Processamento de Sinais em Tempo Discreto

- Prof. Dr. Samuel Lourenço Nogueira

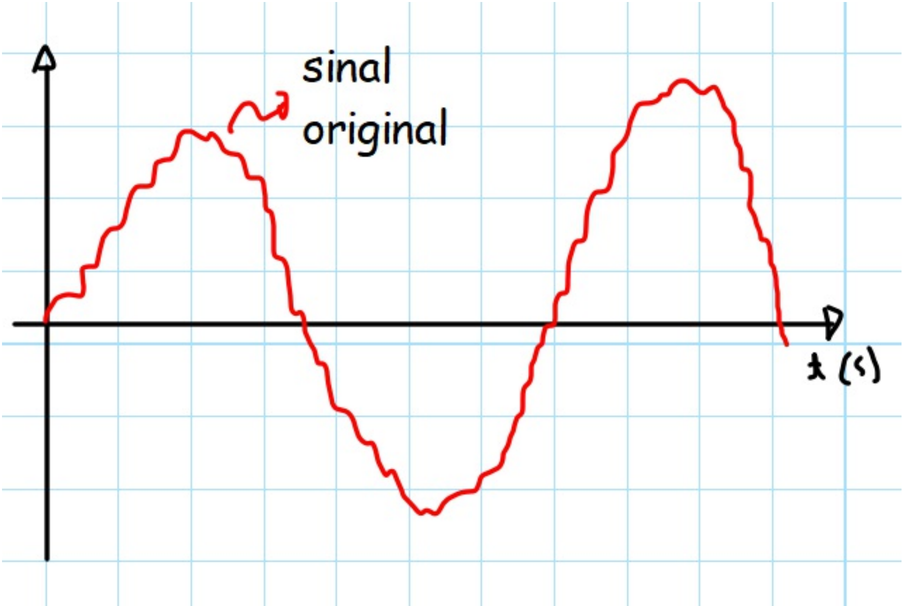
- Aulas passadas (Sobre Fourier)
  - Introdução Transformada de Fourier
  - Transformada de Discreta Fourier
  - Transformada Discreta Inversa de Fourier
  - Funcionamento DFT

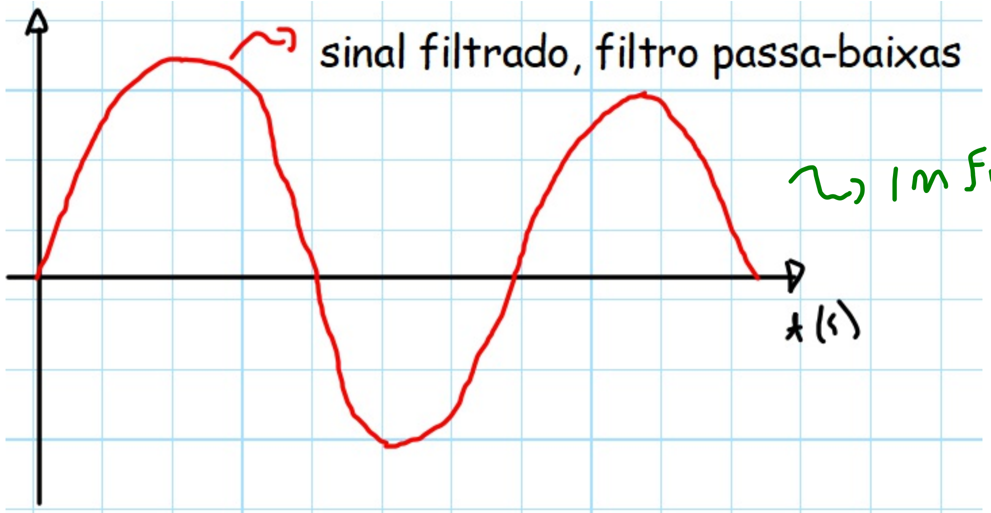
Conteúdo Programático

- Filtro de Resposta Finita ao Impulso (Filtro FIR)
- Resposta em Frequência de Filtro FIR

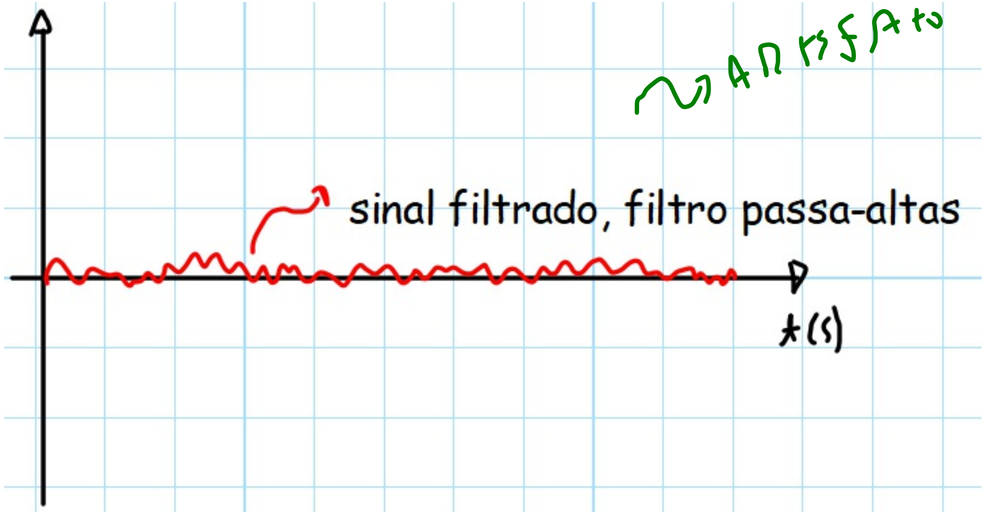
INTRODUÇÃO

Remover artefatos (informações indesejadas) dos sinais





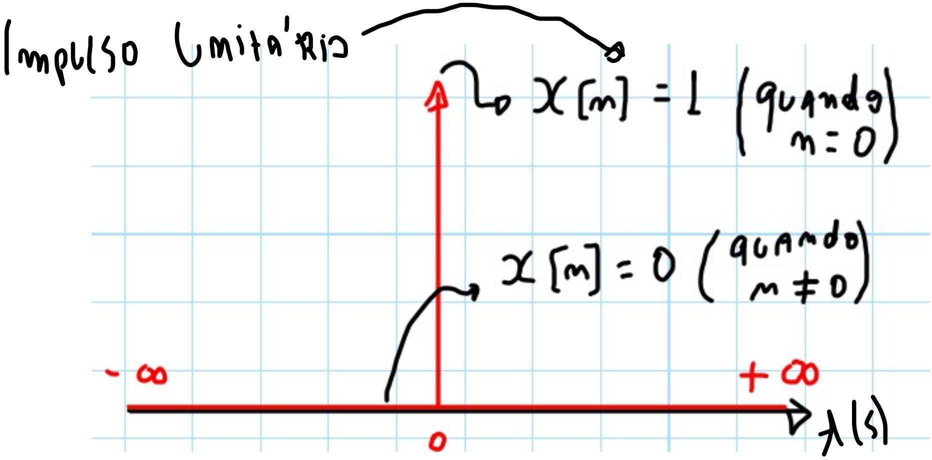
Informação



Artefato

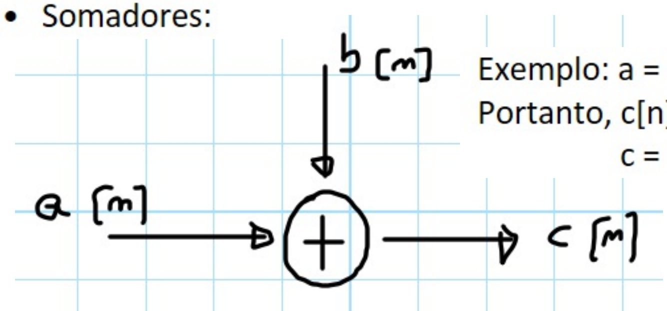
FILTRO DE RESPOSTA FINITA AO IMPULSO - FIR

Do inglês, Finite Impulse Response (FIR), constituem uma classe simples de filtros que executam grande parte do trabalho de processamento de sinais digitais.



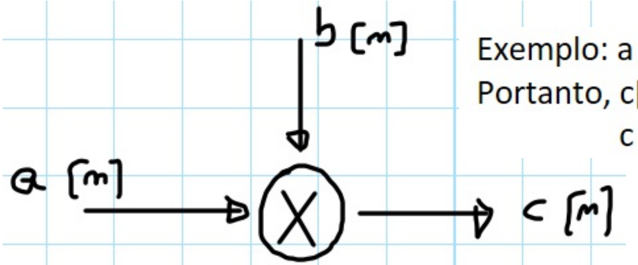
=> Componentes de um filtro FIR

- Somadores:



Exemplo: a = [1, 2, 3, 4] e b = [5, 6, 7, 8]
Portanto, c[n] = a[n] + b[n]
c = [6, 8, 10, 12]

- Multiplicadores

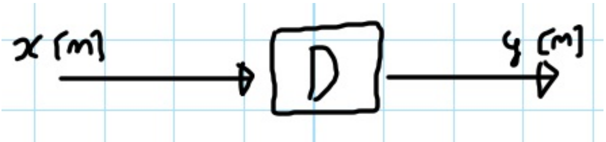


Exemplo: a = [1, 2, 3, 4] e b = [5, 6, 7, 8]
Portanto, c[n] = a[n] * b[n]
c = [5, 12, 21, 32]

FILTRO DE RESPOSTA FINITA AO IMPULSO - FIR

Do inglês, Finite Impulse Response (FIR), constituem uma classe simples de filtros que executam grande parte do trabalho de processamento de sinais digitais.

- Unidades de retardo (atraso)



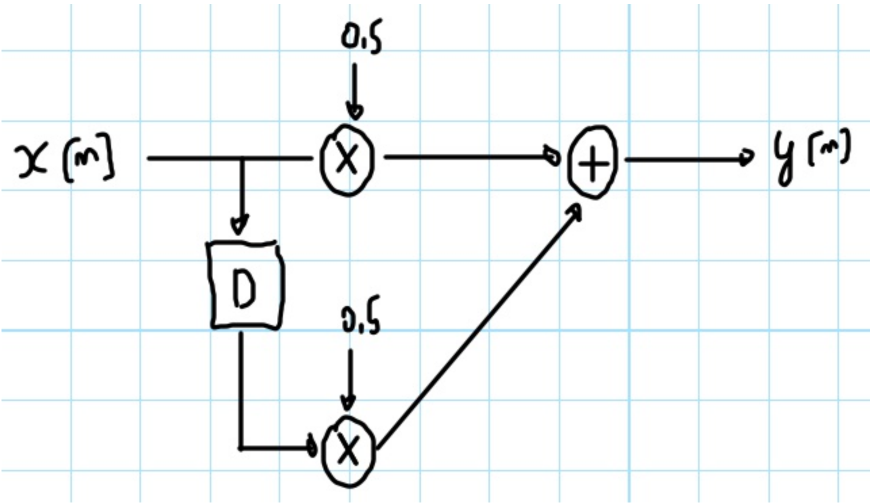
$x[-1] = 0$		$y[0] = 0$
$x[0] = 1$	$\Rightarrow$	$y[1] = 1$
$x[1] = 3$		$y[2] = 3$
$x[2] = 5$		$y[3] = 5$

$$\therefore \; y[n] = x[n-1]$$

FILTRO DE RESPOSTA FINITA AO IMPULSO - FIR

=> Estruturas de um filtro FIR

Um filtro FIR, é constituído por uma combinação de somadores, multiplicadores e unidades de atraso



Supondo x[n] = [1, 0]

$$y[0] = 0,5 \cdot \cancelto{1}{x[0]} + 0,5 * \cancelto{0}{x[-1]} = 0,5$$

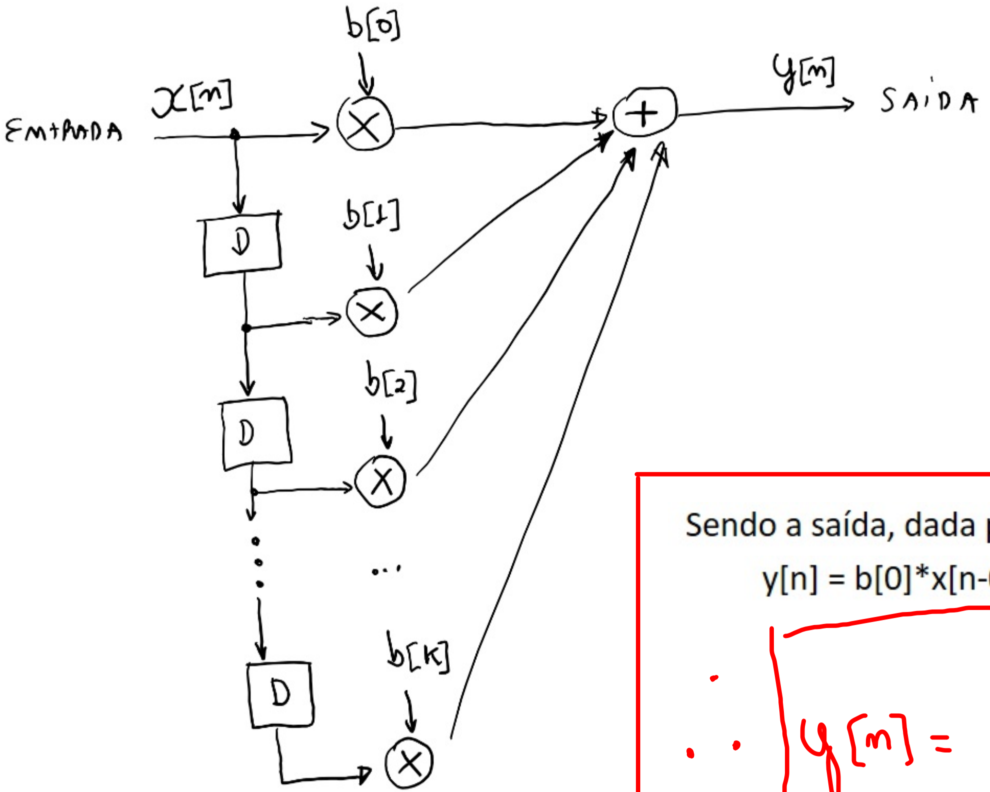
$$y[1] = 0,5 \cdot \cancelto{0}{x[1]} + 0,5 * \cancelto{1}{x[0]} = 0,5$$

$$\Downarrow$$

$$\boxed{y[n] = 0,5 * x[n] + 0,5 * x[n-1]}$$

FILTRO DE RESPOSTA FINITA AO IMPULSO - FIR

Assim, um FIR pode ser generalizado como:



Sendo o filtro FIR é conhecido pela **ordem K**, que representa a quantidade de coeficientes "b", que seria K+1 menos 1.

K

Assim um filtro com 11 coeficientes teria ordem 10.

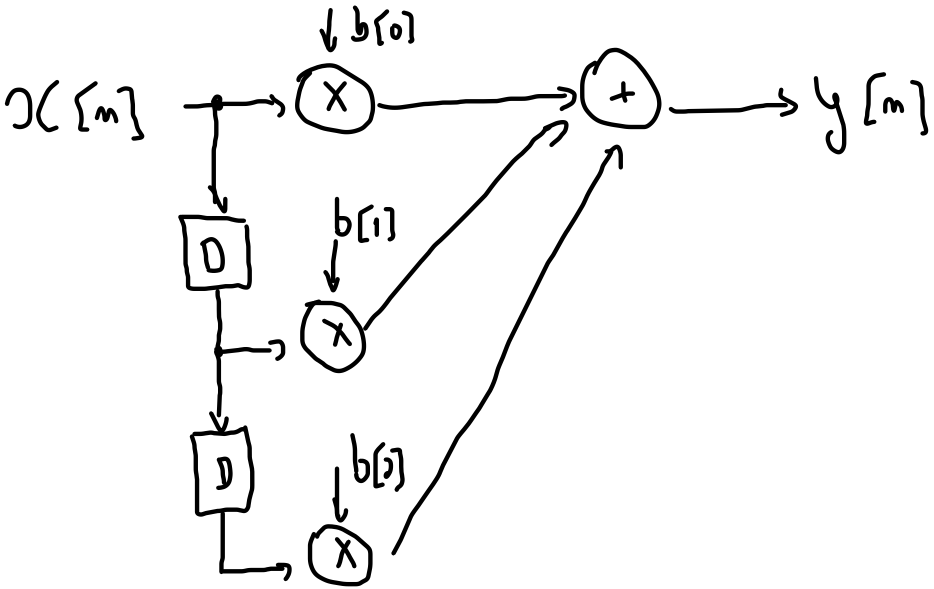
Sendo a saída, dada por:

$$y[n] = b[0]*x[n-0] + b[1]*x[n-1] + b[2]*x[n-2] + ... + b[k]*x[n-k]$$

$$\therefore \boxed{y[n] = \sum_{k=0}^{K} b[k] \cdot x[n-k]}$$ convolução!

FILTRO DE RESPOSTA FINITA AO IMPULSO - FIR

Exemplo: Dados b=[0.1, 0.2, 0.3], e x = [1, 0, 0, 2, 0, 1, 4, 3], y = ?



$x[n-0]$	$x[n-1]$	$x[n-2]$	$y[n]$
1	0	0	0,1
0	1	0	0,2
0	0	1	0,3
2	0	0	0,2
0	2	0	0,4
1	0	2	0,7
4	1	0	0,6
3	4	1	1,4
0	3	4	1,8
0	0	3	1,9

*0.1 *,2 *,3

FILTRO DE RESPOSTA FINITA AO IMPULSO - FIR

Exemplo: Dados b=[0.1, 0.2, 0.3], e x = [1, 0, 0, 2, 0, 1, 4, 3], y = ?

$$y[\,] = \begin{array}{ccc} x[n-2] & x[n-1] & x[n-0] \\ \end{array}$$

$$y[\,] = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 2 \\ 0 & 2 & 0 \\ 2 & 0 & 1 \\ 0 & 1 & 4 \\ 1 & 4 & 3 \\ 4 & 3 & 0 \\ 3 & 0 & 0 \end{bmatrix} \times \begin{array}{c} \\ \end{array}\begin{bmatrix} 0.3 \\ 0.2 \\ 0.1 \end{bmatrix} \begin{array}{l} \to b[2] \\ \to b[1] \\ \to b[0] \end{array}$$

é o mesmo que

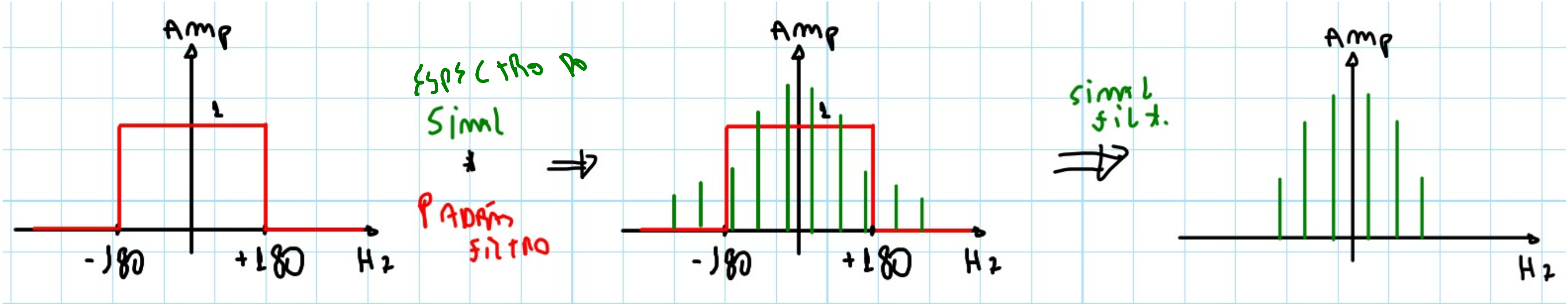
$$\boxed{y[n] = \sum_{k=0}^{K} b[k] \cdot x[n-k]}$$

FILTRO DE RESPOSTA FINITA AO IMPULSO - FIR

Os filtros FIR podem implementar diferentes funções mudando seus coeficientes b[k] e a ordem K. Podendo interferir somente em determinadas frequências do

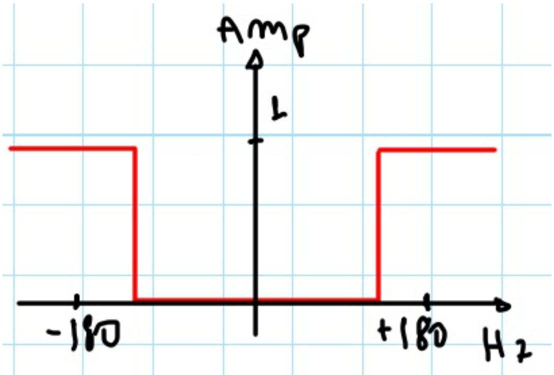
Os principais tipos são:

Lowpass (passa-baixa)

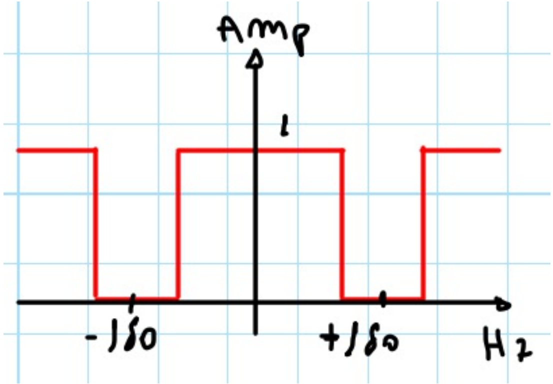


FILTRO DE RESPOSTA FINITA AO IMPULSO - FIR

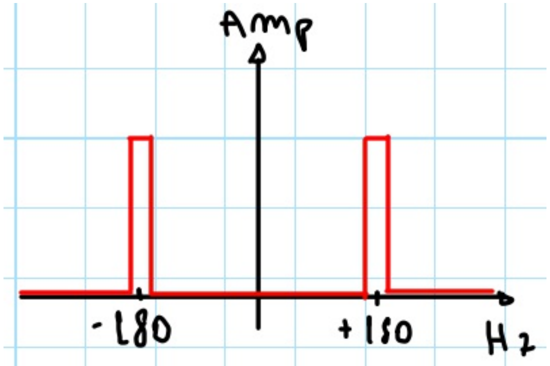
- Highpass (passa-alta)



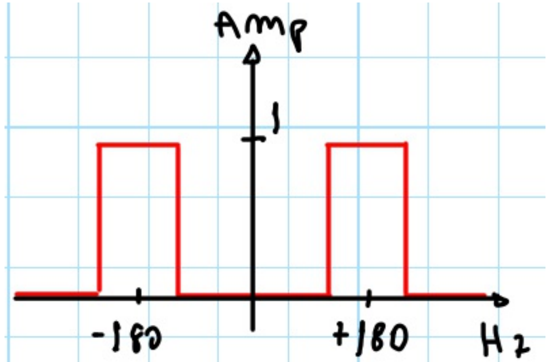
- Bandstop (rejeita-faixa)



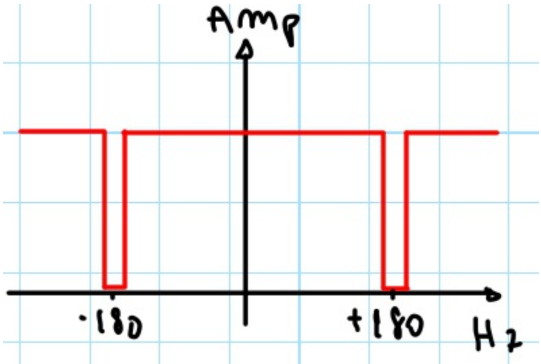
- Laser-line bandpass (passa-faixa estreito)



- Bandpass (passa-faixa)



- Notch (rejeita-faixa estreito)



No Matlab e Octave, a função **fir1** projeta os coeficientes **b** para os filtros acima.

>> Topico6Exemplo1.m

RESPOSTA EM FREQUÊNCIA DE FILTROS FIR