

AE1 - Revisão de Sinais e Sistemas

Jessica de Souza

Engenharia de Telecomunicações, Instituto Federal de Santa Catarina

[<jessica.souzajds@gmail.com>](mailto:jessica.souzajds@gmail.com)

9 de janeiro de 2019

1 Introdução

O objetivo desta atividade é realizar alguns experimentos envolvendo conteúdos de Sinais e Sistemas utilizando a ferramenta MATLAB ou OCTAVE. Podemos citar os seguintes assuntos que foram abordados nesta atividade: filtragem de sinais, análise de sinais no domínio da frequência, revisão de sinais nos tempos contínuo e discreto, etc. O livro referência destes experimentos é chamado "Processamento Digital de Sinais: Projeto e Análise de Sistemas" do autor DINIZ e colaboradores.

Este relatório divide-se em quatro partes: (1) Análise de sinais em tempo discreto utilizando diferentes tipos de filtragem, (2) análise de sinais no domínio da frequência, (3) análise de filtros e amostragem e (4) geração e filtragem de um sinal DTMF.

2 Experimentação

2.1 Análise de Sinais em Tempo Discreto Utilizando Diferentes Tipos de Filtragem

Este experimento consiste em mostrar que um sinal de saída $y(t)$ vindo de um determinado filtro digital, pode ser obtido através de diferentes métodos no Octave, dos quais citamos: conv, filter, transformada de Fourier. Tais métodos foram testados e verificamos sua eficiência realizando medições do tempo de execução de cada um deles.

Para iniciar os testes, foram criados dois sinais $x(t)$ e $h(t)$, com os seus respectivos número de amostras $N_h = 10$ e $N_x = 20$. Na figura 1 podemos observar os resultados das filtrações realizadas.

O primeiro gráfico é o resultado das três filtrações, que foram sobrepostos para que possamos visualizar que todos os métodos levam a praticamente o mesmo resultado, com exceção do sinal y_2 . Podemos ver que apenas o sinal y_2 tem tamanho 20, diferente dos demais que possuem tamanho 29. O segundo gráfico mostra o sinal y_1 , que foi gerado utilizando a operação conv(), que consiste da convolução entre $h(t)$ e $x(t)$. O terceiro gráfico mostra o sinal y_2 , que foi gerado através da operação filter(), podemos observar que ele mostra menos informação do que os outros pois a operação filter tem como o vetor de saída o tamanho do maior vetor de entrada, não criando novos vetores, isso não é indicado se o experimento requer a obtenção de todas as amostras não-nulas da saída. O quarto gráfico mostra o sinal y_3 que foi gerado utilizando também o método filter, porém seu tamanho foi ajustado para a soma do comprimento de N_h e N_x menos 1, seu resultado foi igual ao de y_1 .

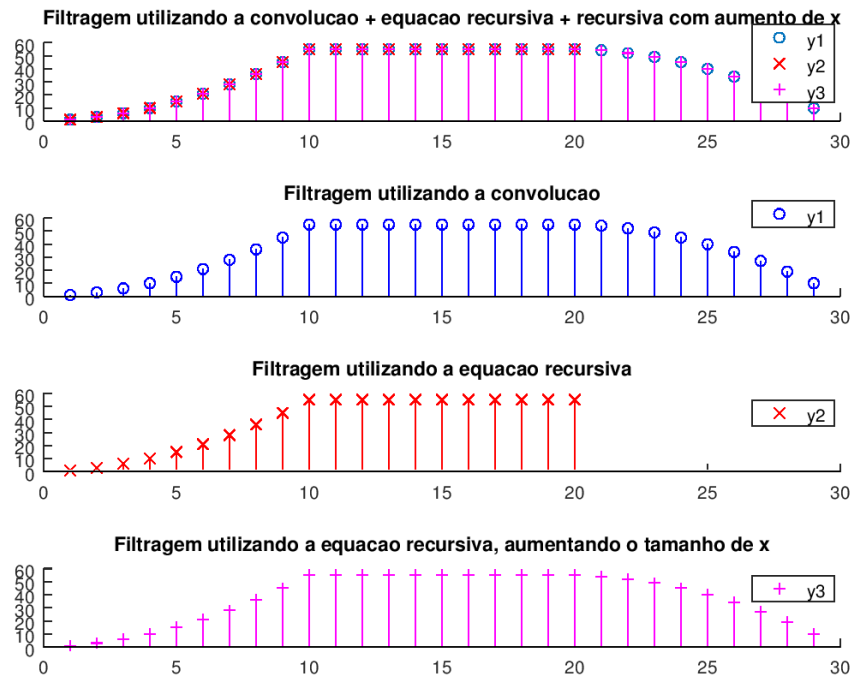


Figura 1 – Resultado das filtragens utilizando diferentes métodos

Além dos filtros mostrados na figura 1, outros filtros foram realizados para que pudéssemos verificar seus tempos de execução. A tabela 1 mostra a relação de todos os filtros realizados e quanto tempo cada um levou em 3 execuções em série. Podemos observar que a função `fftfilter()` obteve um tempo maior de execução pois a função `fftfilter()` possui um loop em seu interior e ele fica localizado em um arquivo `.m` separado no MATLAB, enquanto a função `conv()` é embutida, tornando mais rápido acessá-la e executá-la.

Filtros						
Evento	<code>conv()</code>	<code>filter()</code>	<code>filter(maior)</code>	<code>fft()</code>	<code>fftfilter()</code>	Tempo
E1	6503.8204	0	0	0	9006.0234	us
E2	4000.9022	0	0	0	8006.0959	us
E3	3501.8921	501.15585	0	0	6505.0125	us

Após esta primeira verificação dos filtros, foi aumentado os tamanhos dos sinais N_h e N_x , para 400 e 10000, respectivamente. Verificando os tamanhos obtidos para $y(t)$, foram obtidos para 4 operações o tamanho de 10399 ($N_h + N_x - 1$) e para 1 operação o tamanho de 10000, e razão dos tamanhos de cada operação foi explicado anteriormente. A figura 2 ilustra como ficaram os filtros com uma maior quantidade de amostras.

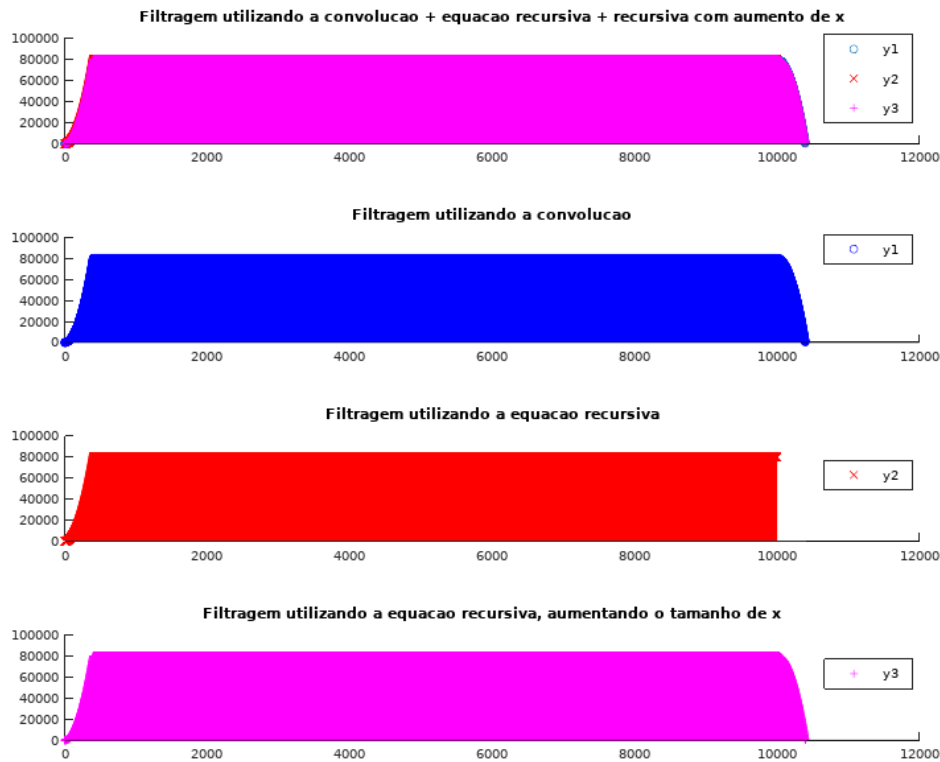


Figura 2 – Resultado das filtragens utilizando diferentes métodos e mais amostras

Verificando os tempos de execução, foram obtidos os seguintes resultados:

Filtros						
Evento	conv()	filter()	filter(maior)	fft()	fftfilt()	Tempo
E1	13008.8329	6004.09508	7007.12204	10005.9509	10027.8854	us
E2	13008.1177	8006.09589	6002.90298	11008.9779	8004.90379	us
E3	12008.9054	7007.12204	7024.04976	9985.92377	9006.97708	us

Dessa vez o método `conv()` levou mais tempo do que o `fftfilt()`. O `fftfilt()` é mais indicado quando há uma grande quantidade de amostras, é o que indica DINIZ (2014), pois o sinal transformado para o domínio da frequência diminui a complexidade da operação (a convolução se torna multiplicação) e consequentemente o tempo de execução.

2.2 Análise de Sinais no Domínio da Frequência

A figura a seguir mostra um sinal senoidal com frequência de 40 Hz e frequência de amostragem de 200 Hz (primeiro gráfico), um ruído gerado aleatoriamente com a função `rand()` e um ruído branco gerado com a função `awgn()` com ambos somados ao sinal original. Estes sinais foram filtrados na frequência de 40 Hz utilizando a transformada de Fourier (segundo gráfico) de forma que possamos ver sua componente na frequência, que é mais fácil de identificar do que no domínio do tempo devido ao ruído.

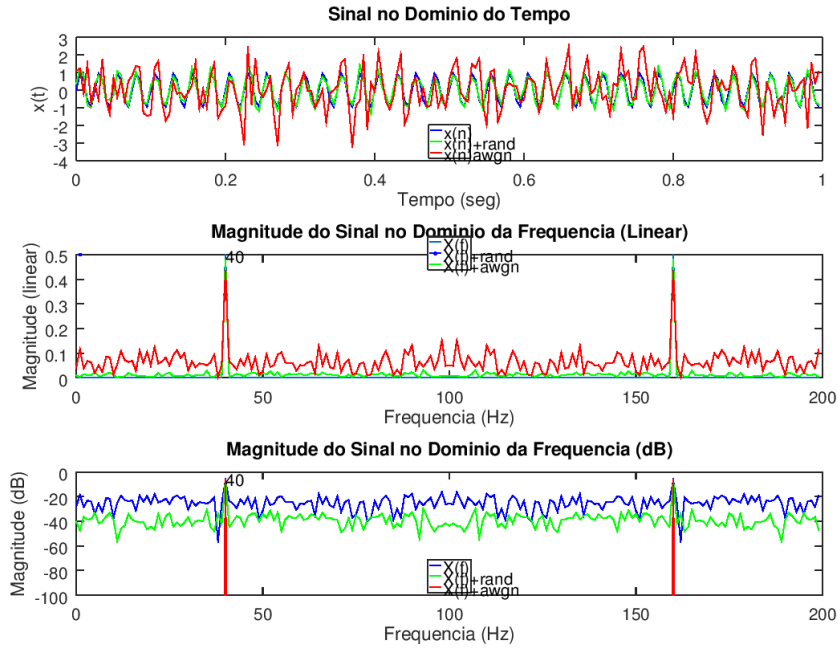


Figura 3 – Resultado das filtragens a transformada de Fourier

O primeiro ruído, gerado com `randn()`, utiliza um parâmetro k de intensidade baixa (por volta dos 0,2). Se variarmos o seu valor entre 0 e 2, podemos verificar que quanto maior fica este valor, mais desfigurado o ruído fica pois este impacta no formato do sinal significativamente (devido a função que o gerou). No domínio do tempo o ruído se destacou mais do que o sinal original, sendo impossível retirar qualquer informação ao observá-lo. Porém no domínio da frequência linear ainda conseguimos visualizar a frequência do sinal original ao filtrá-lo. Já ao verificar o sinal no domínio da frequência em dB, não temos visualmente um valor confiável desta filtragem. A figura a seguir mostra como ficou o sinal original e filtrado com k no seu valor máximo.

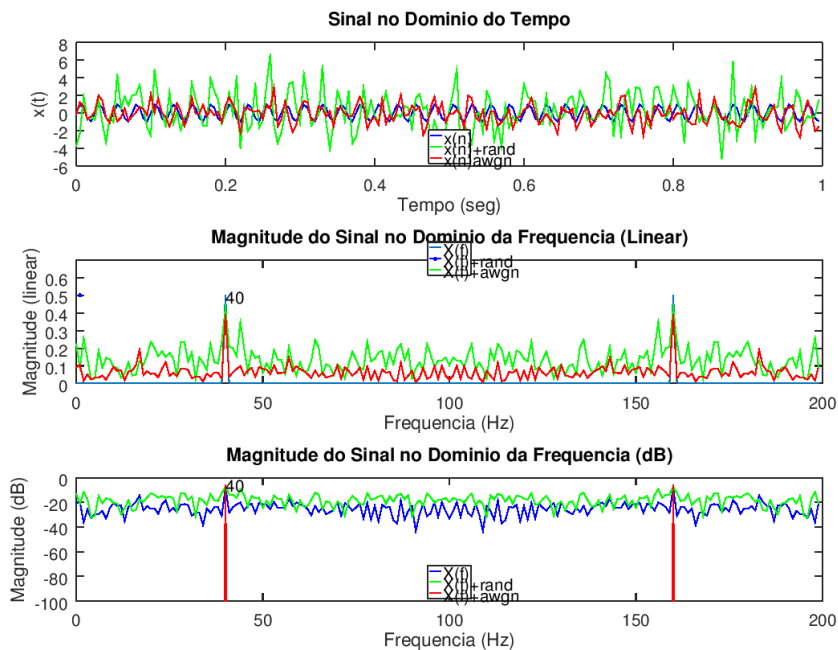


Figura 4 – Resultado das filtragens a transformada de Fourier com variação em k .

O segundo ruído, gerado com a função `awgn()` utiliza um parâmetro `snr` que é a relação sinal ruído desejada. Inicialmente este valor estava em 0. Se variarmos este valor, por exemplo, entre um valor alto ($\text{snr} = 100$) e baixo ($\text{snr} = 1$), poderemos verificar que quanto maior o valor de `snr`, melhor fica a visualização do sinal. Isso ocorre porque a equação da relação sinal ruído é quanto de sinal temos para um determinado ruído. O parâmetro `snr` com um valor alto nos dá uma maior quantidade de sinal. Ao variar este tipo de ruído, podemos verificar que este é significativamente melhor de se analisar tanto no domínio do tempo quanto da frequência, pois seus picos ficam bem nítidos após a filtragem. A figura a seguir mostra o sinal no domínio do tempo e da frequência com o parâmetro $\text{snr} = 100$.

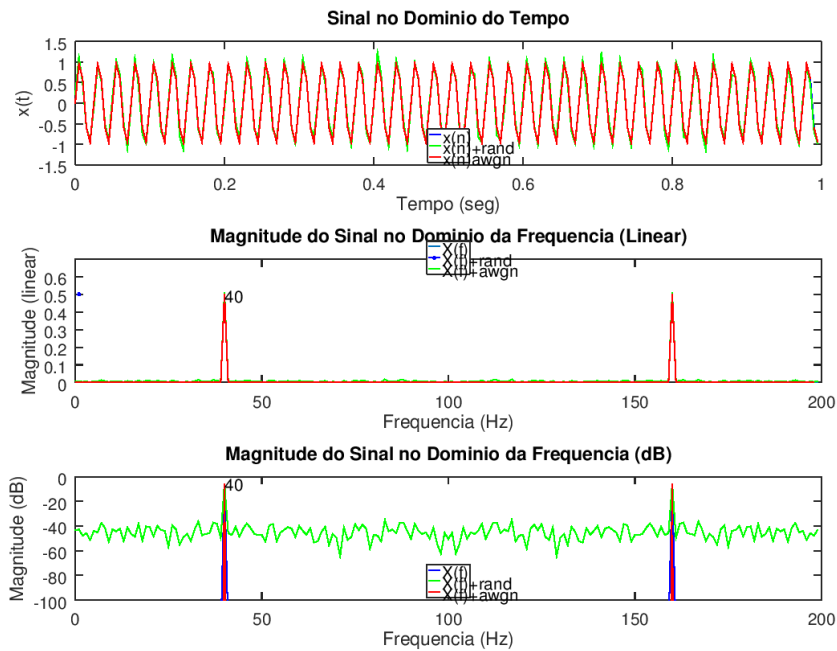


Figura 5 – Resultado das filtrações a transformada de Fourier com variação em `snr`.

Se mantermos os valores da constante de ruído $k = 0.3$ e a de relação sinal ruído $\text{snr} = 40$ e variarmos a frequência do sinal entre 0 até 200 Hz, podemos verificar que de 0 a 100 Hz os sinais no tempo e na frequência são os mesmos que de 200 a 100, nesta ordem. O espectro vai se movendo gradativamente até 100 Hz, com espelhamento na faixa de frequência acima de 100 Hz e a situação contrária é análoga.

2.3 Análise de Filtros e Amostragem

O experimento a seguir consiste em gerar dois sinais, com amplitudes $A_1 = 0.3$ e $A_2 = 0.8$, e amostrá-los a frequências determinadas. Inicialmente, a frequência de amostragem foi ajustada para 6 Hz e podemos observar na figura a seguir que o número de amostras discretas é o mesmo da frequência. Se aumentarmos esse número para 20 Hz obtemos uma maior quantidade de amostras, facilitando sua interpretação. A figura a seguir mostra como ficou no domínio do tempo os dois sinais com $f_s = 20$ Hz (figura da esquerda) e com $f_s = 6$ Hz (figura da direita).

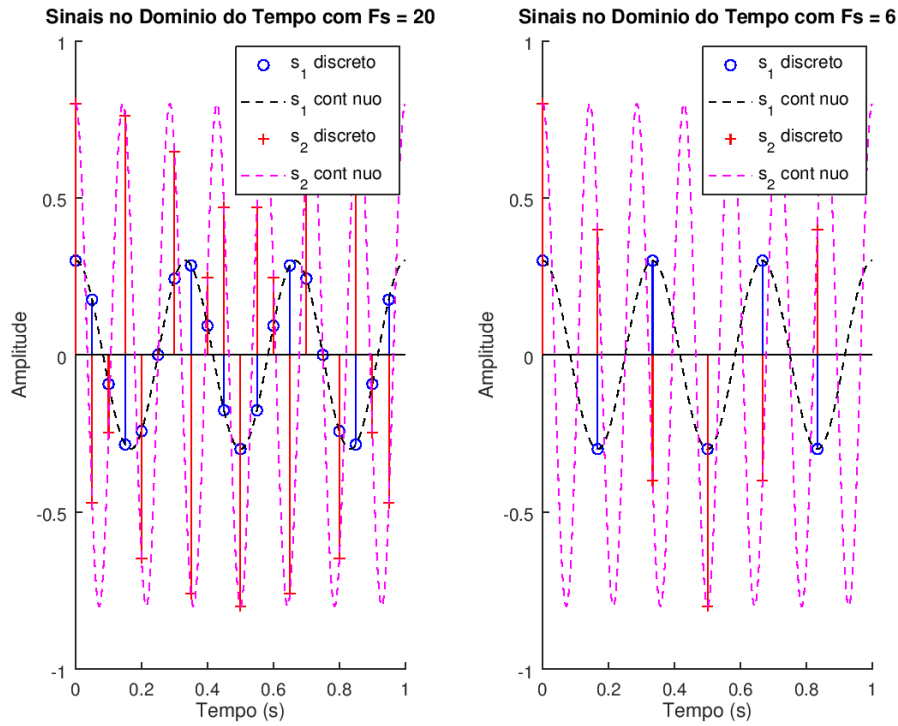


Figura 6 – Sinal no domínio do tempo com variação em F_s

A próxima figura representa os mesmos sinais porém no domínio da frequência. A frequência de amostragem foi variada em certos pontos, os quais discutiremos a seguir.

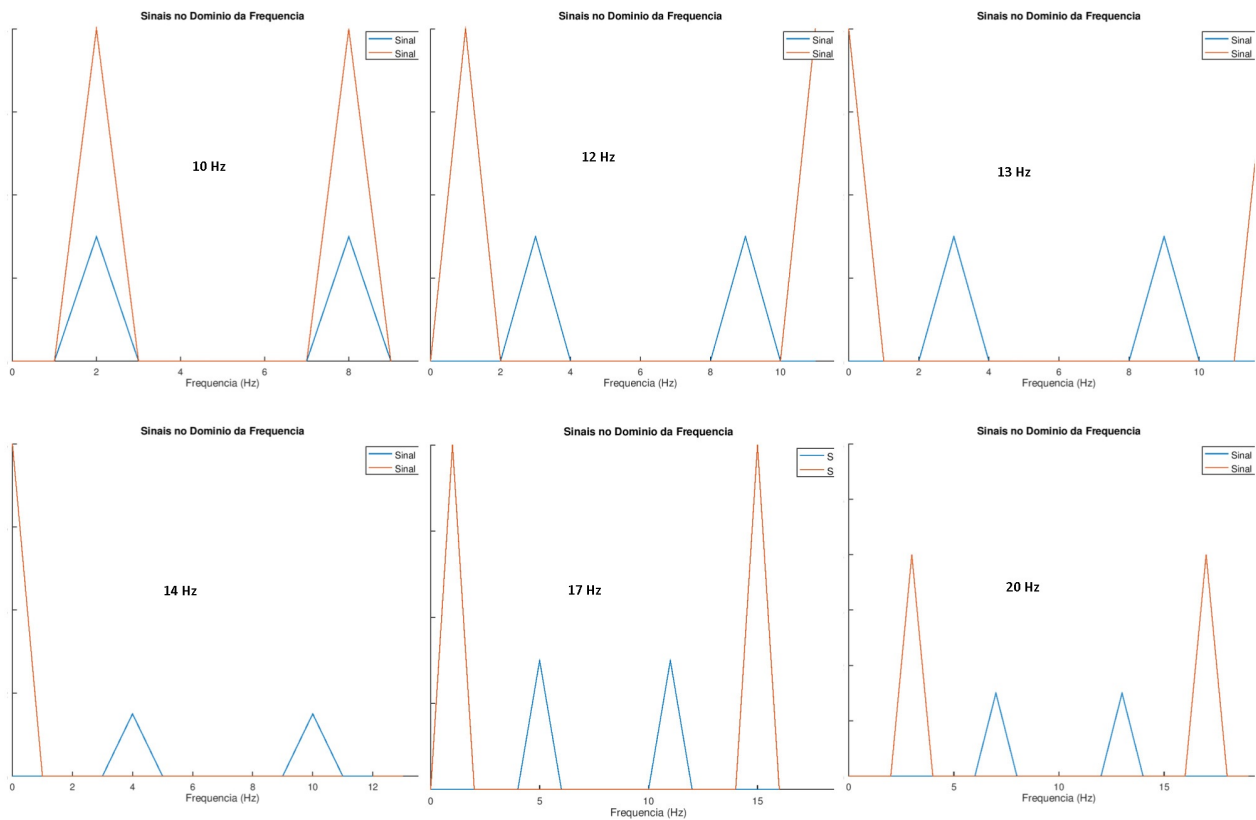


Figura 7 – Sinais no domínio da frequência com variação em F_s

Podemos observar que há sobreposição dos sinais 1 e 2 quando a frequência de amostragem é inferior ou igual à frequência de Nyquist (que seria 14 Hz). Para que não haja sobreposição, o sinal deve ser amostrado à uma frequência maior que a frequência de Nyquist, que é o que ocorre nas frequências de 17 Hz e 20 Hz.

2.4 Geração e Filtragem de um Sinal DTMF.

O objetivo deste experimento é realizar a leitura de um sinal DTMF gerado, filtrá-lo para visualizarmos suas frequências características e aplicar um filtro passa-faixa para visualizar apenas uma frequência.

Para a geração do dígito 0 do DTMF foi utilizado o software Audacity e foram colocados os parâmetros: duração de 2s e frequência de amostragem 4 KHz. A figura a seguir mostra o sinal no domínio do tempo, foram estabelecidos limites de tempo de 0,5s para que possamos observar que há mais de uma componente de frequência no sinal.

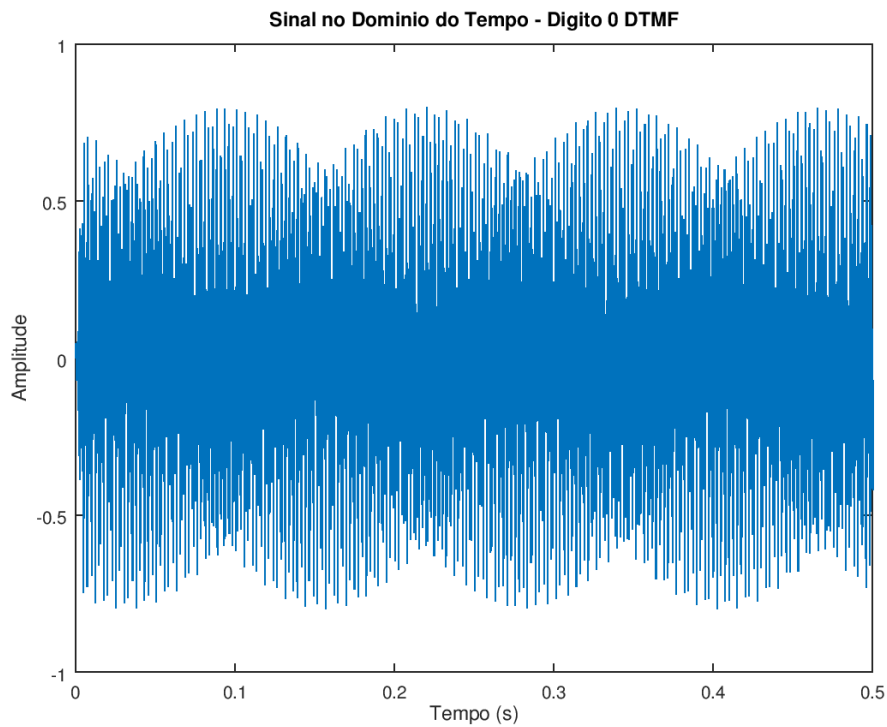


Figura 8 – DTMF - Dígito 0 no domínio do tempo

Para visualizarmos o sinal no domínio da frequência, utilizamos a função `fft()` para que as componentes de frequência sejam mostradas. O dígito 0 do DTMF é gerado com as frequências de 941 Hz e 1336 Hz. A figura a seguir mostra o sinal no domínio da frequência.

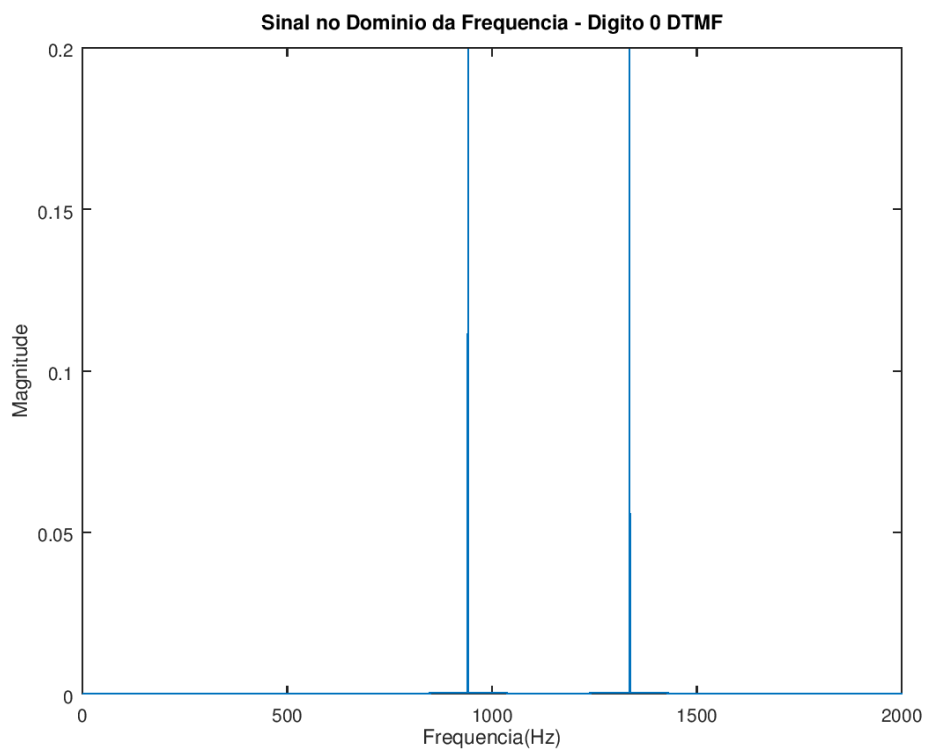


Figura 9 – DTMF - Dígito 0 no domínio da frequência

A seguir, foi criado um filtro passa-faixa para que possamos visualizar apenas a frequência da linha ou da coluna (941 Hz ou 1336 Hz, respectivamente). O filtro foi criado utilizando o método de zeros e polos, onde os zeros determinam as frequências que não devem passar e os polos as frequências permitidas para passagem. Também utilizamos a função `filter()` para realizar a filtragem do sinal DTMF. Um polo foi posicionado em cada frequência de passagem e os zeros foram posicionados ao redor da frequência oposta. A figura a seguir mostra como ficou o círculo de raio unitário para cada frequência.

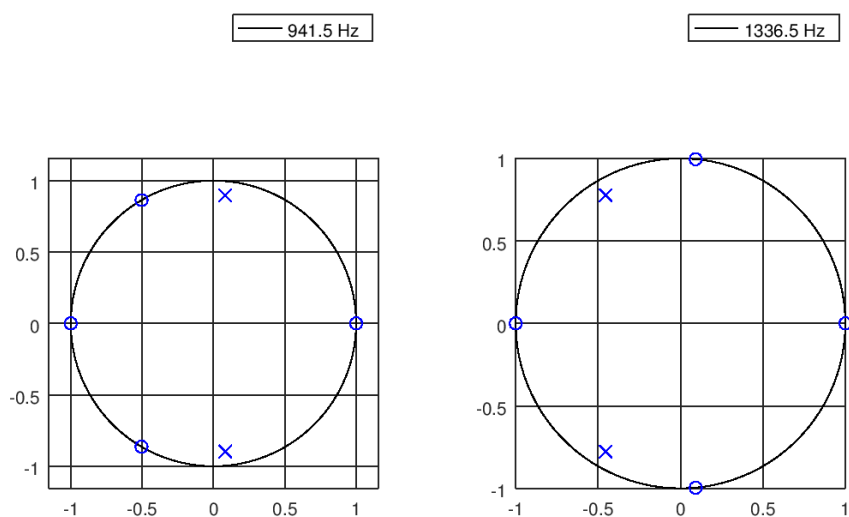


Figura 10 – Resultado dos zeros e polos no domínio complexo

A seguir, como ficou a faixa de passagem do filtro para cada frequência:

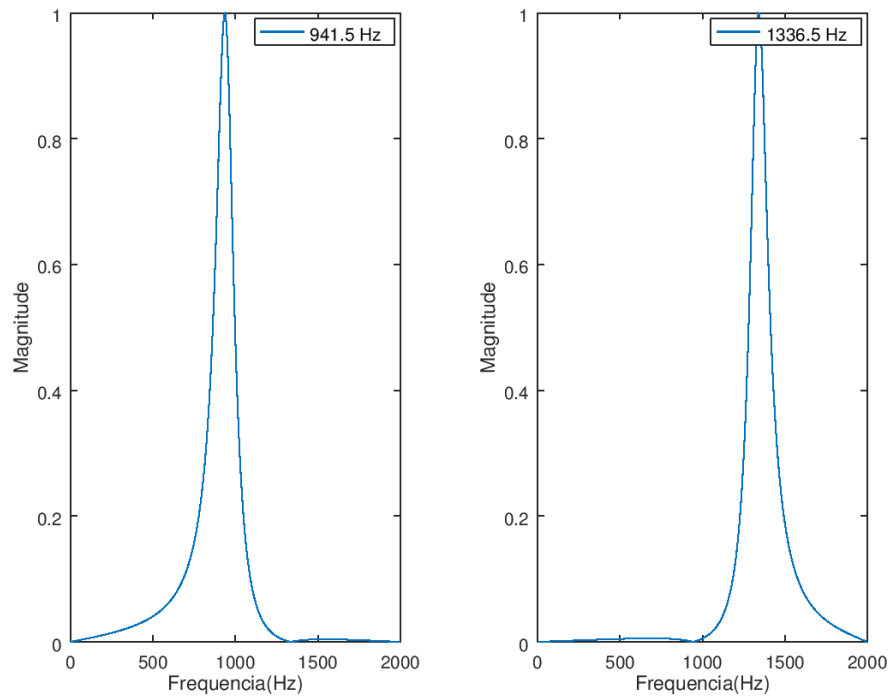


Figura 11 – Resultado do filtro projetado

Aplicamos o filtro passa-faixa ajustado para a frequência de linha e de coluna, e obtivemos o seguinte resultado. Apesar de utilizarmos apenas um valor de zero e um de polo, o filtro teve um comportamento satisfatório pois atingiu a necessidade.

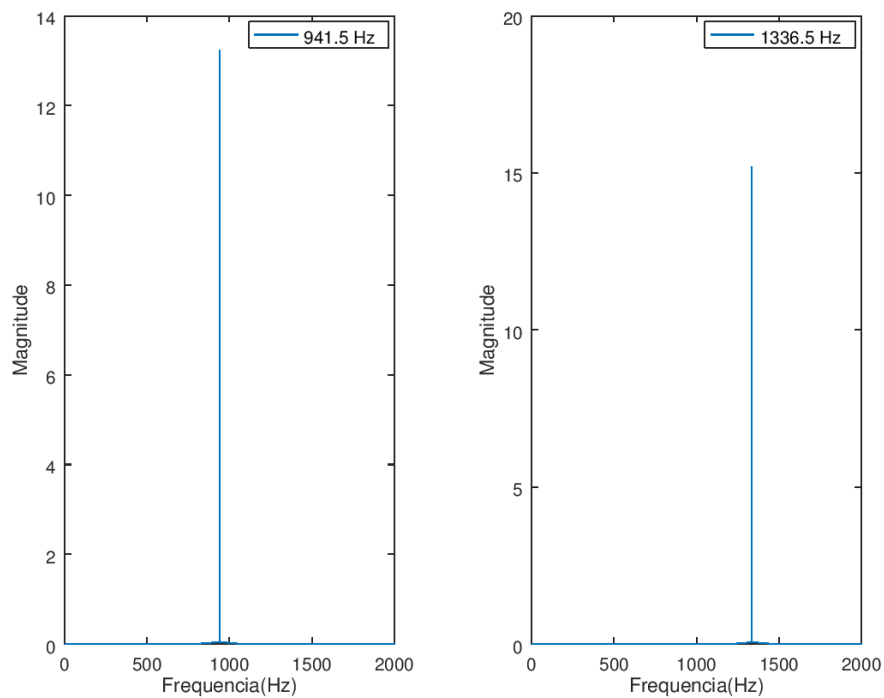


Figura 12 – O sinal DTMF após a filtragem, no domínio da frequência

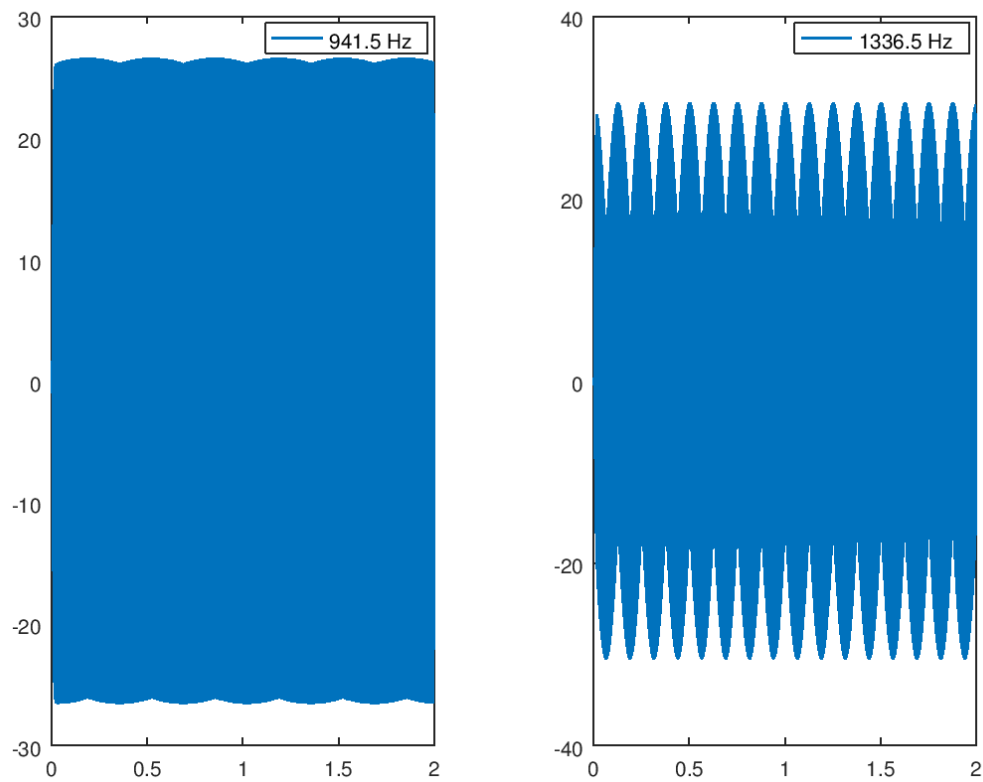


Figura 13 – O sinal DTMF após a filtragem, no domínio do tempo

3 Referências

Para todos os experimentos, foi utilizado como referência:

1. DINIZ, P. S. R., DA SILVA, E. A. B., e LIMA NETTO, S. Processamento Digital de Sinais: Projeto e Análise de Sistemas. 2. ed. Porto Alegre: Bookman, 2014. 976 p. ISBN 978-8582601235.