	561	488	467
PB	-50 dB	733	652	585
PB	-60 dB	905	817	685
PA	-40 dB	561	485	467
PA	-50 dB	733	651	585
PA	-60 dB	907	817	685
PF	-40 dB	561	491	473
PF	-50 dB	733	649	589
PF	-60 dB	907	821	691
RF	-40 dB	561	491	473
RF	-50 dB	733	653	589
RF	-60 dB	907	821	689

Analisando os resultados obtidos, observamos que o método proposto atende rigorosamente às especificações de projeto com o menor número de coeficientes, conforme destacado em negrito. Para ilustrar o comportamento dos filtros no domínio da frequência, as Figuras 2 e 3 apresentam as respostas em magnitude para os projetos PB a -60 dB e PA a -50 dB, respectivamente. Nessas imagens, verificamos que a atenuação mínima na banda de rejeição é atingida, mantendo o *ripple* na banda de passagem dentro da tolerância estabelecida. Adicionalmente, notamos uma clara simetria no desempenho: filtros complementares demandam ordens idênticas ou muito próximas para os mesmos níveis de atenuação. Podemos também destacar a notável similaridade de desempenho ao

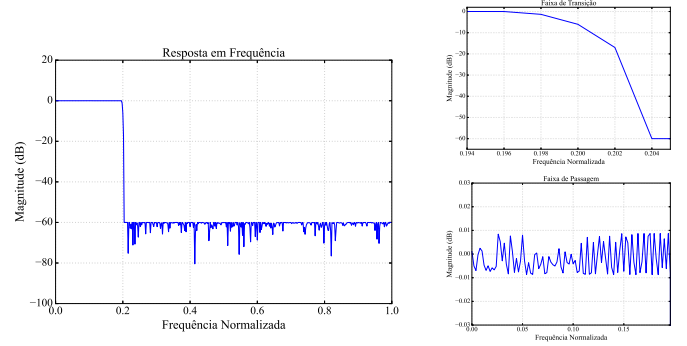


Fig. 2. Resposta em frequência do filtro PB projetado com o método proposto, incluindo detalhes da faixa de transição e de *ripple* na faixa de passagem. O valor mínimo de atenuação na faixa de rejeição é -60 dB. O filtro gerado tem um total de 685 coeficientes.

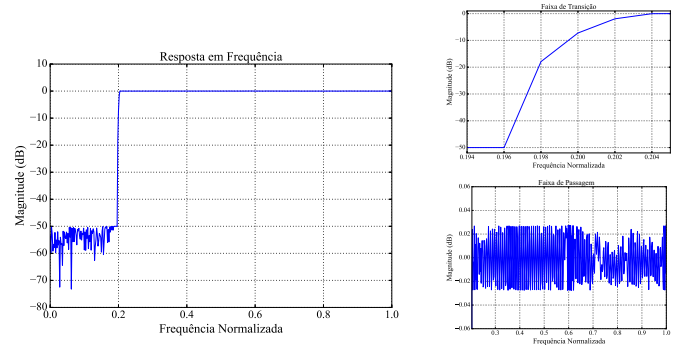


Fig. 3. Resposta em frequência do filtro PA projetado com o método proposto, incluindo detalhes da faixa de transição e de *ripple* na faixa de passagem. O valor mínimo de atenuação na faixa de rejeição é -50 dB. O filtro gerado tem um total de 585 coeficientes.

compararmos os filtros com apenas uma faixa de transição (PB e PA) com aqueles que possuem duas faixas de transição (PF e RF). Embora a exigência de controlar duas regiões de transição tenda a elevar a complexidade do projeto, o método proposto conteve o aumento da ordem a uma diferença marginal, comprovando a eficácia do algoritmo.

Em comparação ao método de janelamento de Kaiser, a superioridade da abordagem proposta é evidente e mais acentuada à medida que as exigências do projeto aumentam. Enquanto o método de Kaiser exige incrementos substanciais na ordem do filtro para satisfazer restrições mais rígidas, o método proposto apresenta um crescimento significativamente mais eficiente. Como exemplo, a Fig. 4 ilustra a resposta do filtro PF a -40 dB, com redução de aproximadamente 15,6% na ordem. Ao transitar para especificações de -60 dB, observamos que essa margem de ganho é amplificada em todos os casos da Tabela I.

Adicionalmente, ao confrontar os resultados com o algoritmo de Parks-McClellan, notamos que o diferencial de desempenho do método proposto é amplificado pelo rigor das restrições de *ripple* e atenuação. Embora o Parks-McClellan seja reconhecido por sua otimalidade no sentido *equiripple*, a estratégia de otimização via rede neural permite encontrar soluções mais compactas conforme impomos especificações

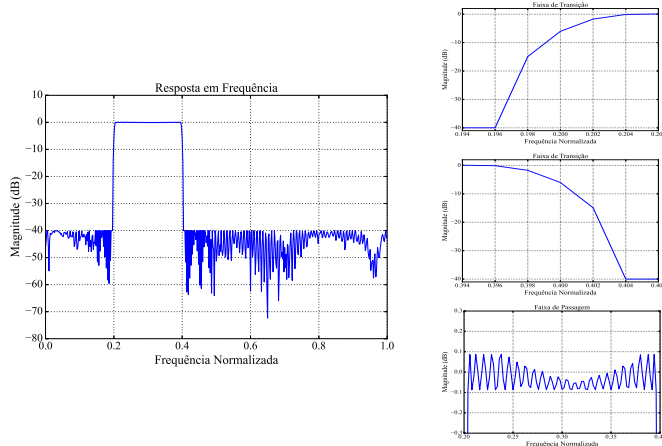


Fig. 4. Resposta em frequência do filtro PF projetado com o método proposto, incluindo detalhes das faixas de transição e de *ripple* na faixa de passagem. O valor mínimo de atenuação das faixas de rejeição é -40 dB. O filtro gerado tem um total de 473 coeficientes.

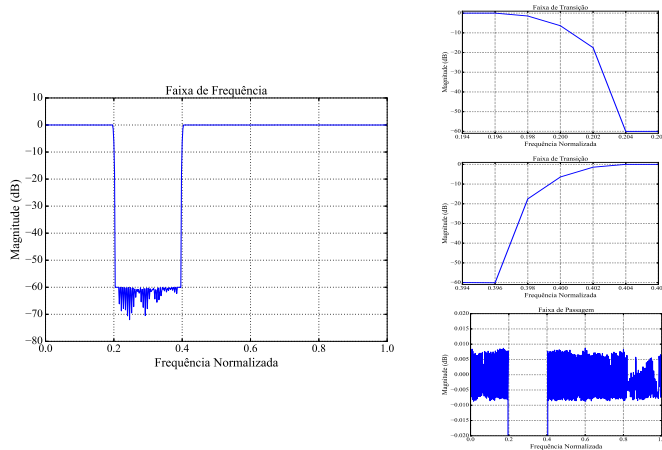


Fig. 5. Resposta em frequência do filtro RF projetado com o método proposto, bem como detalhes das faixas de transição e de *ripple* da faixa de passagem. O valor mínimo de atenuação da faixa de rejeição é -60 dB. O filtro gerado tem um total de 689 coeficientes.

mais rígidas. Na Fig. 5, o filtro RF a -60 dB demonstra uma redução de 16,1% em relação ao Parks-McClellan (689 contra 821 coeficientes). Esse padrão de escalonamento indica que o ajuste dinâmico da envoltória $w_\theta(n)$ explora o espaço de busca de forma mais eficaz, superando as limitações de convergência dos métodos clássicos quando as restrições de projeto impõem desafios maiores à minimização da ordem.

IV. CONCLUSÕES

Este artigo propôs uma arquitetura híbrida baseada em janelamento adaptativo executado por redes neurais artificiais, visando o projeto de filtros FIR com ordem reduzida. Utilizando uma envoltória de ponderação gerada dinamicamente e uma função de perda orientada a especificações, o presente esquema refinou a resposta ao impulso para atender a critérios rigorosos de faixa de transição, *ripple* e atenuação.

Os resultados experimentais obtidos confirmaram a superioridade da abordagem apresentada frente aos métodos de Parks-McClellan e janela de Kaiser, com reduções de até

16,15% e 24,47% no número de coeficientes, respectivamente. Tais resultados implicam ganhos diretos em termos de custo computacional, latência e consumo de energia, tornando esta abordagem especialmente atrativa para aplicações em tempo real e sistemas com restrição de recursos.

REFERÊNCIAS

- [1] N. S. Amer and S. B. Belhaoui, "EEG Signal Processing for Med. Diagnosis, Healthcare, and Monitoring: A Comprehensive Review," *IEEE Access*, v. 11, pp. 143116–143142, 2023.
- [2] A. Alkhateeb, G. Leus and R. W. Heath, "Multi-layer precoding: A potential solution for full-dimensional massive MIMO systems," *IEEE Trans. on Wireless Commun.*, v. 16, pp. 5810–5824, 2017.
- [3] G. Simkus, M. Holters and U. Zölzer, "Error resilience enhancement for a robust ADPCM audio coding scheme," *IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, 2014, pp. 3685–3689.
- [4] A. Said and W. A. Pearlman, "A new, fast, and efficient image codec based on set partitioning in hierarchical trees," *IEEE Trans. on Circuits and Systems for Video Technology*, v. 6, pp. 243–250, 1996.
- [5] A. S. Donald, "Digital Signal Processing for Predicting Stock Prices," *British J. of Computer, Networking and Information Technology*, vol. 4, no. 2, pp. 12–21, 2021.
- [6] T. Singh, P. Agarwal and V. K. Pandey, "ECG baseline noise removal techniques using window based FIR filters," *Int. Conf. on Med. Imaging, m-Health and Emerging Commun. Systems*, 2014, pp. 131–136.
- [7] J. G. Proakis and D. G. Manolakis, *Digital Signal Processing: Principles, Algorithms, and Applications*, 4th ed. Upper Saddle River, NJ, USA: Pearson, 2006.
- [8] N. Yang, M. Usman, X. He, M. A. Jan and L. Zhang, "Time-Frequency Filter Bank: A Simple Approach for Audio and Music Separation," *IEEE Access*, v. 5, pp. 27114–27125, 2017.
- [9] C. Wu, Z. Shi, and V. Govindaraju, "Fingerprint image enhancement method using directional median filter," *Biometric Technology for Human Identification*, v. 5404, pp. 66–75, Aug. 2004.
- [10] M. Singh, S. Bansal, S. Ahuja, R. K. Dubey, B. K. Panigrahi, and N. Dey, "Transfer learning-based ensemble support vector machine model for automated COVID-19 detection using lung computerized tomography scan data," *Med. Biol. Eng. Comput.*, v. 59, no. 4, pp. 825–839, 2021.
- [11] W. Zhao, J. Hu, D. Jia, H. Wang, Z. Li, C. Yan, and J. Fang, "Deep Learning Based Patient-Specific Classification of Arrhythmia on ECG signal," *Inter. Conf. of the IEEE Eng. in Medicine and Biology Society*, 2019, pp. 1500–1503.
- [12] A. C. Barbosa, E. S. Ferreira, V. F. S. B. Grilo, L. Mattos, and J. Lima, "Impact of EMG Signal Filters on Machine Learning Model Training: A Comparison with Clustering on Raw Signal," *Int. Conf. on Optimization, Learning Algorithms and Applications*, 2023, pp. 50–62.
- [13] M. J. Costa, T. M. Rutkowski, and M. T. Reid, "Advances in modern EEG-BCI signal processing: A review," *J. of Neural Eng.*, v. 15, 2018.
- [14] P. K. Sadasivan and D. N. Dutt, "Use of finite wordlength FIR digital filter structures with improved magnitude and phase characteristics for reduction of muscle noise in EEG signals," *Med. and Biological Eng. and Computing*, v. 33, pp. 306–312, 1995.
- [15] P. C. Bhaskara, M. D. Uplane, "High Frequency Electromyogram Noise Removal from Electrocardiogram Using FIR Low Pass Filter Based On FPGA," *Int. J. of Eng. & Technology*, v. 25, pp. 497–504, 2016.
- [16] A. Chandra and S. Chattopadhyay, "Design of hardware efficient FIR filter: A review of the state-of-the-art approaches," *Eng. Science and Technology, an International Journal*, v. 19, pp. 212–226, 2016.
- [17] J. F. Kaiser, "Nonrecursive digital filter design using the I_0 -sinh window function," *IEEE Int. Sym. on Circuits and Systems*, pp. 20–23, 1974.
- [18] T. W. Parks and J. H. McClellan, "A program for the design of linear-phase finite impulse response digital filters," *IEEE Trans. on Audio and Electroacoustics*, pp. 195–199, 1972.
- [19] S. O. Haykin, "Neural networks and learning machines," 3rd ed. Upper Saddle River, NJ, USA: Pearson Education, 2009.
- [20] C. M. Bishop, *Pattern Recognition and Machine Learning*. New York: Springer, 2006.
- [21] D. P. Kingma, and J. Ba, "Adam: A Method for Stochastic Optimization," *Proc. of the Int. Conference on Learning Representations*, 2015.
- [22] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, 1986.