

Simulador de filtros para análise de sinais

Laboratório de Instrumentação Elétrica – PSI3214
Rodrigo Anjos de Souza, Elisabete Galeazzo

17/09/20

Simulador de Filtros para Análise de Sinais

Objetivos

1. Familiarização com projeto de filtros determinísticos em LabVIEW
2. Apresentação de ferramentas de análise espectral de ondas em LabVIEW

Sumário

- 1 Apresentação de Análise Espectral em Ondas
 - 1.1 Representação de Sinais no Domínio da Frequência
 - 1.2 Exemplos
- 2 Análise de Arquitetura
 - 2.1 Análise do Executável
 - 2.2 Inicialização
 - 2.3 Processamento
 - 2.4 Finalização
- 3 Filtros
 - 3.1 Intuição Geral de um Filtro
 - 3.1.1 Projeto Equalizador – PSI3212
 - 3.2 Apresentação visual de um filtro
 - 3.3 Aplicação em um sinal
- 4 FFT
 - 4.1 Aplicação do Filtro em FFT (Comparação)
 - 4.2 Frequências dominantes
- 5 Conclusão do Programa

1 Apresentação de Análise Espectral de Sinais – uma introdução

Como apresentado em aula e verificado através das atividades experimentais, a *Série de Fourier* é uma ferramenta matemática muito utilizada para a caracterização de sinais periódicos. Foi visto que, através desta série, um sinal periódico qualquer pode ser representado pela soma de senos, com determinados valores de frequência, magnitude e fase. Neste tutorial veremos, de maneira empírica, **como a representação de um sinal elétrico periódico qualquer, por meio de séries, pode ser útil para melhorar a compreensão de suas características.**

1.1 Representação de Sinais no Domínio da Frequência

Como descrito, *Séries de Fourier* são aplicadas para caracterizar sinais periódicos e representá-los no domínio da frequência. Contudo, na maioria das situações reais, nos deparamos com sinais que não possuem tal característica, ou seja, são aperiódicos. Aqui então nos deparamos com a primeira ressalva deste documento: para realizar a representação de sinais periódicos no domínio da frequência fazemos uso de *Séries de Fourier*, ao passo que a *Transformada de Fourier* é utilizada para determinar todas as frequências associadas a sinais aperiódicos. A *Transformada de Fourier* é uma ferramenta matemática mais abrangente, que relaciona qualquer função integrável nos domínios do tempo e da frequência, possibilitando determinar todas as frequências associadas ao sinal em análise.

Neste tutorial, temos o propósito de aplicar essas ferramentas matemáticas para a análise de sinais, que são muito importantes e bastante utilizadas ao longo do curso de graduação, no entanto alertamos que todos os conceitos são apresentados aqui de forma bem simplificada, pois serão vistos com mais calma e profundidade em disciplinas mais avançadas do curso de Engenharia Elétrica. Para fins deste tutorial, no entanto, basta que seja criada uma intuição sobre o conceito de representação do sinal no domínio da frequência: **todo e qualquer sinal possui quantidades determináveis de energia, dadas pelas amplitudes dos senos que o descreve, e em certas frequências.**

1.2 Exemplos

Como exemplo da abstração descrita acima, tome um sinal senoidal, como na Fig. 1.1. Veja que o sinal é um seno puro, com frequência de 10 Hz e amplitude 1 V_p. Se utilizarmos a teoria dada em aula, é trivial chegarmos à conclusão de que sua representação no domínio da frequência será uma raia de amplitude 1 e frequência de 10 Hz, como ilustrado na Fig. 1.2.

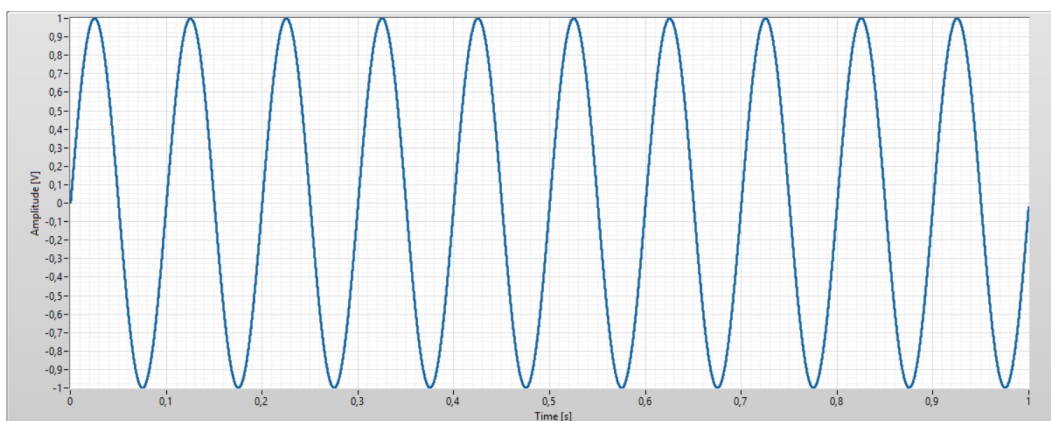


Fig. 1.1 – Sinal senoidal com frequência 10 Hz.

Simulador de filtros para análise de sinais

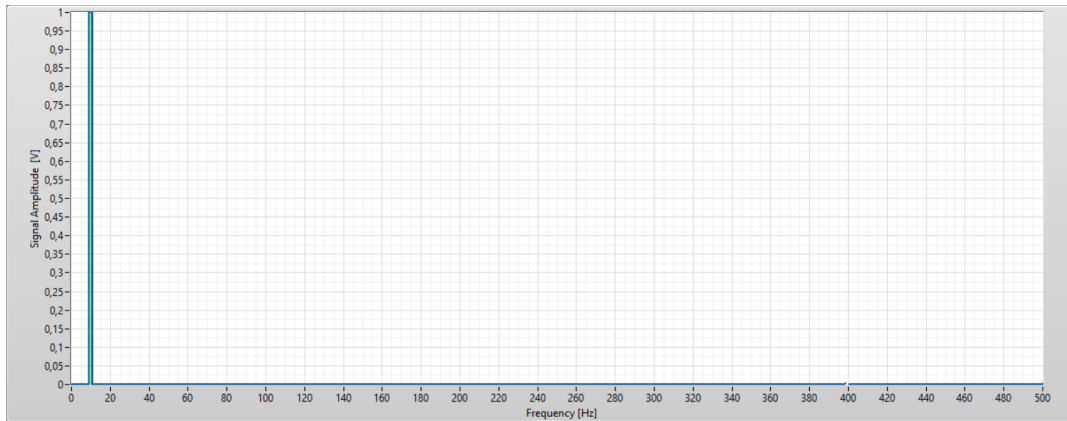


Fig. 1.2 – Representação pela Série de Fourier .

O mesmo conceito pode ser aplicado a sinais um pouco mais complexos e menos intuitivos, como o sinal mostrado na Fig. 1.3.

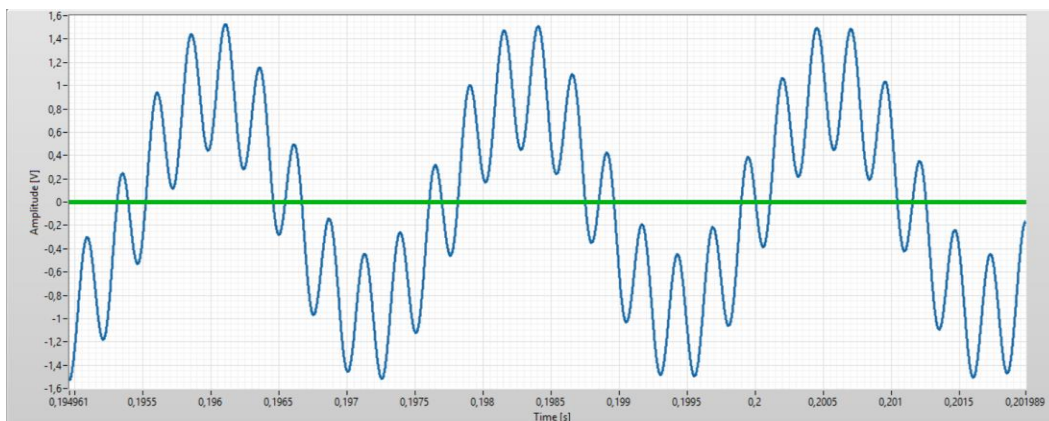


Fig. 1.3 – Sinal composto pela soma de 2 sinais senoidais de frequências distintas.

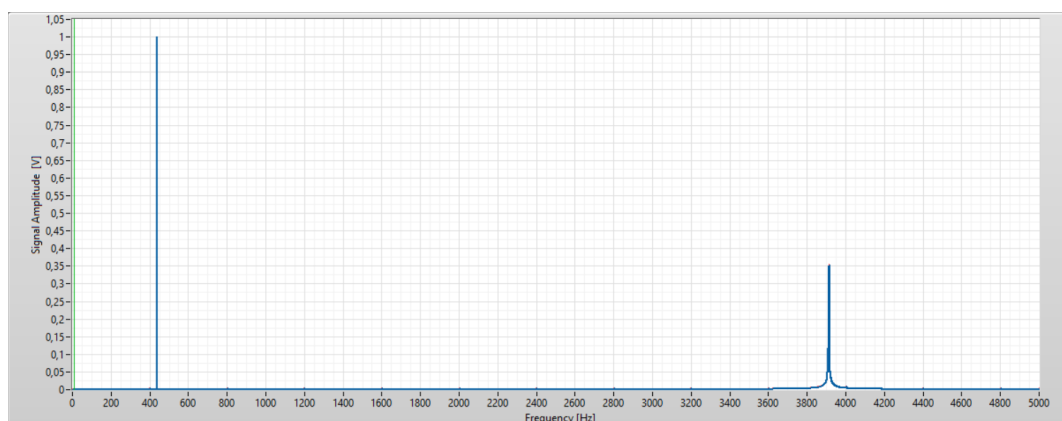


Fig. 1.4 – Representação Espectral da soma de dois sinais senoidais.

Note que a utilização da representação do sinal no domínio da frequência, Fig. 1.4, deixa óbvia a natureza do sinal apresentado na Fig. 1.3. Apesar do exemplo escolhido ser relativamente simples de ser compreendido mesmo no domínio do tempo, para outros sinais veremos que a utilização da representação no domínio da frequência é imprescindível.

2 Análise de Arquitetura

Feita uma breve introdução ao assunto de análise espectral, e destacada a importância de se realizar a análise do sinal tanto no domínio do tempo quanto no da frequência, vamos agora entender a funcionalidade do VI que será construído neste tutorial.

2.1 Análise do Executável

Os objetivos deste tutorial são: *criar uma intuição* sobre as características de um sinal por meio da série de Fourier, bem como compreender, de forma mais detalhada, como *realizar um projeto de filtro* em LabVIEW.

Abra o projeto [**Tutorial 3.lvproj**], e abra o VI principal: [**tut3_main.vi**]. Perceba que o VI é semelhante ao utilizado no *Tutorial sobre Conversão AD*. Neste VI, no lado esquerdo do painel frontal temos os blocos onde são visualizados os sinais gerados, e no lado direito estão os controles dos parâmetros dos sinais e os indicadores.

Execute o VI. Programe o “signal generation” para um sinal senoidal de 440 Hz.

Para visualizar o mesmo sinal no domínio da frequência, clique no menu “FFT View – [Frequency Domain]” (situado acima do gráfico amplificado) (Fig 2.1).



Fig. 2.1 – Menu de mudança de Visualização.

Podemos ver com mais clareza as frequências que compõe o sinal por meio deste gráfico.

Assim como fizemos no tutorial 1, utilize as ferramentas da paleta do gráfico para aplicar zoom ao gráfico principal, e utilize os cursores para identificar com precisão qual a frequência gerada e qual sua amplitude.

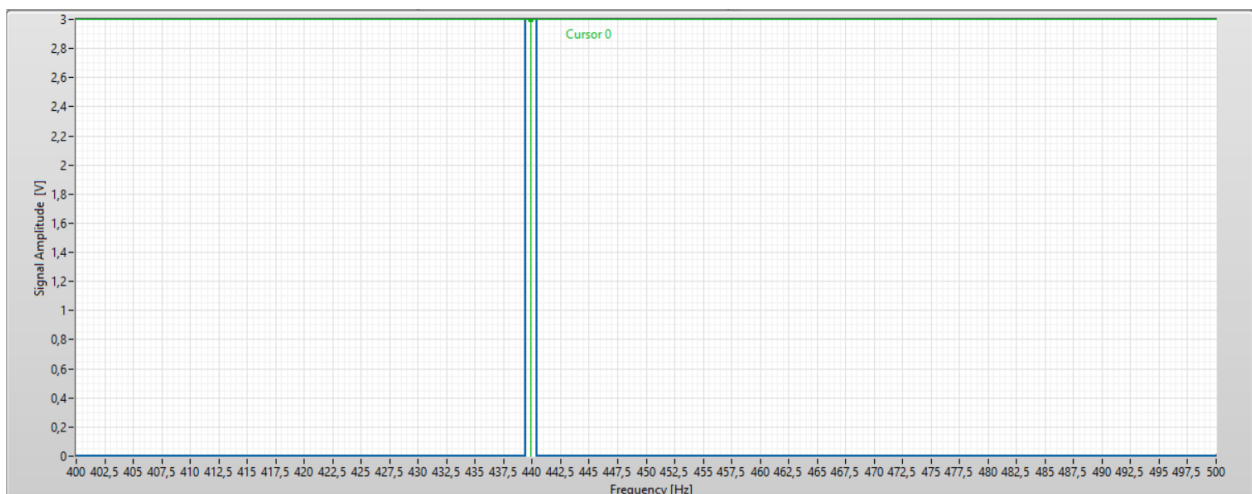
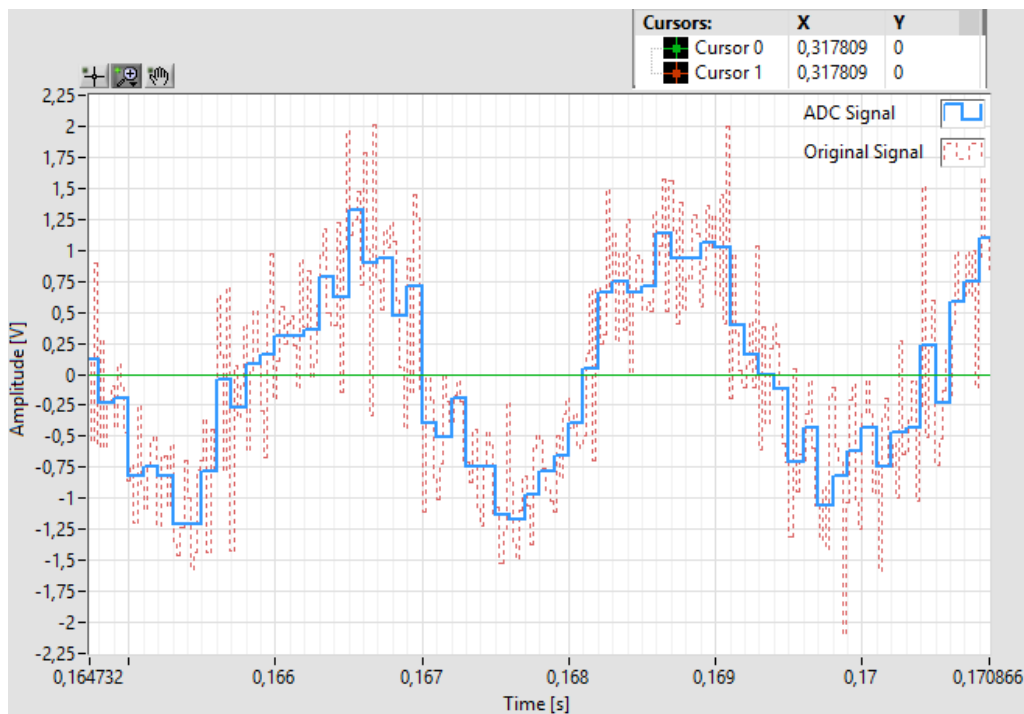


Fig. 2.2 – Representação espectral do sinal gerado.
(Obs: o sinal do .vi está sendo representado em escala monolog)

Simulador de filtros para análise de sinais

Como vimos nos exemplos do item 1.2, podemos também gerar ondas mais complexas, como ondas compostas por mais de uma senoide, ruídos e até mesmo senoides somadas a ruídos, sendo estas exploradas com mais atenção na seção que tratará sobre filtros.

Por enquanto, selecione a opção “Sine” e “enable noise” no menu “Signal Type”. Clique no menu “Signal View”, para visualização do sinal para o domínio do tempo. Ajuste os parâmetros do ADC para visualizar o sinal após a conversão. Altere a escala no eixo x para visualizar melhor o sinal senoidal bem ruidoso, destacado em vermelho. Este sinal, após passar por um filtro passa-baixas de primeira ordem e frequência de corte de 50 kHz, fica com um aspecto menos ruidoso, como o sinal indicado no traço em azul na Fig. 2.3.



Os controles do filtro estão posicionados no canto inferior direito do programa (entre eles: tipo do filtro, ordem, frequência de corte), como mostrado na Figura 2.4a, e o gráfico da resposta em frequência do “filtro projetado” é indicado no gráfico ao selecionar “Filter View – [frequency domain]”, como o exemplo da Figura 2.4b.

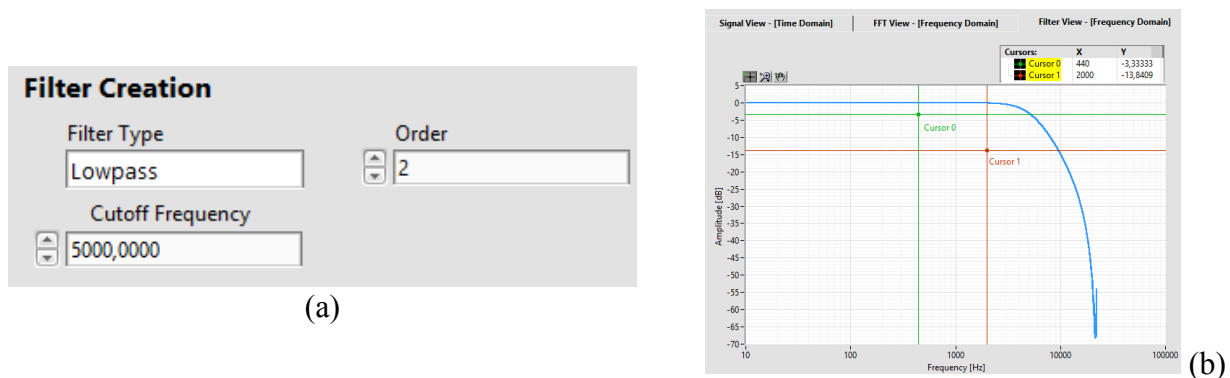


Fig. 2.4 – Controles para geração de um filtro e a resposta em frequência do sinal após passar pelo filtro.

Simulador de filtros para análise de sinais

Perceba que o gráfico da Fig. 2.4b, denominado também de diagrama de Bode, corresponde à resposta da amplitude em função da frequência do filtro criado. Modifique o controle “Cutoff Frequency” e veja quais as consequências neste gráfico.

Conhecidas as funcionalidades do programa, vamos agora compreender como ele foi construído e qual parte iremos programar neste tutorial, seguindo a mesma filosofia dos tutoriais anteriores.

Simulador de filtros para análise de sinais

2.2 Inicialização

Neste programa também estão presentes as três etapas características de um VI baseado em uma arquitetura “General Purpose VI”. Deste modo, sabemos que na etapa de inicialização serão iniciados os valores de gráficos, os controles e serão configurados os canais de áudio (para que possamos utilizar os alto-falantes do computador), antes de efetuar-se qualquer processamento do programa.

2.3 Processamento

É na etapa de processamento que é realizada a programação da interface com usuário, assim como a geração de um sinal e seu devido processamento. Neste momento estamos apenas interessados em saber como aplicar um filtro nos sinais, e não como os sinais são gerados. Deste modo, focaremos nossa atenção na programação destacada na seção em verde na Fig. 2.5, denominada “Signal Filtering and Spectral Analysis”.

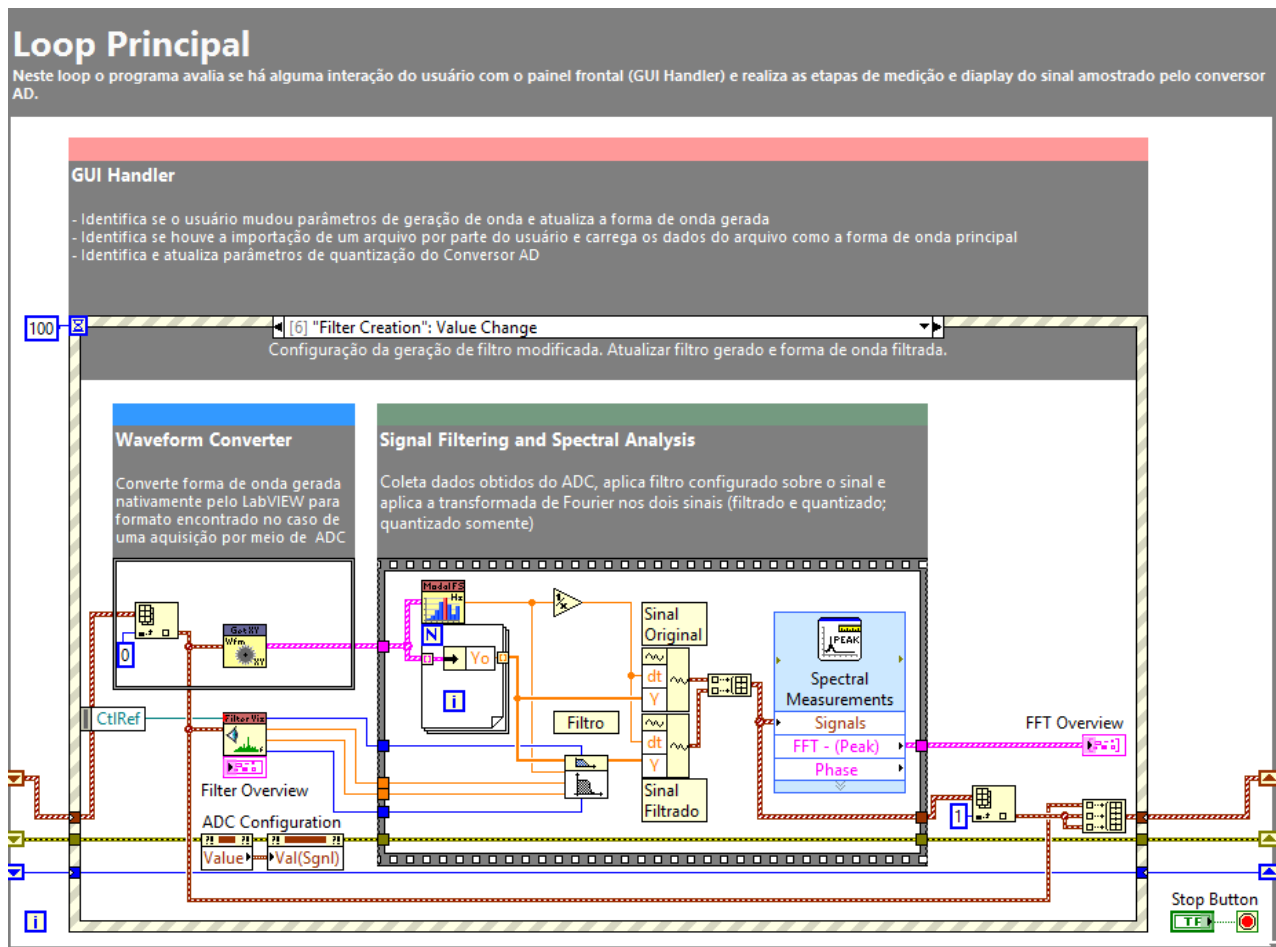


Fig. 2.5 – Etapa de Processamento, janela de criação do filtro.

Como notará, a programação que será realizada nesse tutorial é relativamente simples e demandará pouco tempo. Dito isso, é importante ressaltar que o objetivo do tutorial não é fazê-lo memorizar todos os métodos para se realizar uma análise espectral em LabVIEW, mas sim ajudá-lo a

Simulador de filtros para análise de sinais

compreender e consolidar os conceitos de análise de sinais no domínio da frequência em seus usos mais casuais.

2.4 Finalização

Por sua vez, na etapa de finalização é realizado o fechamento adequado do dispositivo de áudio e a correção de quaisquer erros gerados ao longo do programa.



Fig. 2.6 – Finalização do VI.

3 Filtros

Feche o programa [tut3_main.vi] e agora abra o programa [pr5.vi]. Tendo-se consolidado os conceitos básicos sobre representações dos sinais no domínio do tempo e da frequência e sobre transformações, vamos agora nos aprofundar no uso de filtros visando sua implementação em LabVIEW.

3.1 Intuição Geral de um Filtro

Em termos gerais, é importante compreendermos primeiramente que **quaisquer sistemas em engenharia têm, dentre outros parâmetros, uma faixa de operação na qual nos interessamos pelos sinais gerados ou capturados**. Fora dessa região assumiremos que quaisquer sinais são indesejados, denominados ruídos.

Em síntese, o objetivo de um filtro é retirar ruídos e preservar os sinais dentro da faixa de interesse. Na disciplina PSI3214 serão estudados filtros determinísticos com vista à sua caracterização no domínio da frequência. Todos os filtros que veremos serão construídos a partir de suas atenuações em certas faixas de frequências, Fig. 3.1, assim como fora realizado no projeto da disciplina de Laboratório de Circuitos Elétricos, PSI3212.

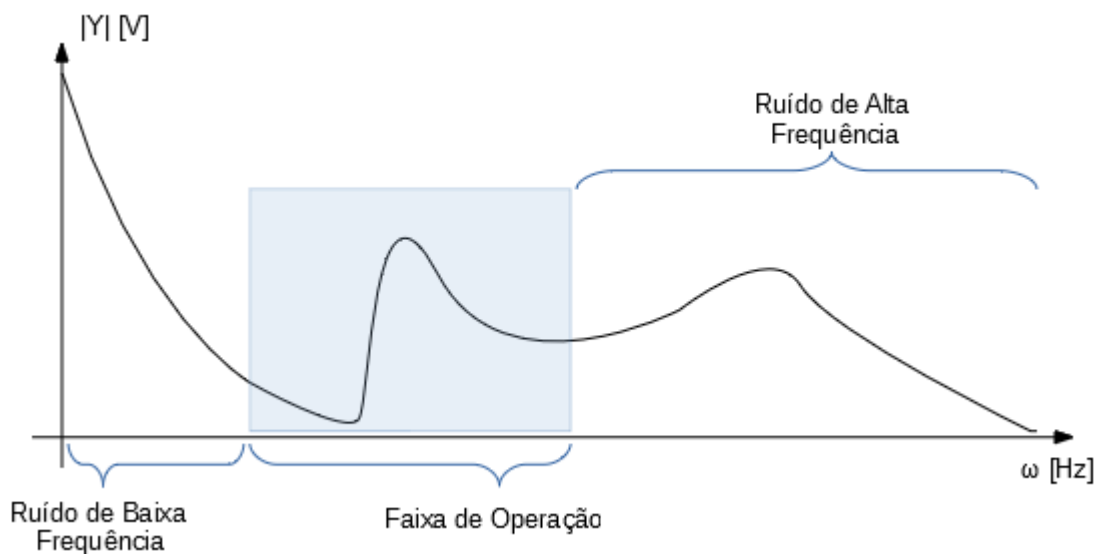


Fig. 3.1 – Faixa de operação de um sistema.

3.2 Construção de Filtros em LabVIEW

Vamos à etapa de construção de nosso filtro. Note que a região destacada em verde ([pr5.vi]), designada “Signal Frequency Domain Analysis”, encontra-se praticamente vazia, contendo apenas os controles necessários para criação do filtro desejado. Assim, para que seja possível efetuar qualquer cálculo com o vetor de pontos XY, antes de tudo é necessário que ele seja convertido em sinais com uma taxa amostragem determinada. (Por que tal ressalva? **Vale lembrar que os pontos adquiridos por um conversor AD não possuem necessariamente o mesmo intervalo de tempo entre amostras**. Justifica-se então essa primeira etapa). Como este procedimento já foi realizado no tutorial de Conversão AD, utilizaremos um VI que já se encontra em nosso projeto.

Simulador de filtros para análise de sinais

Para tal, efetue:

- (i) – Abra o projeto, [**Tutorial 3.lvproj**]
- (ii) – Expanda a pasta “Auxiliary VI’s”
- (iii) – Clique e arraste o programa “**Get modal Sampling Frequency.vi**” para a área verde no diagrama de blocos de seu VI.

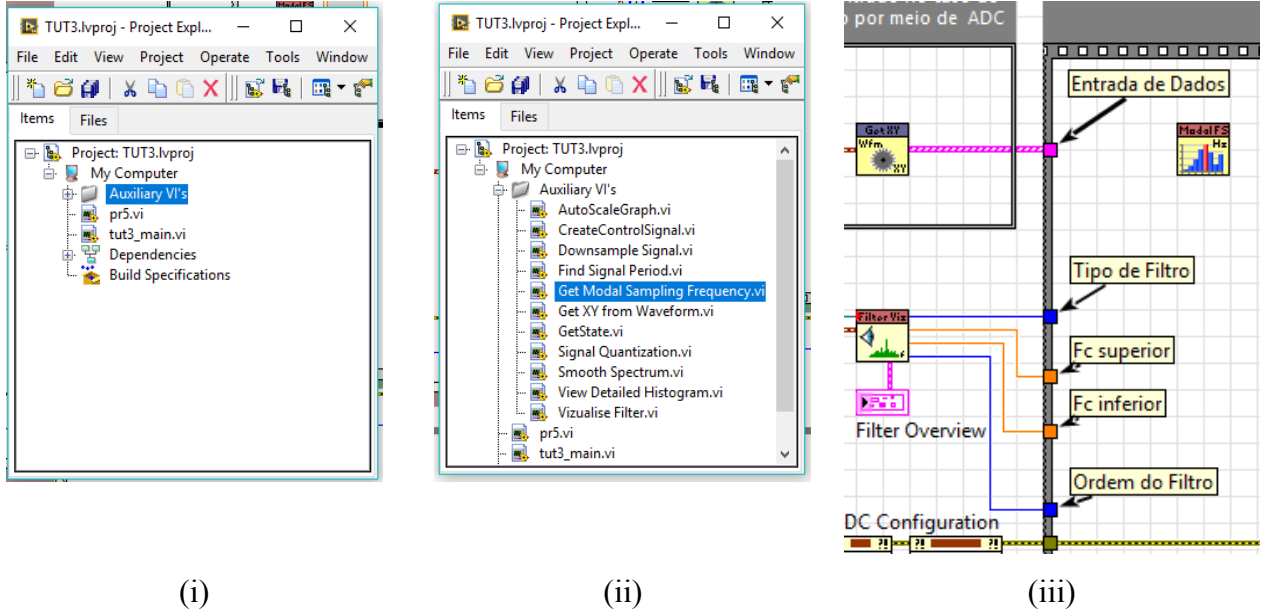


Fig. 3.2 – Procedimento para usar um VI que se encontra no projeto.

Feito isso, dê um duplo clique no VI arrastado para que possa visualizar o seu Painel Frontal. Perceba que este VI é similar aos outros até agora trabalhados. Use o atalho **Ctrl+E** para abrir seu diagrama de blocos, como o exemplificado na Fig. 3.3, e analise seu funcionamento. Note as semelhanças com o programa desenvolvido no Tutorial 2 (Conversão AD).

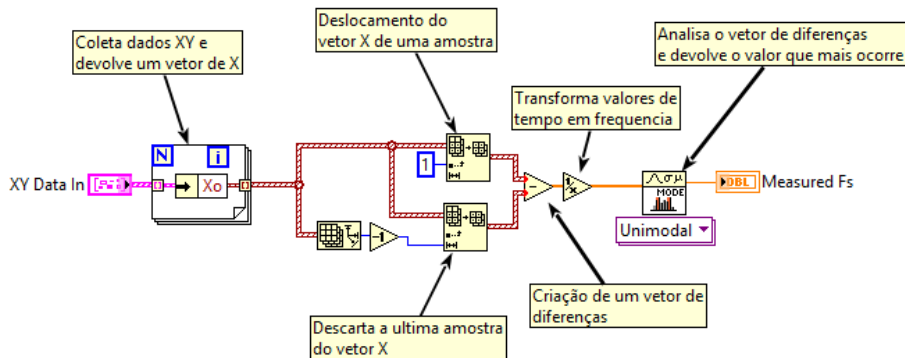


Fig. 3.3 – Programação para determinar a frequência de amostragem.

Simulador de filtros para análise de sinais

Ao analisarmos o diagrama da Fig. 3.3, concluímos que este VI que foi incorporado (“**modal Sampling Frequency.vi**”) aceita dados no formato XY na entrada e uma saída que nos indica qual sua taxa de amostragem. Feche o Diagrama de Blocos e o Painel Frontal desse VI.

Efetue, então, as indicações ilustradas na Fig. 3.4:

- (i, ii) – Conecte o vetor de dados “XY Data” à entrada do bloco Modal FS;
- (iii) – Crie um indicador para a taxa de amostragem medida (clique com o botão direito no terminal de saída e selecione create>indicator)

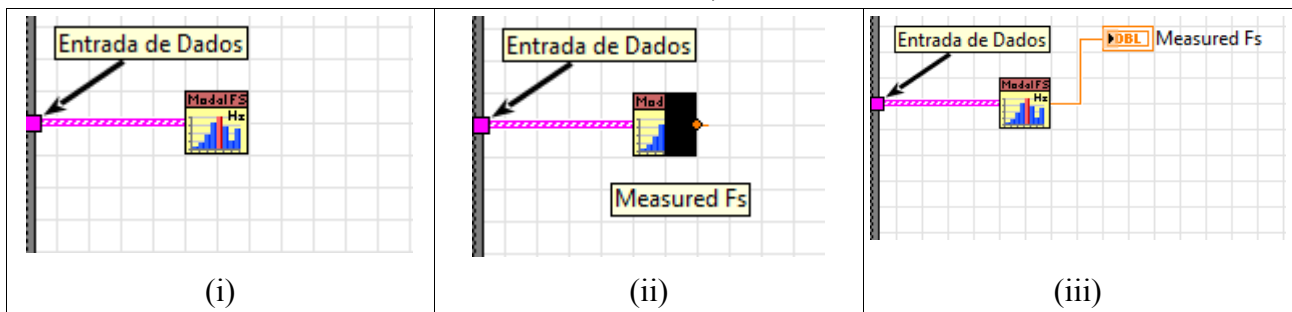


Fig. 3.4 – Conexão dos dados XY ao VI estimador da frequência de amostragem (ou Modal Sampling Frequency).

Nos passos a seguir, vamos efetuar uma programação para reconstruir o sinal adquirido com uma taxa de amostragem constante. Para isso, efetue os seguintes passos, conforme ilustrações das Fig. 3.5 e Fig.3.6:

- (i) – Crie um laço “For”;
- (ii) – Crie um bloco “Unbundle by Name” (aba “Cluster, Class & Variant”) e posicione dentro do laço;
- (iii) – Conecte: o sinal “XY Data” à entrada do laço; o túnel do esquerdo do laço à entrada do bloco “Unbundle by Name”; a saída do bloco “Unbundle by Name” à borda direita do laço.
- (iv) – Mude, se necessário, o elemento dentro do bloco “Unbundle by Name” para Yo (clique com o botão esquerdo para abrir o menu e selecione).

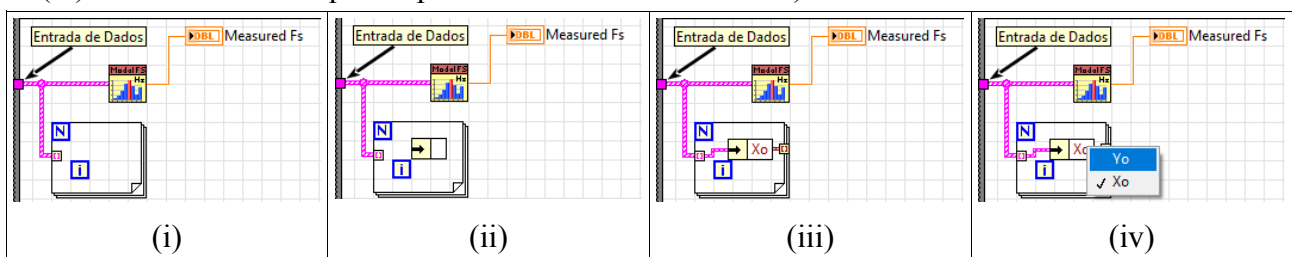


Fig. 3.5 – Seleção das amostras Y do sinal de entrada.

Simulador de filtros para análise de sinais

- (i) – Crie um bloco “Build Waveform” (na aba “Waveform”) e um bloco “reciprocal” (na aba “Numeric”);
- (ii, iii) – Expanda o bloco “Build Waveform”, arrastando sua parte superior. Clique com o botão esquerdo no elemento recém-criado para abrir o menu e selecione “dt”;
- (iv) – Conecte sinal “Measured Fs” à entrada do bloco “reciprocal”, e a saída desse bloco ao terminal “dt” do bloco “Build Waveform”. Conecte a saída do laço “For” ao terminal “Y” do bloco “Build Waveform”.

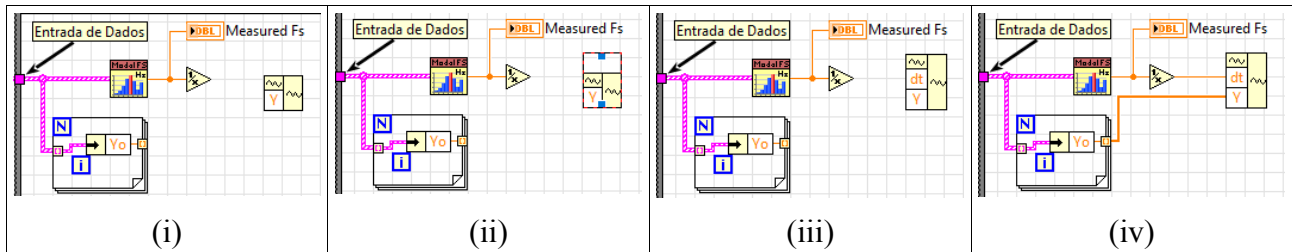


Fig. 3.6 – Reconstrução do sinal original com taxa de amostragem constante.

Resumindo, todas as operações acima foram efetuadas para que pudéssemos coletar apenas as amostras “y” do vetor XY. Criamos então um vetor Y com tais amostras e construímos um sinal do tipo “Waveform”, agora com taxa de amostragem constante.

Vamos construir agora a etapa de filtragem do sinal: abra a paleta de funções do Diagrama de Blocos. Navegue até a seção “Signal Processing>Filters” e observe que temos nesta “pasta” uma vasta gama de filtros que podemos fazer uso, cada qual com suas vantagens e desvantagens. Nesse tutorial utilizaremos o mais simples e intuitivo dos filtros disponíveis: o tipo de filtro denominado “Butterworth Filter”. Após a construção do filtro veremos com mais calma cada um de seus aspectos. Por enquanto, efetue a seguinte programação, seguindo as ilustrações das Fig. 3.7 e Fig. 3.8:

- (i) – Crie um bloco “Butterworth Filter”;
- (ii) – No bloco do filtro (i), conecte o sinal “Measured Fs” à entrada “sampling freq: fs” e o sinal Y da saída do laço “For”, à entrada “X”.
- (iii) – Conecte os valores utilizados para criação do filtro (“Tipo de Filtro”, “Fc superior”, “Fc inferior” e “Ordem do Filtro”) às entradas correspondentes do bloco do filtro: “filter type”, “high cutoff freq: fh”, “low cutoff freq: fl” e “order”.

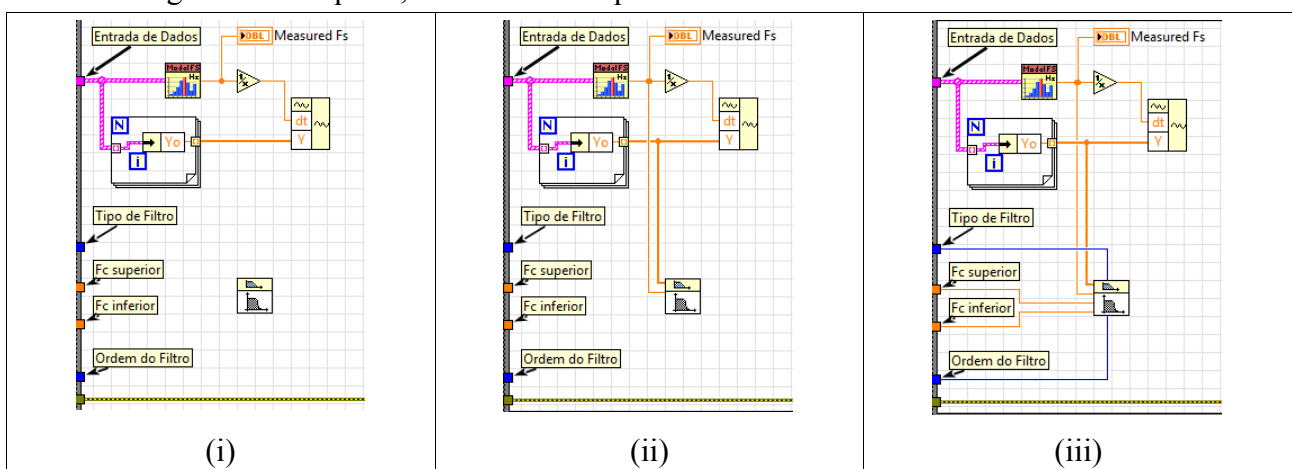


Fig. 3.7 – Criação do filtro Butterworth.

Simulador de filtros para análise de sinais

- (i) – Crie mais um bloco “Build Waveform” e arraste sua borda superior para que possamos adicionar mais de um parâmetro ao mesmo tempo.
- (ii, iii) – Neste novo bloco “build Waveform” criado, conecte o sinal dt , já usado em outros blocos, à sua entrada dt , e o sinal de saída do bloco “Butterworth Filter”, denominado “Filtered X”, à sua entrada Y .

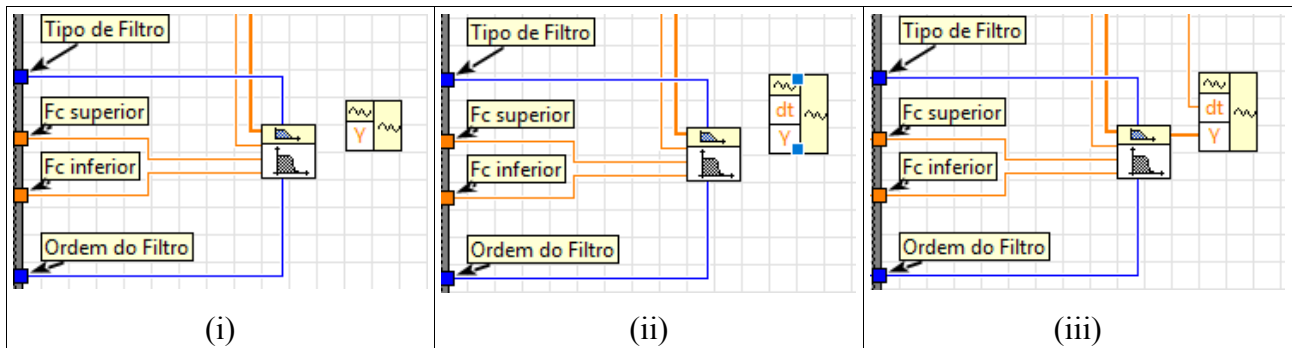


Fig. 3.8 – Construção do sinal filtrado.

Teremos criado, com esses passos de programação, dois sinais distintos. No bloco “Build Waveform” superior será gerado o sinal com os valores de Y originais, imutados. No bloco inferior, por sua vez, será gerado um sinal que apresenta valores de Y atenuados, de acordo com a ação do filtro criado.

Vamos agora criar um vetor contendo os dois sinais para que possam ser visualizados no mesmo gráfico. Efetue a seguinte programação, de acordo com as ilustrações indicadas na Fig. 3.9:

- (i) – Crie um bloco “Build Array” (na aba “Array”) e expanda-o para que possa acomodar duas entradas.
- (ii) – Conecte as saídas dos dois blocos “Build Waveform” ao bloco recém criado como indicado. O sinal superior deverá entrar no terminal superior do bloco “Build Array”, e a saída do bloco inferior deverá ser conectada ao terminal inferior.
- (iii) – Conecte a saída do bloco “Build Array” ao túnel marrom inferior da saída de Dados.

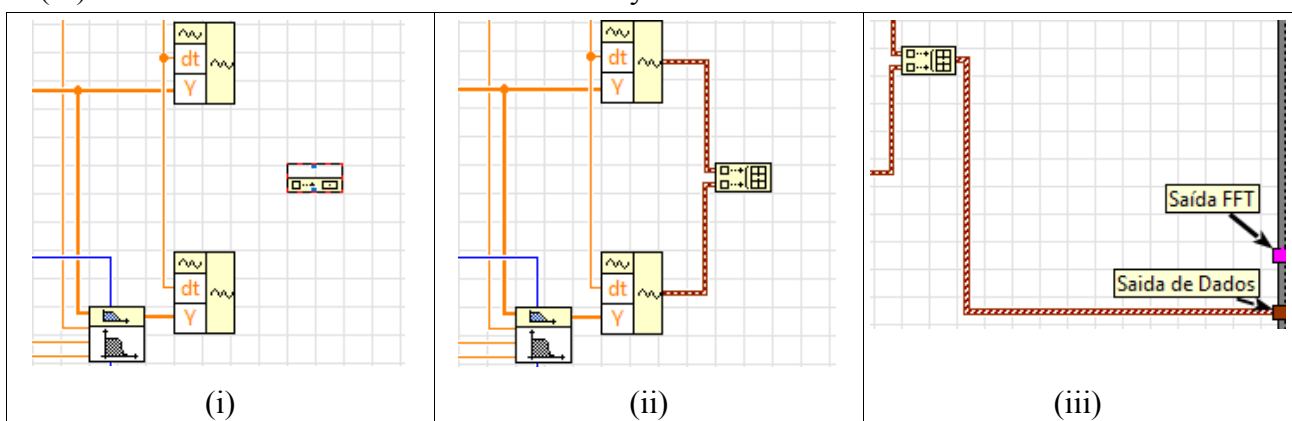


Fig. 3.9 – Junção dos dois sinais em um vetor.

Com isso temos um programa funcional pronto para ser testado! Execute o programa e veja se tudo funciona segundo o esperado.

Simulador de filtros para análise de sinais

3.3 Aplicação em um sinal

Com o programa em execução, vamos verificar alguns pontos interessantes. Primeiramente, configure os controles dos sinais e do filtro para que tenham as seguintes características:

– Signal Type: Sine	– Filter Type: Lowpass
– Signal Amplitude: 3 V	– Low Cutoff Freq: 5E+4 Hz
– Signal Frequency: 440 Hz	– High Cutoff Freq: 5E+4 Hz
– Sampling Frequency: 1E+6	– Order: 1

Observe o sinal no gráfico principal e note que a senoide criada tem atenuação desprezível, pois sua amplitude é praticamente 3 V.

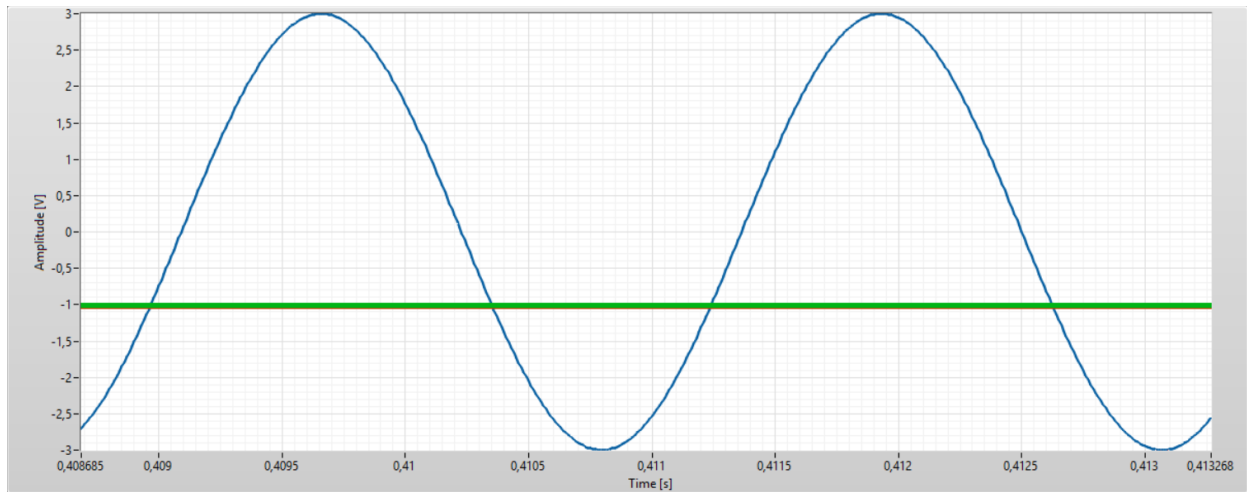


Fig. 3.10 – Senoide 440 Hz com filtro passa-baixa ($F_c = 50$ kHz).

Mude o valor do controle “Low Cutoff Freq” para 440 Hz. Utilize os cursores para confirmar que o ganho entre o sinal original, em vermelho, e o sinal filtrado, em azul, é de -3 dB.

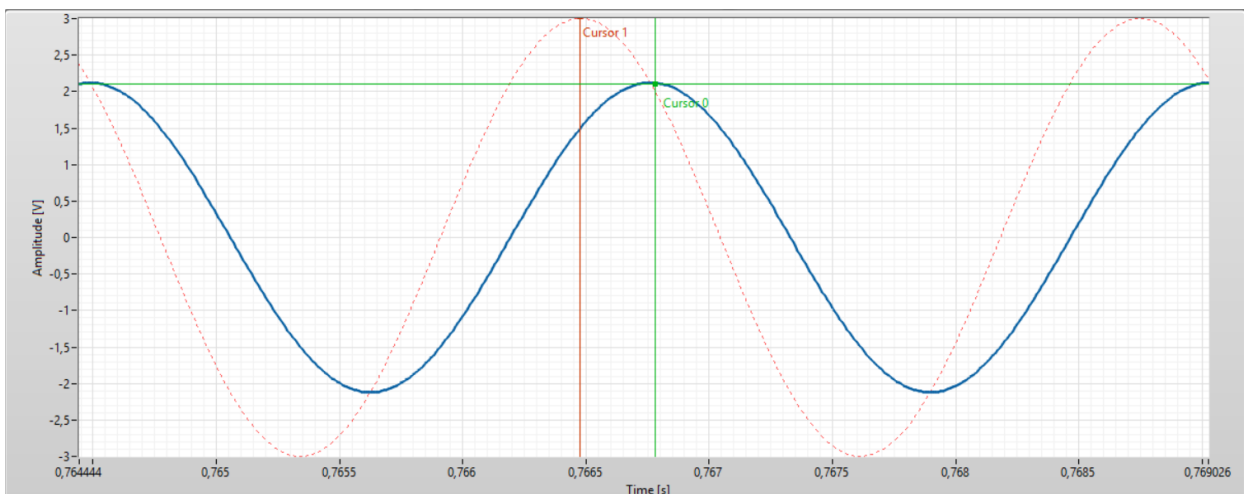


Fig. 3.11 – Senoide 440 Hz com filtro passa-baixa ($F_c = 440$ Hz).

Note também que há um atraso de fase entre os sinais de aproximadamente 45° , como esperado de um filtro passa-baixa de primeira ordem.

Mude agora o controle “Signal Type” para “Square” e veja o resultado (deve ser similar ao da Fig. 3.12).

Simulador de filtros para análise de sinais

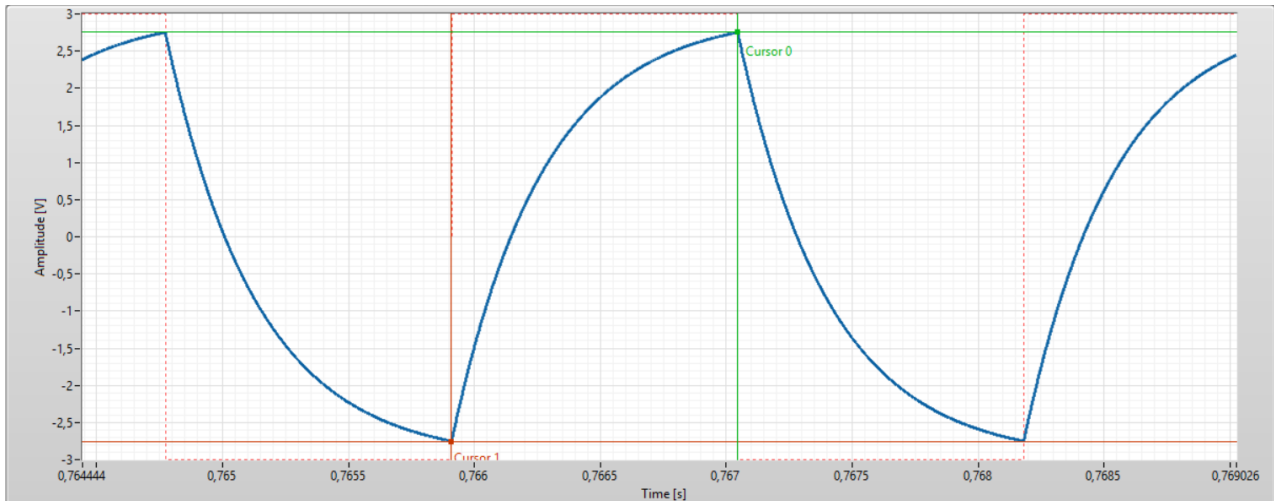


Fig. 3.12 – Onda quadrada 440 Hz com filtro passa-baixa ($F_c = 440$ Hz).

Perceba que temos uma resposta idêntica àquela de um circuito RC, como realizado na simulação do tutorial 1. Aqui, no entanto, utilizamos apenas um parâmetro, a frequência de corte do filtro, para determinar uma resposta que foi obtida no tutorial 1 com grande sacrifício de clareza.

Vamos agora iniciar a parte que mais nos interessa nesse tutorial. Mude o valor do controle “Signal Type” para “Sine with Noise” e veja os resultados.

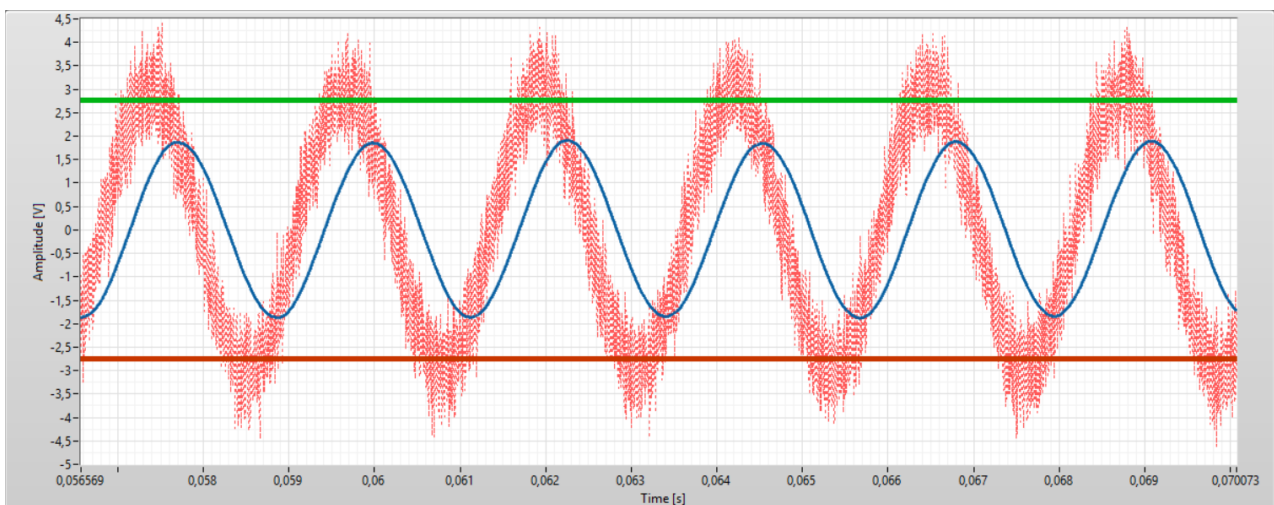


Fig. 3.13 – Senoide 440 Hz + ruído gaussiano, sem e com filtro passa-baixas ($f_c = 440$ Hz).

Finalmente os frutos de nossos esforços estão sendo recompensados! Fomos capazes de obter uma estimativa do sinal, esboçado em azul, após tê-lo “passado” por um filtro passa-baixas com frequência de corte de 440 Hz.

4 FFT

Pronto! Temos finalmente um filtro funcional, correto?

De fato, o filtro construído na seção anterior atenua frequências indesejadas para o sinal testado. Contudo, se o sinal fosse composto por outras frequências, ficaria muito atenuado após passar pelo filtro, provavelmente muito mais atenuado do que o desejado. Tente mudar as frequências do controle “Signal Frequency” para averiguar esse comportamento.

Para compreender melhor como o sinal é atenuado após a ação do filtro em relação ao original, vamos implementar um bloco de **Transformada de Fourier**, para que seja possível visualizar o mesmo sinal no domínio da frequência.

Interrompa a execução do programa (se ainda estiver com o comando “run” acionado) e abra seu diagrama de blocos para a implementação desta nova funcionalidade.

A implementação de medidas como FFT em LabVIEW é simples e direta:

- (i) – Delete a constante “FFT Constant” já existente no seu VI, selecionando-a e apertando o botão “delete”. Crie um bloco “Spectral Measurements” (vá na aba “Signal Processing > Waveform Measurements”). Feita sua criação, deverá surgir uma janela de configuração da medida espectral, na qual podemos selecionar parâmetros pertinentes.
- (ii) – Configure o bloco para que apresente as medidas com Magnitude (Pico), Linear, e selecione a opção de nenhuma janela (“None”). Feito isso, clique em OK.
- (iii) – Conecte o sinal de saída do bloco “Build Array” à entrada “Signals” do bloco de FFT e, em seguida, conecte o terminal de saída “FFT – (Peak)” ao mesmo túnel que fora utilizado pela constante “FFT Constant”. Deste modo, conectamos a saída da FFT aos dois indicadores “FFT Preview” e “FFT Overview”, localizados na área delimitada “Signal Visualization”.

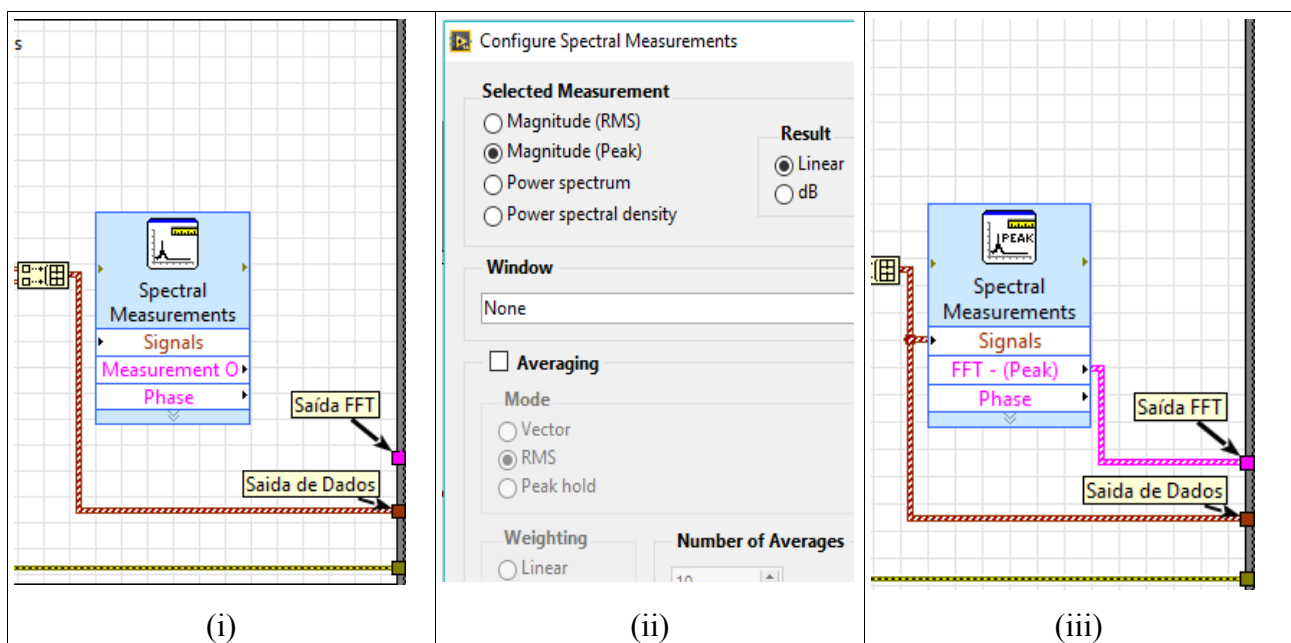


Fig. 4.1 – Construção das medidas espectrais.

Feito isso, temos nosso programa pronto e desenvolvido em LabVIEW para avaliar os sinais tanto no domínio do tempo quanto no domínio da frequência. Poderemos também avaliar como ficará o sinal nos dois domínios após a ação de um filtro projetado.

Simulador de filtros para análise de sinais

4.1 Aplicação do Filtro em FFT (Comparação)

Execute o programa e insira os valores abaixo nos controles:

– Signal Type: Square	– Filter Type: Lowpass
– Signal Amplitude: 1 V	– Low Cutoff Freq: 5E+5 Hz
– Signal Frequency: 440 Hz	– High Cutoff Freq: 5E+5 Hz
– Sampling Frequency: 1E+6	– Order: 1

Note que obteremos uma onda quadrada como na figura abaixo:

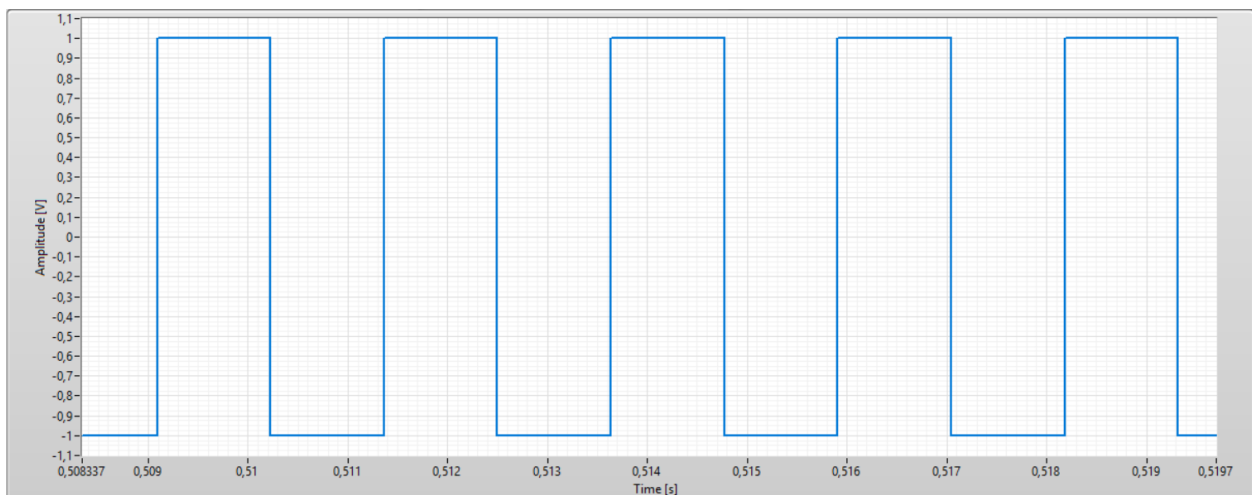


Fig. 4.2 – Onda quadrada 440 Hz com filtro passa-baixa ($f_c = 50$ kHz).

Agora que podemos visualizar a FFT do sinal, clique no menu “Change Graph View” e selecione a opção “FFT View”. Nela deveremos ver a seguinte figura:

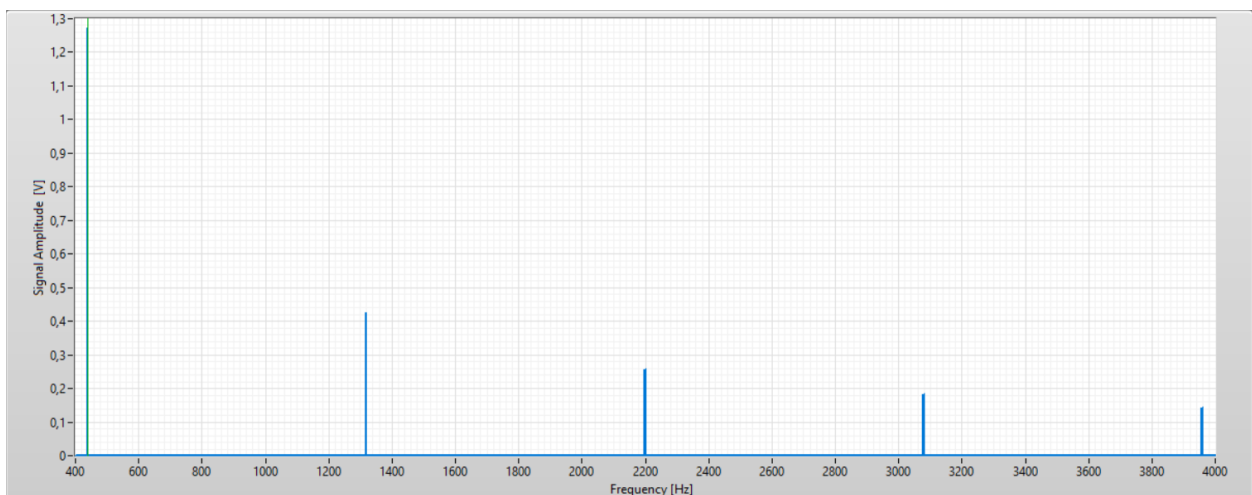


Fig. 4.3 – Espectro de uma onda quadrada de 440 Hz.

Note que a Fig. 4.3 mostra as raiais correspondentes à Série de Fourier da onda quadrada, obtidas através de um VI que calcula a FFT do sinal, ao selecionar uma janela contendo períodos inteiros do sinal.

Modifique agora o valor da frequência de corte do filtro passa-baixas para 500 Hz e veja como os picos da FFT se comportam. Mude o gráfico para domínio do tempo e veja o sinal correspondente. Note que o sinal se deforma por conta da atenuação dos harmônicos ímpares de ordem superior e

Simulador de filtros para análise de sinais

que, apesar de possuírem frequências maiores do que a frequência de base, são imprescindíveis para caracterização da onda quadrada.

Do mesmo modo, outros sinais mais complexos, como notas musicais, também possuem caracterizações mais complexas no domínio da frequência. Utilizando o botão de importação do VI, importe o arquivo de nota musical utilizado no Tutorial de Conversão AD e veja como ele é caracterizado por suas frequências.

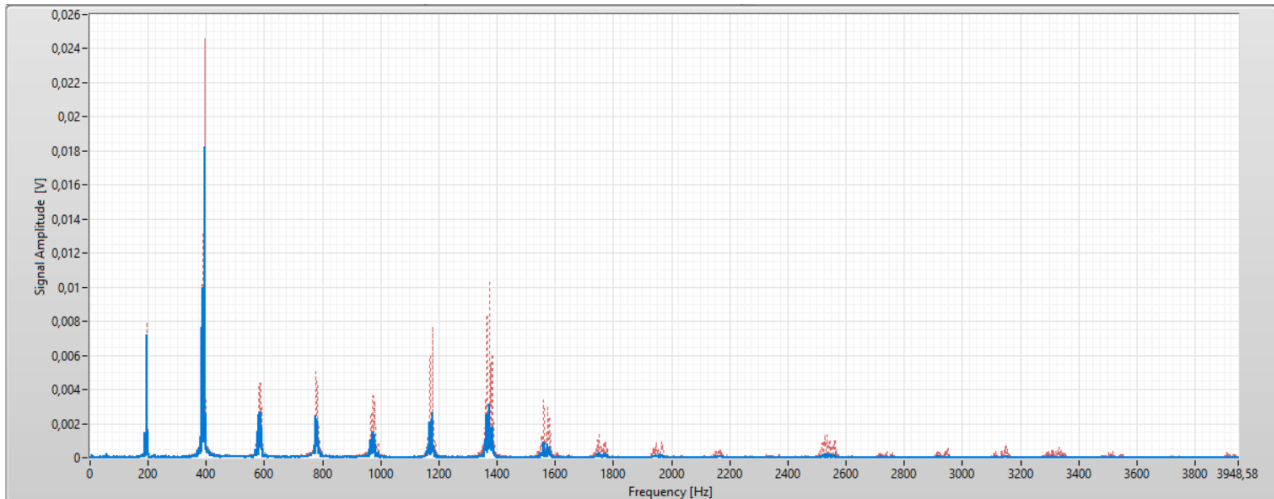


Fig. 4.4 – Distribuição espectral de uma nota de violino.

Observe que neste exemplo temos frequências com amplitudes não desprezíveis até 2 kHz. Modifique a frequência de corte do filtro para que sejam incluídos todos os harmônicos até esta frequência. Veja se seu sinal filtrado no domínio do tempo se aproxima adequadamente do original.

4.2 Frequências dominantes

Vimos que um sinal qualquer pode conter diversas frequências que o caracterizam, e que, conhecendo-se tais frequências, um filtro poderá ser projetado adequadamente. Por exemplo, suponha que um sinal desconhecido seja adquirido por um ADC. Através da análise espectral deste sinal é possível identificar quais são suas “frequências dominantes”, ou seja, aquelas que possuem maior energia, ou aquelas que apresentam maior amplitude no domínio da frequência. Por meio desta análise identifica-se a frequência dominante. É importante destacar que a análise do período de um sinal no domínio do tempo é traduzida no domínio da frequência como a busca pela frequência de maior amplitude do sinal.

Lembre-se daquele problema discutido no Tutorial 2 para determinação do período (e frequência) de um sinal desconhecido no domínio do tempo. Naquele tutorial verificamos que o algoritmo de “zero-crossing” desenvolvido não foi muito preciso quando o aplicamos para sinais mais complexos e ruidosos. Naquele exemplo utilizamos o mesmo sinal apresentado na Fig. 4.4, porém nosso algoritmo de detecção de período/frequência forneceu como resposta um valor em torno de 350 Hz, 50 Hz abaixo da frequência dominante de 400 Hz. Agora que está mais familiarizado com a análise espectral, note que através dela temos a solução para tal problema.

Conclusão do Programa

Ao longo desse tutorial vimos como sinais podem ser filtrados em LabVIEW, e como obter uma representação de um sinal no domínio da frequência. Por meio dessas ferramentas, pudemos compreender melhor a natureza de alguns sinais como de ondas quadradas e notas musicais, de modo que, dado um sinal de entrada dentro de certas condições de contorno, sabemos qual o melhor filtro para ele.

Vamos agora colocar todos esses conceitos e ferramentas em prática, aplicando-os no projeto extraclasse!