

# Aplicação de Filtros Digitais em Tempo Real com o PYDAQ

Vinícius Silvestre, Cayo Castoril, Samir Martins \*

*\* GCoM - Grupo de Controle e Modelagem, Departamento de Engenharia Elétrica, Universidade Federal de São João del Rei, MG  
E-mails: vitinhoviničius8@gmail.com, cayorwcs@gmail.com, martins@ufsj.edu.br*

**Abstract:** This work presents the implementation of real-time digital filters in the PYDAQ library. PYDAQ, which currently averages 556 downloads per month, is an open-source Python-based tool designed for data acquisition and experimental analysis. This study focuses on the design and implementation of FIR and IIR (Finite/Infinite Impulse Response) filters in PYDAQ, enabling noise reduction and expanding the library's existing functionalities. The research combines theoretical investigation and practical implementation, using devices such as Arduino and National Instruments data acquisition boards for real-time applications. The presented results demonstrate that the developed tool can be a valuable asset in practices involving active learning methodologies and research. Moreover, it can be used for laboratory virtualization—a growing demand in hybrid engineering education—as evidenced by the executed tests.

**Resumo:** Neste trabalho é apresentada a implementação de filtros digitais em tempo real na biblioteca PYDAQ. O PYDAQ (Martins, 2023), que atualmente conta com uma média de 556 downloads por mês, é uma ferramenta de código aberto, baseada em Python, voltada para aquisição de dados e análise experimental. Este trabalho foca no projeto e implementação de filtros FIR e IIR (Finite/Infinite Impulsive Response) no PYDAQ, permitindo a redução de ruídos e ampliando as várias funcionalidades já existentes no PYDAQ. A pesquisa combina investigação teórica e implementação prática, utilizando dispositivos como Arduino e placas de aquisição de dados da National Instruments para aplicações em tempo real. Os resultados apresentados comprovam que a ferramenta desenvolvida pode ser um importante aliado em práticas envolvendo metodologias ativas de ensino e pesquisa, podendo inclusive ser utilizada como virtualização de laboratórios, demanda crescente para ensinos de engenharia em formato híbrido, como os testes executados puderam demonstrar.

**Keywords:** Digital Filters; PYDAQ; IIR; FIR; Signal processing.

**Palavras-chaves:** Filtros Digitais; PYDAQ; Processamento de sinais; IIR; FIR.

## 1. INTRODUÇÃO

O desenvolvimento dos filtros digitais foi impulsionado pelo avanço da eletrônica digital e pela crescente demanda por técnicas eficazes de processamento de sinais. A popularização dos computadores digitais na década de 1960 revolucionou a análise e o tratamento de sinais, permitindo a implementação de algoritmos mais sofisticados de filtragem (Lyons, 2011).

A filtragem digital é essencial para modificar o espectro do sinal, permitindo a passagem de determinadas frequências enquanto atenua outras, sendo amplamente utilizada em áreas como telecomunicações, processamento de imagens médicas e controle de sistemas (Rozaqi et al., 2019; Çelebi, 2020; Patel and Shah, 2021; Rosli et al., 2018; Aslankaya et al., 2024). Diferentemente dos filtros analógicos, que operam em sinais contínuos, os filtros digitais processam sequências discretas de valores, proporcionando maior flexibilidade e precisão (Zhang et al., 2014). O avanço da computação e o acesso a ferramentas de código aberto,

como Python, têm facilitado a implementação dessas técnicas.

Neste contexto, o PYDAQ (Martins, 2023) surge como uma solução acessível para aquisição e processamento de sinais em tempo real, integrando bibliotecas como SysIdentPy para análise avançada. Este trabalho propõe a expansão do PYDAQ com a implementação de filtros digitais FIR (Finite Impulse Response) e IIR (Infinite Impulse Response), permitindo que usuários apliquem filtragem de sinais em tempo real por meio de uma interface gráfica intuitiva desenvolvida com PySide6.

Além da contribuição técnica, essa funcionalidade fortalece o uso do PYDAQ como ferramenta educacional, promovendo metodologias ativas de ensino, como a Aprendizagem Baseada em Projetos (PBL) (Wang and Nyamapfene, 2022). A acessibilidade da ferramenta permite que estudantes experimentem conceitos de processamento de sinais de forma prática e interativa, ampliando o acesso a tecnologias avançadas no ensino de Engenharia.

Este artigo está organizado da seguinte forma: a Seção 2 apresenta os conceitos preliminares necessários para o desenvolvimento do trabalho. A Seção 3 descreve a metodologia adotada, enquanto a Seção 4 apresenta os resultados obtidos. Por fim, a Seção 5 contém as conclusões e sugestões para futuros desenvolvimentos.

## 2. CONCEITOS PRELIMINARES

Nesta seção, descreve-se os conceitos preliminares básicos para a implementação da funcionalidade de filtragem de sinais em tempo real no PYDAQ.

### 2.1 Filtros Digitais

Os filtros digitais são classificados em duas grandes categorias: FIR (Finite Impulse Response) e IIR (Infinite Impulse Response) (Patel and Shah, 2021). Os filtros FIR, cuja resposta ao impulso tem duração finita, ganharam popularidade devido à sua estabilidade incondicional e facilidade de implementação em hardware e software. O projeto desses filtros pode ser realizado por meio do método de janela, técnica introduzida para simplificar a síntese de respostas em frequência. A equação geral de um filtro FIR é representada por:

$$y[n] = \sum_{k=0}^{M-1} b_k x[n-k] \quad (1)$$

em que  $y[n]$  representa a saída do filtro,  $x[n]$  é a entrada,  $n$  é o número da amostra realizada,  $b_k$  são os coeficientes do filtro e  $M$  corresponde à ordem do filtro.

A equação geral do filtro FIR também é escrita com a transformada Z, e é dada por:

$$H[z] = \sum_{k=0}^{M-1} b_k z^{-k} \quad (2)$$

onde  $H[z]$  representa a função de transferência do filtro,  $b_k$  são os coeficientes do filtro e  $z^{-k}$  representa o deslocamento no tempo da entrada do filtro, onde  $k$  é o índice dos coeficientes.

Para o projeto de um filtro FIR, podem ser aplicadas diversas funções de janela, incluindo Hann, Hamming, Bartlett, Blackman, Gaussian e Kaiser (He et al., 2017), cada uma com características específicas em termos de atenuação da banda de transição e efeito na resposta em frequência.

Os filtros IIR, cuja resposta ao impulso pode se estender indefinidamente, podem ser desenvolvidos a partir de técnicas de modelagem analógica-digital, permitindo a implementação de filtros com alta seletividade e menor ordem em comparação aos FIR, (Lyons, 2011), ao custo de não linearidade de fase. A equação geral de um filtro IIR pode ser expressa como:

$$y[n] = \sum_{k=0}^M b_k x[n-k] - \sum_{j=1}^N a_j y[n-j] \quad (3)$$

onde  $y[n]$  é a saída do filtro no instante  $n$ ,  $x[n-k]$  são os valores de entrada do filtro nos instantes  $n-k$  para  $k = 0, 1, \dots, M-1$ ,  $y[n-k]$  os valores anteriores da saída do filtro nos instantes  $n-k$  para  $k = 1, 2, \dots, N$ ,  $a_j$  e  $b_k$  são os coeficientes do filtro,  $M$  é a ordem do filtro para a entrada e  $N$  a ordem do filtro para a saída.

No domínio Z, um filtro IIR pode ser escrito como:

$$H(z) = \frac{\sum_{k=0}^{M-1} b_k z^{-k}}{1 + \sum_{k=1}^N a_k z^{-k}} \quad (4)$$

onde  $H(z)$  é a função de transferência do filtro IIR,  $b_k$  os coeficientes do numerador,  $a_k$  os coeficientes do denominador,  $M$  e  $N$  são as ordens do filtro para a entrada e saída respectivamente.

Entre as principais abordagens para o projeto de filtros IIR, destacam-se as topologias clássicas: Butterworth, Chebyshev Tipo I, Chebyshev Tipo II e Elíptico. Essas configurações foram selecionadas por oferecerem diferentes compromissos entre linearidade de fase, seletividade e eficiência computacional, garantindo flexibilidade na escolha do filtro mais adequado para cada aplicação.

### 2.2 PYDAQ

A biblioteca PYDAQ (Martins, 2023) foi desenvolvida para fornecer uma interface gráfica simples, intuitiva e gratuita para a aquisição e processamento de sinais em tempo real utilizando placas Arduino e NIDAQ, National Instruments Data Acquisition. Seu desenvolvimento surgiu da necessidade de uma solução acessível para o processamento de sinais digitais, fundamental em diversas aplicações em engenharia, ensino e pesquisa. Inicialmente, a biblioteca oferecia funcionalidades básicas, como aquisição e envio de dados em tempo real por meio de técnicas de Identificação de Sistemas usando modelos NARX polinomiais, além de testes de resposta a degrau. Com o tempo, novas capacidades foram incorporadas, incluindo a obtenção de modelos matemáticos em tempo real (Junior et al., 2020). O PYDAQ se consolidou como uma ferramenta relevante na comunidade científica e acadêmica, e atualmente conta com uma média de 463 *downloads* mensais de acordo com os dados acessados no repositório no dia 08 de junho de 2025. Além de suas contribuições para o avanço técnico, o PYDAQ se destaca como uma poderosa ferramenta de ensino, alinhada às práticas de metodologias ativas, como a Aprendizagem Baseada em Projetos (PBL). Sua interface acessível e interativa possibilita que estudantes de Engenharia Elétrica experimentem conceitos de aquisição e processamento de sinais de forma prática e intuitiva, promovendo um aprendizado mais dinâmico e alinhado às necessidades do mercado. Dessa forma, o PYDAQ não apenas facilita a compreensão de conceitos teóricos, mas também prepara os alunos para enfrentar desafios reais por meio de experiências *hands-on*, essenciais no contexto educacional contemporâneo.

### 2.3 PySide6

No contexto do ensino de Engenharia, a utilização do PySide6 possibilita a criação de interfaces interativas que

promovem a aprendizagem prática, alinhada às metodologias ativas. A construção de ferramentas didáticas que integram teoria e experimentação permite que os estudantes explorem conceitos fundamentais de processamento de sinais de forma intuitiva e aplicada.

Para facilitar o design das interfaces, o Qt Designer, uma ferramenta gráfica integrada ao Qt Creator, pode ser utilizado para a criação visual das telas, gerando arquivos .ui. Esses arquivos podem ser facilmente integrados ao código Python, garantindo uma separação clara entre a lógica da aplicação e a interface gráfica. Essa modularidade torna o desenvolvimento mais acessível para estudantes e pesquisadores interessados na construção de ferramentas educacionais interativas.

Dessa forma, o uso do *PySide6* em um ambiente de ensino pode proporcionar aos alunos a possibilidade de testar e visualizar, de forma prática, os efeitos da filtragem digital em sinais reais. A interação com esses conceitos em um ambiente experimental facilita a assimilação dos princípios teóricos e estimula um aprendizado mais ativo e exploratório, essencial para a formação de engenheiros capacitados para enfrentar desafios reais.

### 3. METODOLOGIA

Este trabalho teve como objetivo a implementação de filtros digitais em tempo real utilizando a biblioteca PYDAQ, visando ao desenvolvimento de uma ferramenta de ensino para a virtualização de laboratórios e a aplicação de metodologias ativas na educação em engenharia. Para isso, foram projetados e integrados filtros digitais do tipo FIR (Finite Impulse Response) e IIR (Infinite Impulse Response), proporcionando um ambiente interativo para aquisição e processamento de sinais, sem a necessidade de conhecimentos avançados em programação. Além disso, a interface gráfica foi desenvolvida com *PySide6*, permitindo a manipulação intuitiva dos parâmetros dos filtros.

#### 3.1 Desenvolvimento da interface gráfica

Para garantir uma experiência interativa e acessível ao usuário, foi projetada uma interface gráfica utilizando *PySide6*, com o auxílio do Qt Designer. A interface permite a configuração intuitiva dos filtros digitais, possibilitando a seleção de parâmetros como tipo de filtro, frequência de corte, ordem e função de janela. Além disso, a ferramenta fornece uma visualização em tempo real do sinal filtrado, tornando o processo mais didático e acessível para usuários com diferentes níveis de conhecimento técnico.

#### 3.2 Implementação dos Filtros Digitais

Após o desenvolvimento da interface gráfica e sua integração ao PYDAQ, foi necessário implementar a funcionalidade de filtragem digital. Para isso, utilizou-se a biblioteca *SciPy*, amplamente adotada na comunidade científica e acadêmica devido à sua facilidade de uso, desempenho otimizado e integração com outras bibliotecas do ecossistema Python. No contexto do processamento de sinais, *SciPy* disponibiliza o módulo *scipy.signal*, que permite a implementação de filtros digitais FIR e IIR, análise espectral, transformadas de Fourier e processamento de sinais em tempo discreto.

Antes da escolha das topologias para os filtros FIR e IIR, foi realizada uma simulação utilizando dados ruidosos com o objetivo de testar diferentes métodos de projeto e avaliar sua eficiência na atenuação do ruído. Esse procedimento permitiu comparar as respostas dos filtros em diferentes condições e selecionar as abordagens mais adequadas para a implementação em tempo real.

Para o projeto de filtros FIR, utilizou-se a função *firwin*, baseada no método da janela, permitindo a definição de diferentes funções janela para controlar a resposta em frequência, entre elas: Hamming, Hann, Gauss, Kaiser, Blackman e Bartlett-Hann. Já para os filtros IIR, foram empregadas as funções *butter*, *cheby1*, *cheby2* e *ellip*, que possibilitam a síntese de filtros Butterworth, Chebyshev de Tipo I e II e Elíptico, respectivamente.

Para viabilizar a implementação dos filtros em tempo real no PYDAQ, utilizou-se como base a função *Get Data*, já disponível na biblioteca. Essa função permite que o usuário colete um sinal em tempo real utilizando placas de aquisição, como Arduino ou NIDAQ. A modificação proposta neste trabalho adiciona uma nova funcionalidade, permitindo que o usuário escolha entre adquirir os dados com ou sem filtragem digital. Caso a filtragem seja ativada, será possível selecionar entre filtros FIR e IIR, além de ajustar suas características, além de uma opção para exibir a resposta em frequência do sinal, caso o usuário opte. Assim que os parâmetros são definidos, o projeto do filtro é salvo e aplicado ao sinal adquirido em tempo real pela função *Get Data*, além de que é gerado um arquivo “.dat” com os valores dos coeficientes “b” e “a” para o filtro IIR, e os valores do coeficiente “b” para o filtro FIR. A Figura 1 apresenta um fluxograma com o passo a passo da implementação.

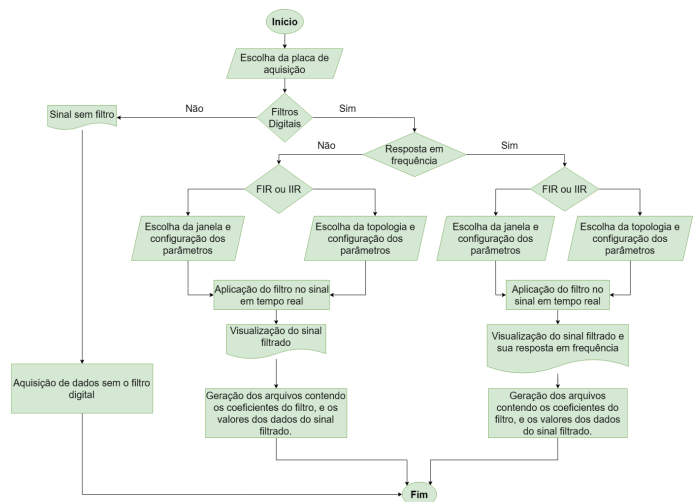


Figura 1. Fluxograma da aplicação dos filtros digitais no sinal.

## 4. RESULTADOS

### 4.1 Interface Gráfica para Configuração dos Filtros Digitais

A interface gráfica desenvolvida possibilita ao usuário configurar e selecionar os parâmetros do filtro digital a ser aplicado ao sinal adquirido em tempo real. A interface conta com uma caixa de seleção para a escolha entre filtros

FIR (*Finite Impulse Response*) e IIR (*Infinite Impulse Response*), além de um controle para definir se a resposta em frequência do filtro projetado será exibida ao final da aquisição do sinal.

Para a síntese dos filtros FIR, a interface permite a configuração de diversos parâmetros essenciais ao projeto do filtro digital. A Figura 2 ilustra a interface correspondente a essa configuração.

- Função janela (*windowing method*): Hamming, Hann, Gauss, Kaiser, Blackman ou Bartlett-Hann (Anshul and Rathi, 2017), utilizadas para minimizar o efeito de truncamento na resposta em frequência.
- Ordem do filtro ( $N$ ), definida pelo usuário com base no compromisso entre atenuação da banda de transição e distorção de fase.
- Tipo de filtro: passa-baixa, passa-alta, passa-faixa ou rejeita-faixa.
- Frequência de corte ( $f_c$ ), normalizada pela frequência de amostragem ( $f_s$ ), definida pelo inverso do período de amostragem, na tela inicial do próprio PYDAQ.

No caso dos filtros IIR, a interface permite a definição de parâmetros essenciais para o projeto do filtro, abrangendo desde a escolha da topologia até ajustes finos de ripple e ordem do sistema. A Figura 2 apresenta a interface dedicada a essa configuração.

- Frequência de corte ( $f_c$ ), parâmetro essencial para determinar a faixa de atuação do filtro.
- Tipo de filtro: passa-baixa, passa-alta, passa-faixa ou rejeita-faixa.
- Topologia do filtro: Butterworth, Chebyshev Tipo I, Chebyshev Tipo II ou Elíptico (Lyons, 2011).
- Ripple da banda passante ( $\delta_p$ ) e da banda de rejeição ( $\delta_s$ ), relevantes para filtros Chebyshev e Elíptico, permitindo o controle do comprometimento entre distorção na banda passante e atenuação na banda de rejeição.
- Ordem do filtro ( $N$ ), ajustável pelo usuário para otimizar a resposta em frequência de acordo com as restrições do sistema.

A interação com a interface permite que os coeficientes do filtro sejam gerados e aplicados ao sinal adquirido em tempo real. A exibição da resposta em frequência do filtro projetado é opcional e pode ser utilizada para validação visual da resposta espectral antes da aplicação ao sinal de entrada.

Este trabalho propõe uma ferramenta para a virtualização de laboratórios de ensino, possibilitando a coleta e o processamento de sinais em tempo real sem a necessidade de hardware especializado. Para demonstrar essa capacidade, utilizou-se uma placa de aquisição de dados National Instruments NI USB-6008, emulada pelo software NI-MAX, um software proprietário gratuito. A emulação permite a geração de sinais e a adição de ruído, criando um ambiente controlado para análise e experimentação, na ausência de uma placa arduino ou nidaq física.

Com essa abordagem, é possível projetar e analisar filtros digitais diretamente em um computador, tornando viável a realização de experimentos didáticos em um laboratório caseiro. Como exemplo, foi implementado um filtro IIR, do

Figura 2. Interface das janelas para o projeto dos filtros digitais FIR e IIR.

tipo Chebyshev II, configuração passa-baixa, ripple de 1dB na banda de rejeição e frequência de corte de 0,1 Hz. A resposta do sistema filtrado pode ser observada na Figura 3 e sua resposta em frequência na Figura 4.

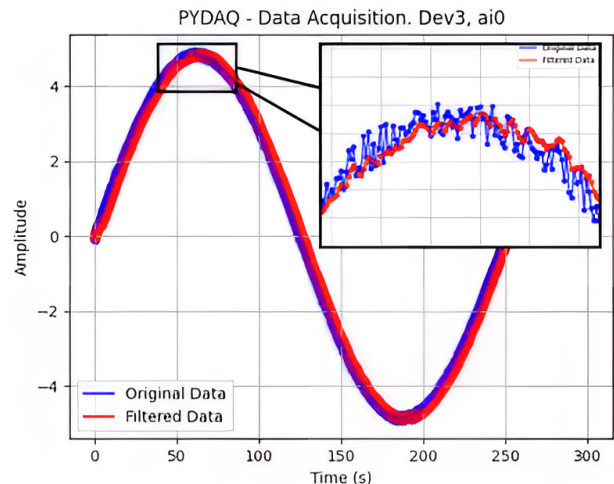


Figura 3. Resultado da simulação.

Após a aquisição dos dados em tempo real com o sinal simulado, verificou-se que a implementação da filtragem digital opera conforme esperado. Esse experimento demonstra como a virtualização de laboratórios possibilita a análise e o projeto de filtros digitais sem a necessidade de hardware especializado, tornando o processo acessível para ensino e pesquisa. Como ilustrado na Figura 3 e 4, o sinal filtrado apresenta uma defasagem de fase, um efeito característico associado à ordem do filtro utilizado. Além disso, observa-se uma atenuação significativa dos ruídos presentes no sinal original, confirmando a eficácia do método empregado. Essa abordagem permite que estudantes e profissionais explorem conceitos de processamento de sinais de forma interativa e replicável, fortalecendo o

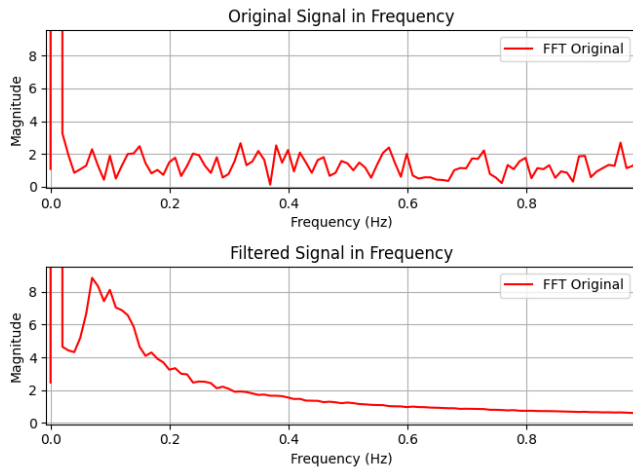


Figura 4. FFT Filtered.

aprendizado prático em ambientes acadêmicos e laboratoriais remotos.

#### 4.2 Teste em Laboratório

A fim de validar o funcionamento do sistema de aquisição e filtragem em tempo real, os testes foram realizados no Laboratório de Controle e Instrumentação (LACOI), localizado no departamento de engenharia elétrica da Universidade Federal de São João del Rei, utilizando um Módulo Universal 2000 Datapool® e um sensor de corrente por efeito Hall CT04. Esse sensor foi escolhido devido à sua pior relação sinal/ruído, sendo a saída em tensão significativamente ruidosa, mesmo mantendo o mensurando (corrente elétrica) fixo. Essa característica torna o CT04 ideal para avaliar a eficácia da filtragem em tempo real, pois simula um cenário desafiador com altos níveis de ruído (Figura 5).

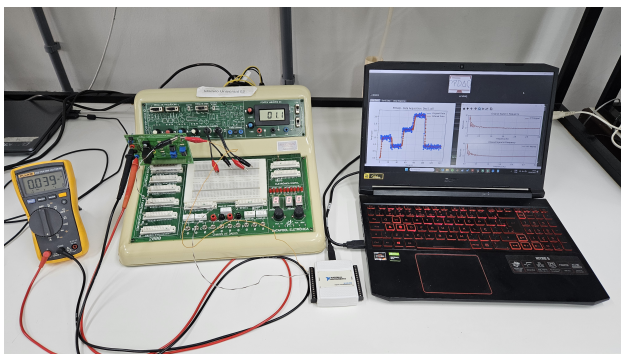


Figura 5. Bancada montada com a placa de aquisição de sinais e o módulo.

Para a realização desse teste, foi projetado um filtro FIR janela de Blackman, ordem 21, passa-baixa e uma frequência de corte de 1Hz (Figura 6).

Com o filtro devidamente configurado, iniciou-se a aquisição de dados variando a tensão aplicada ao módulo de corrente, estabelecendo diferentes patamares para facilitar a análise posterior. Durante os testes, observou-se uma defasagem no sinal filtrado, decorrente do atraso de fase do filtro, um efeito já esperado devido à ordem e o filtro escolhido no projeto. No entanto, verificou-se que a filtragem

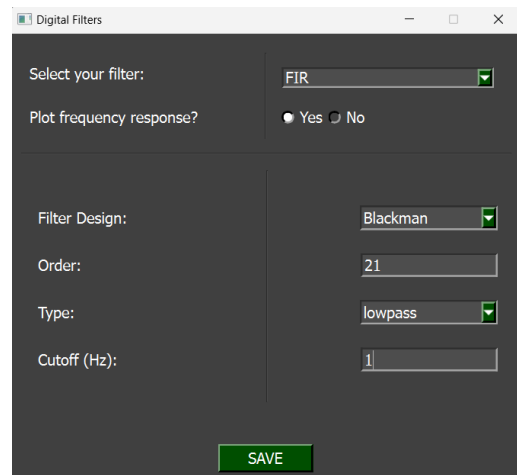


Figura 6. Captura da janela da configuração do filtro FIR para a realização do teste em tempo real em um sistema.

em tempo real funcionou conforme o esperado, sem perdas significativas na amplitude do sinal. Além disso, constatou-se uma atenuação eficaz do ruído, resultando em um sinal consideravelmente mais limpo (Figura 7 e 8).

Para avaliar o desempenho do filtro, foi gerada uma janela exibindo a Transformada Rápida de Fourier (FFT) do sinal original e do sinal filtrado. Como esperado, o filtro apresentou o comportamento desejado, atenuando a passagem das frequências a partir de 1Hz (Figura 8).

Por fim, quatro arquivos de saída são gerados para o usuário:

- data.dat: contém os valores do sinal original adquirido.
- datafiltered.dat: armazena os valores do sinal filtrado, possibilitando futuras análises ou visualizações.
- time.dat: registra os instantes de tempo correspondentes à aquisição e filtragem dos dados.
- filtercoeffs.dat: contém os coeficientes do filtro FIR, especificamente os valores do vetor  $b$ .

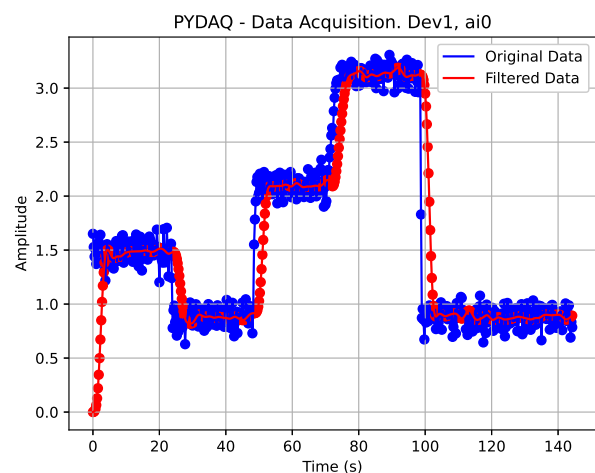


Figura 7. Resultado da filtragem.



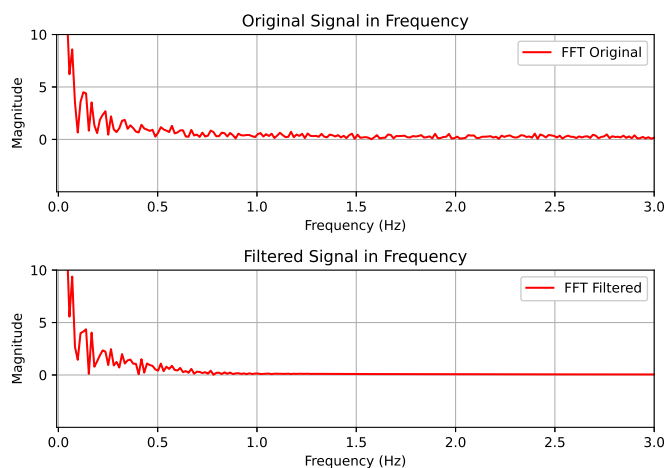


Figura 8. Janela da comparação entre as FFT's.

## 5. CONCLUSÃO

Este trabalho apresentou a implementação de filtros digitais em tempo real utilizando a biblioteca PYDAQ, permitindo a filtragem de sinais adquiridos por placas como Arduino e NIDAQ. A interface gráfica foi desenvolvida com PySide6, proporcionando uma interação intuitiva e fácil configuração dos filtros FIR e IIR. Utilizou-se a biblioteca SciPy para o design e aplicação dos filtros, com suporte a diferentes topologias, como FIR com *firwin* e IIR com *butter*, *cheby1*, *cheby2* e *ellip*.

Simulações e testes realizados no laboratório de Estudos de Controle e Modelagem, LECOM, localizado na Universidade Federal de São João del Rei, comprovaram a eficácia da filtragem digital em tempo real, permitindo a remoção de ruídos e os ajustes do sinal conforme os parâmetros definidos pelo usuário. Além disso, a abordagem adotada demonstra o potencial de ferramentas computacionais na modernização do ensino de engenharia, possibilitando a experimentação interativa de conceitos complexos.

Diante do avanço das tecnologias de simulação e aquisição de dados, a virtualização de laboratórios surge como uma solução essencial para ampliar o acesso a práticas experimentais, reduzindo custos e tornando o aprendizado mais inclusivo. O sistema desenvolvido neste trabalho se alinha a essa tendência, oferecendo uma plataforma acessível para a aplicação de filtros digitais sem a necessidade de infraestrutura laboratorial complexa.

Para trabalhos futuros, considera-se a expansão do sistema para suportar uma maior variedade de filtros, a implementação de técnicas avançadas de filtragem e a otimização do desempenho.

O código-fonte deste trabalho encontra-se disponível no repositório do GitHub: <https://github.com/viniciusfs14>.

## AGRADECIMENTOS

Os autores agradecem à FAPEMIG, CAPES, UFSJ e CNPq pelo apoio financeiro na realização deste estudo e na sua submissão ao SBAI 2025.

## REFERÊNCIAS

- Anshul and Rathi, K. (2017). Comparison of various window techniques for design fir digital filter. In *2017 IEEE International Conference on Power, Control, Signals and Instrumentation Engineering (ICPCSI)*, 428–432. doi:10.1109/ICPCSI.2017.8392331.
- Aslankaya, E., Yilmaz, A., and Bayrak, G. (2024). Enhancing power quality in vehicle-to-grid (v2g) operations of fcevs through the integration of real-time digital iir filters in power calculations. *International Journal of Hydrogen Energy*. doi:<https://doi.org/10.1016/j.ijhydene.2023.12.205>.
- He, P., Chang, H., Gao, H., and Wang, Z. (2017). Design of iir digital filter. In *2017 6th International Conference on Computer Science and Network Technology (ICCSNT)*, 506–507. doi:10.1109/ICCSNT.2017.8343749.
- Junior, W.R.L., da Andrade, L.P.C., Oliveira, S.C.P., and Martins, S.A.M. (2020). Sysidentpy: A python package for system identification using narmax models. *Journal of Open Source Software*, 5(54), 2384. doi:10.21105/joss.02384.
- Lyons, R. (2011). *Understanding Digital Signal Processing*. Prentice Hall, 3<sup>a</sup> ed. edition.
- Martins, S.A.M. (2023). Pydaq: Data acquisition and experimental analysis with python. *Journal of Open Source Software*, 8(92), 5662. doi:10.21105/joss.05662. URL <https://doi.org/10.21105/joss.05662>.
- Patel, V. and Shah, A. (2021). Digital multiband filter design with power spectrum analysis for electrocardiogram signals. In *2021 International Conference on Recent Trends on Electronics, Information, Communication Technology (RTEICT)*, 923–927. doi:10.1109/RTEICT52294.2021.9573714.
- Rosli, R.S., Habaebi, M.H., and Islam, M.R. (2018). Comparative analysis of digital filters for received signal strength indicator. In *2018 7th International Conference on Computer and Communication Engineering (ICCCE)*, 214–217. doi:10.1109/ICCCE.2018.8539273.
- Rozaqi, L., Nugroho, A., Sanjaya, K.H., and Ivonita Simbolon, A. (2019). Design of analog and digital filter of electromyography. In *2019 International Conference on Sustainable Energy Engineering and Application (ICSEEA)*, 186–192. doi:10.1109/ICSEEA47812.2019.8938645.
- Wang, L. and Nyamapfene, A. (2022). Student experiences of pbl during remote learning: A case study. In *2022 IEEE Global Engineering Education Conference (EDUCON)*, 1822–1823. doi:10.1109/EDUCON52537.2022.9766776.
- Zhang, L., Lai, X., and Qiao, H. (2014). Design of allpass iir digital filters using a minimax frequency response error method. In *Proceedings of the 33rd Chinese Control Conference*, 7328–7332. doi:10.1109/ChiCC.2014.6896215.
- Çelebi, M. (2020). Digital filter design based on arduino and its applications. In *2020 Medical Technologies Congress (TIPTEKNO)*, 1–4. doi:10.1109/TIPTEKNO50054.2020.9299311.