SI100B Tutorial: Signal Denoise

Qixuan ZAI, Jianwen LUO

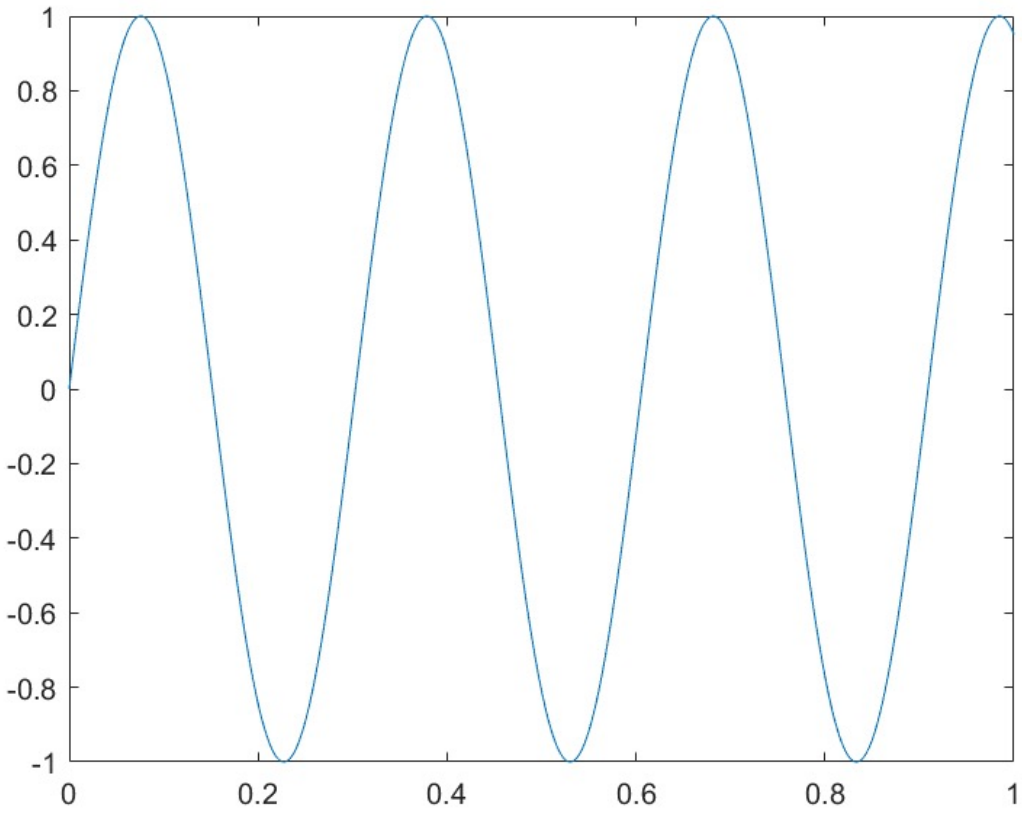
Nov. 2021

Table of Contents

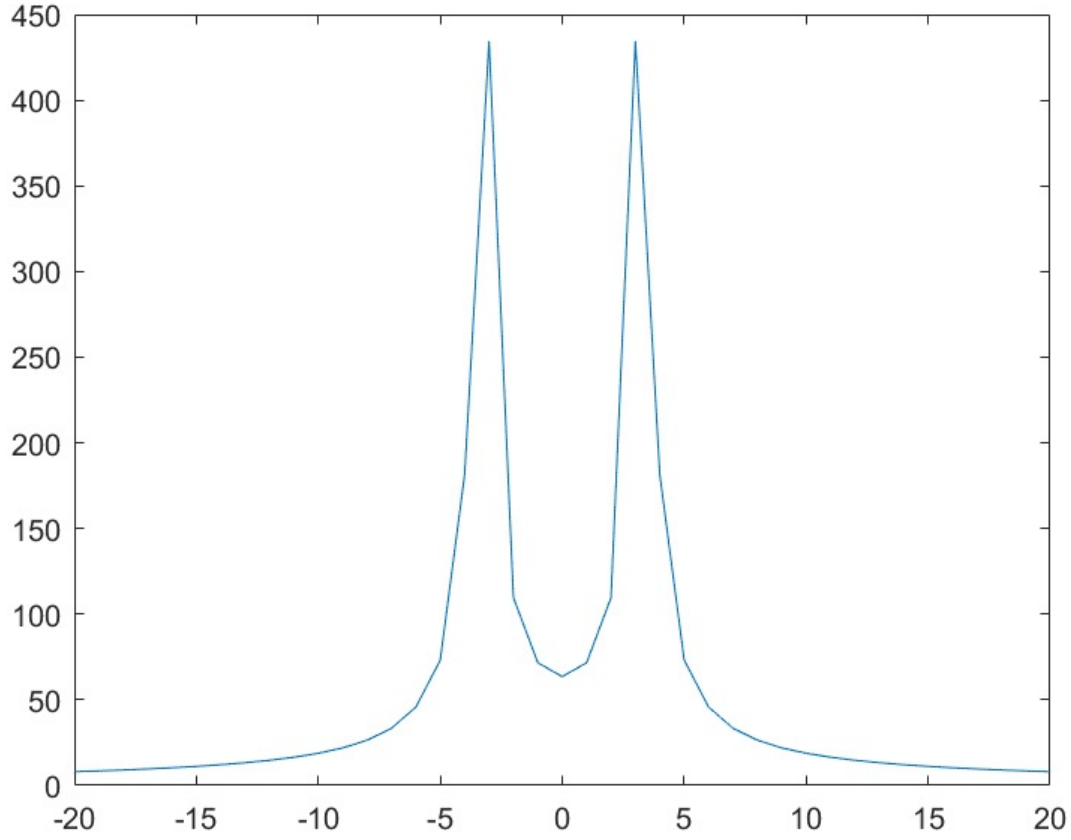
Frequency Domain

A New Viewpoint to Signals

Case Study: Sine

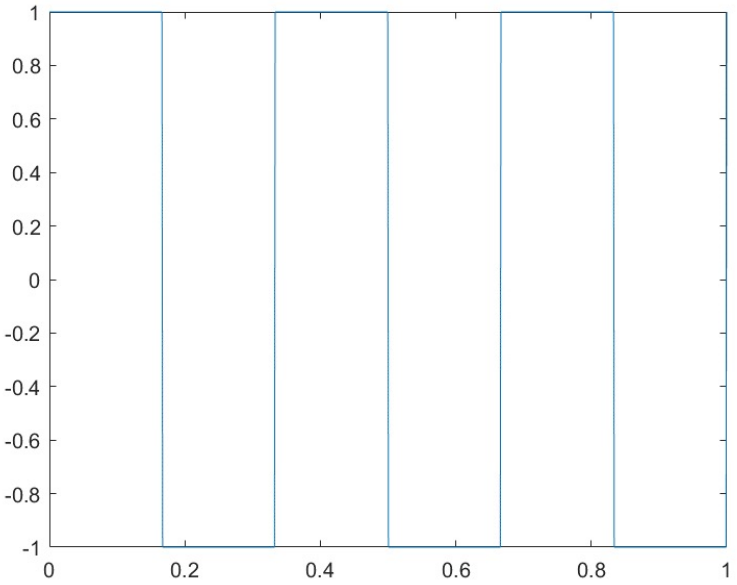


Sine in Time Domain

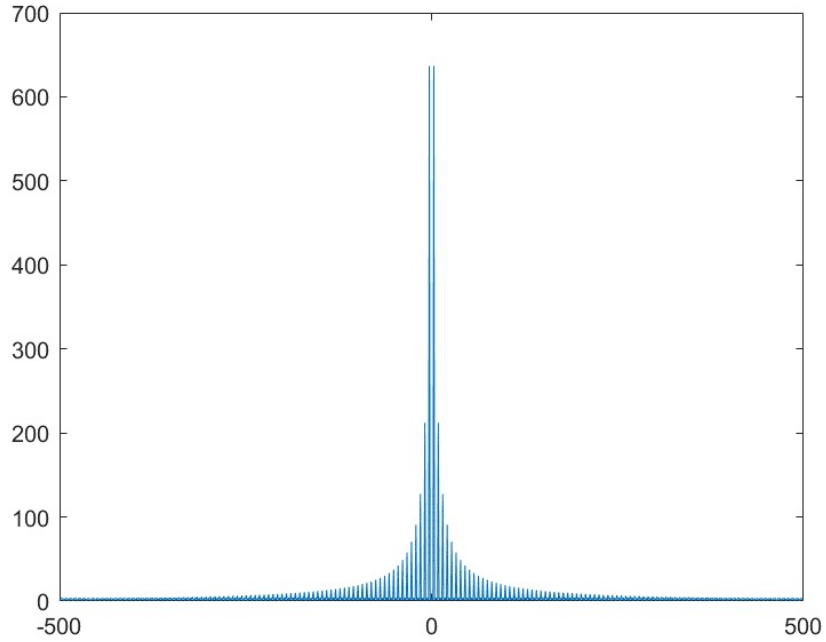


Sine in Freq Domain

Case Study: Square Wave

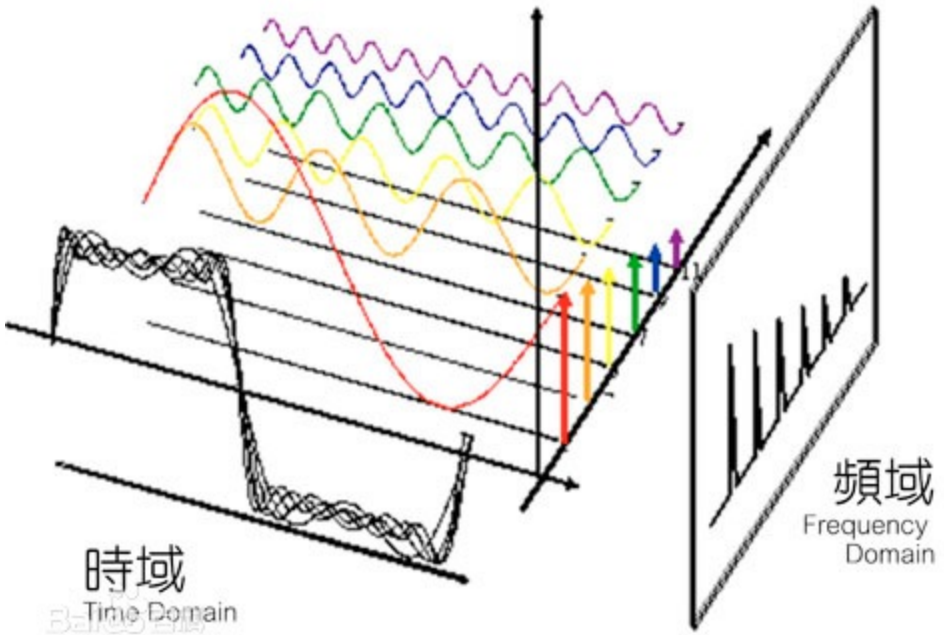


Square Wave in Time Domain



Square Wave in Freq Domain

Cont’d: Square Wave But as Sum



Still Frequency Domain

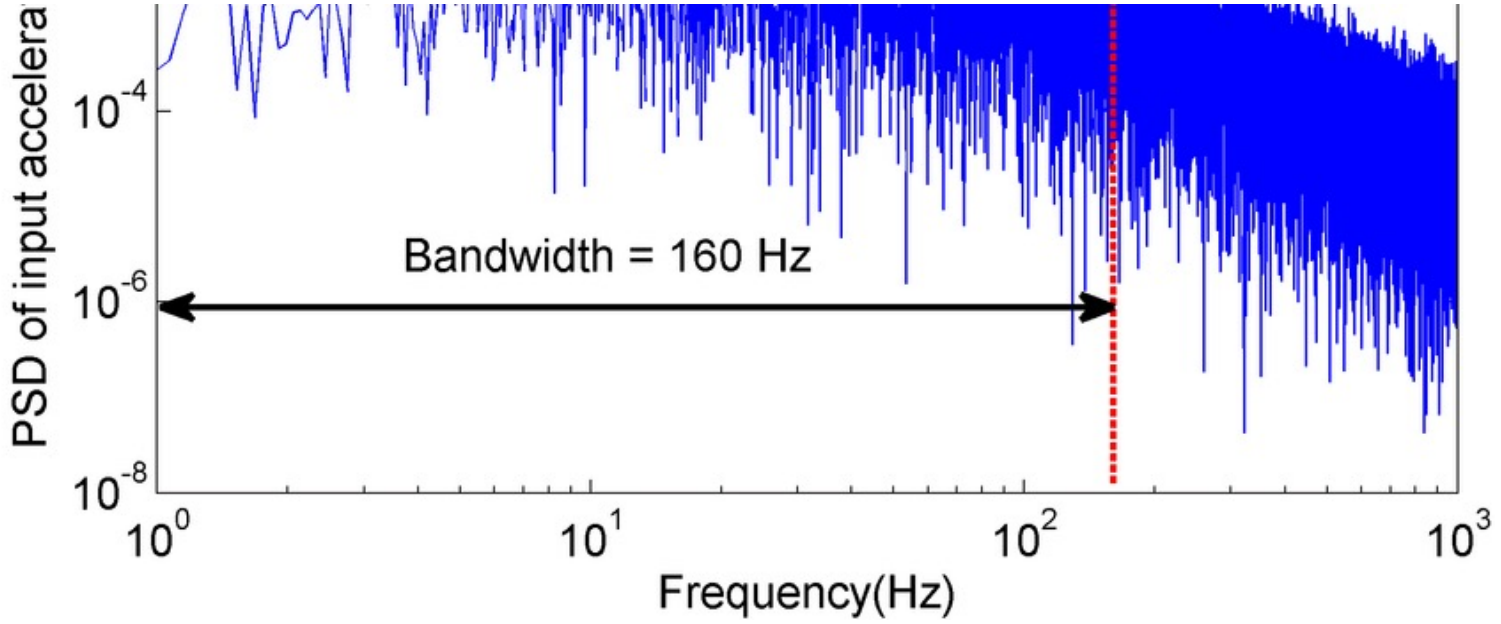
A New Viewpoint to Noises

Types of Noise

- Types of noise in communication system:
- White or colored?
- Bandlimited or Low pass or High pass?
- AWGN: Additive White Gaussian Noise
- https://en.wikipedia.org/wiki/Additive_white_Gaussian_noise
- Reference: Principles Of Communications, Chapter7: RANOM SIGNALS AND NOISE

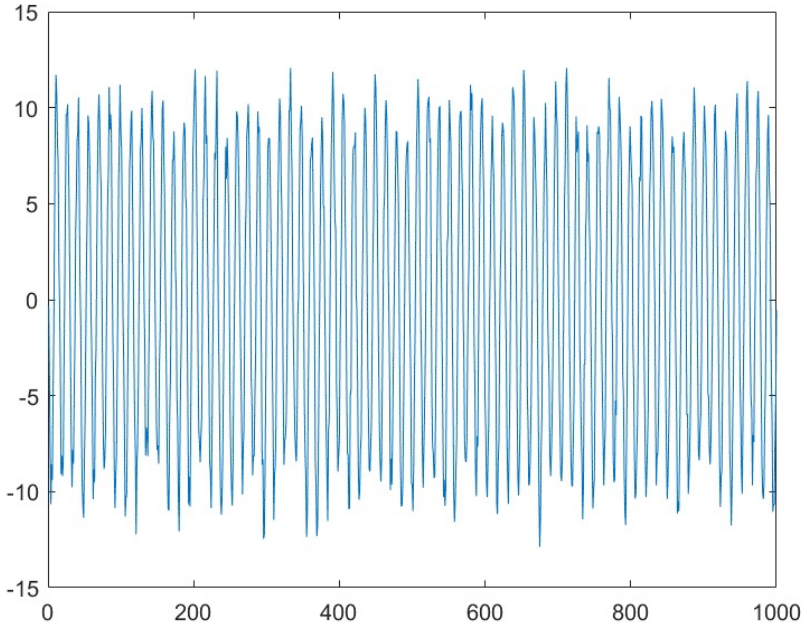
Cont’d

- Only consider colored and bandlimited noise !!!

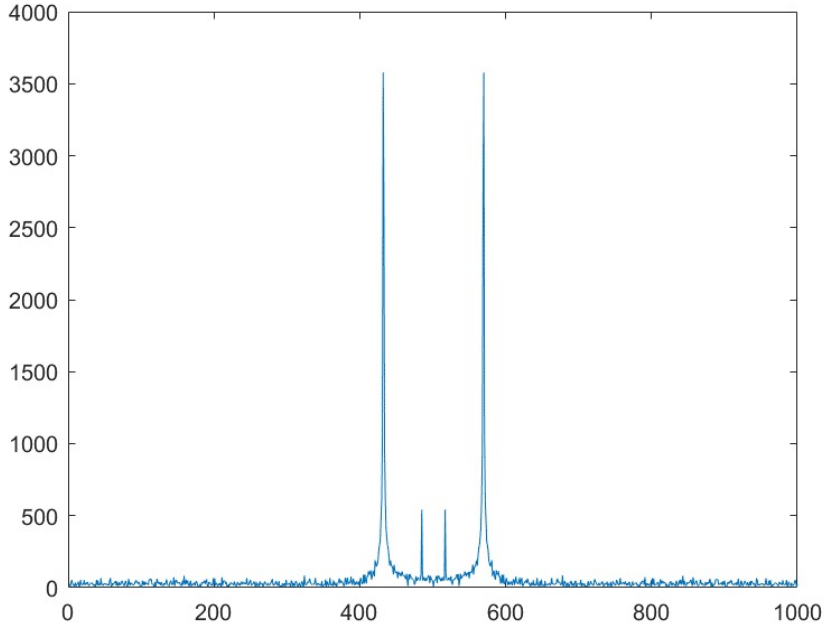


A First Sight: Separated Frequency (Band?)

```
ys = 10 * sin(100000 * ts) + sin(100 * ts) + randn([1 length(ts)]);
```

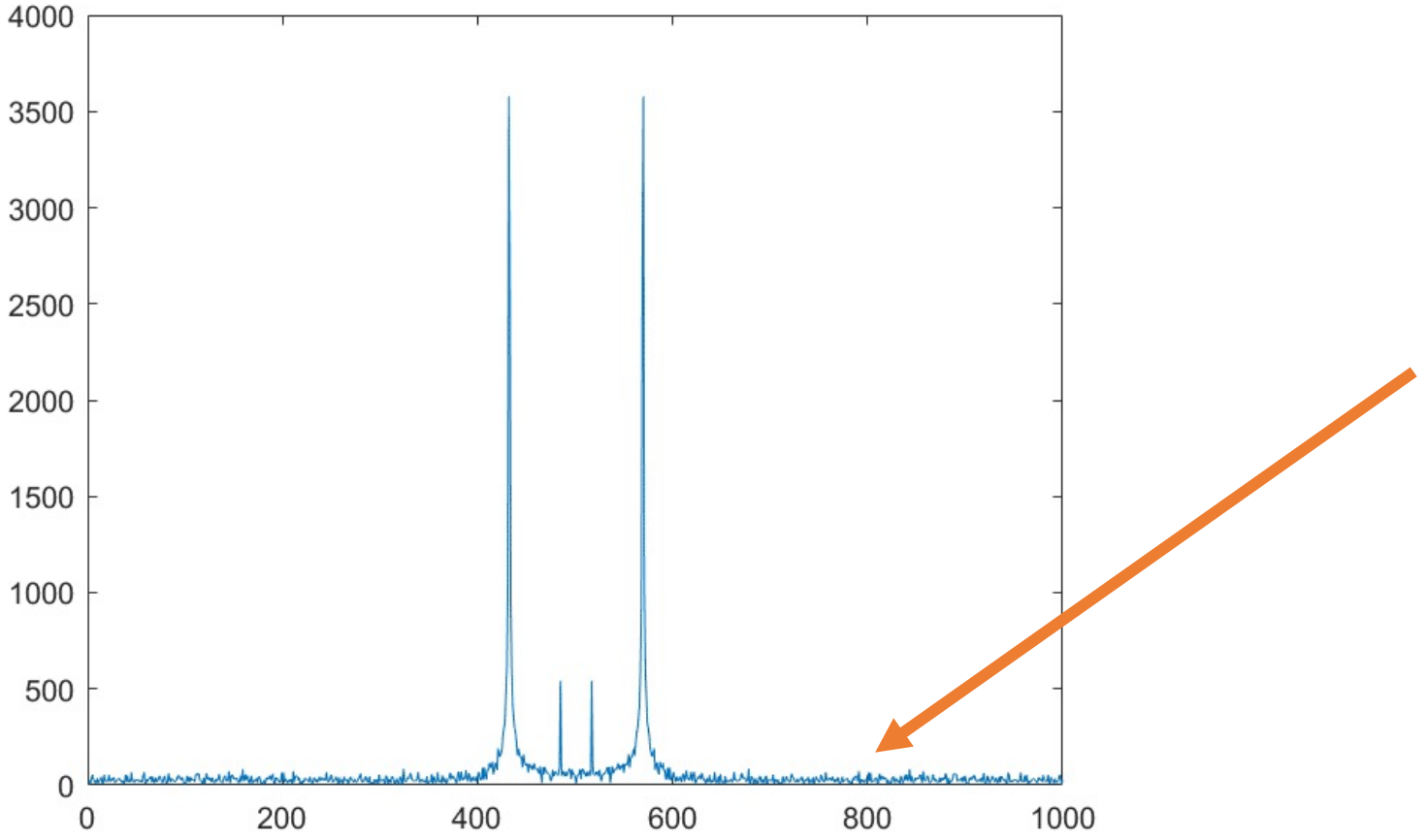


Time Domain



Frequency Domain

SNR Effect: Have You Noticed the Glitches?



Frequency Domain

Discrete Fourier Transform

Time Domain to Freq. Domain

The Formula: Just Have a Look

$$\hat{u}_k = h \sum_{j=1}^{N} e^{-ikx_j} u_j, \; k = -N/2 + 1, \ldots, N/2,$$

$$u_j = \frac{1}{2\pi} \sum_{k=-N/2+1}^{N/2} e^{ikx_j} \hat{u}_k, \; j = 1, \ldots, N,$$

Number of Samples: N
Sampling Frequency: f_s
Sampling Duration: (0 to) t_1
Sampling Period: $T = \frac{1}{f_s} = \frac{t_1}{N}$
Time Indexing: $x_n = x(nT) = x(n\frac{t_1}{N})$
Freq. Indexing: $X_n = X(2\pi\frac{f_s}{N})$

Fast Fourier Transform

Just a Way You Compute DFT

THIS is all You Need (for DFT)!

- X = fft(xs): DFT with Freq. $[0, f_s)$
- X_shift = fftshift(X): DFT result shifted to Freq. $[-f_s/2, f_s/2)$
- abs(X): Amplitude Spectral
- angle(X): Phase Spectral
- ----------
- xs = ifft(X): Travel from Freq. Domain back to Time Domain!
- X = ifftshift(X_shift): shifted DFT result back as un-shifted

FIR Filters and Convolution

Know How and Better Know What

Why Filter?

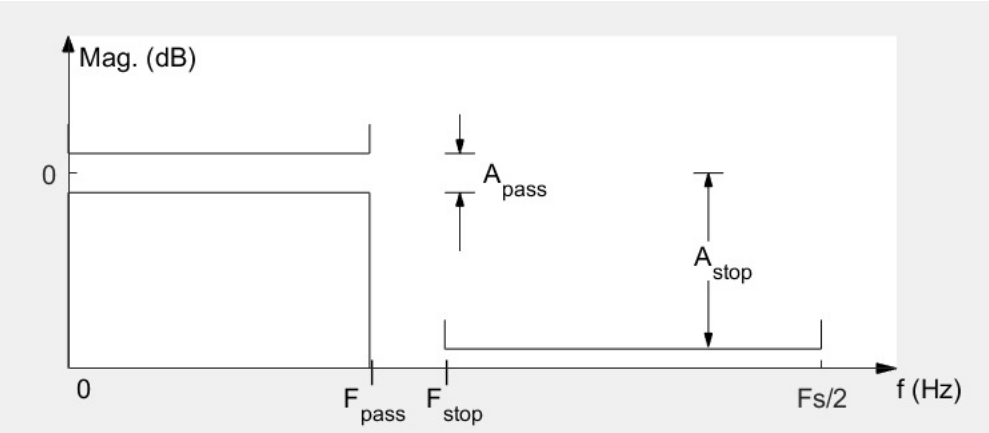
- Why not FFT and Truncate and IFFT?
  - Require the whole signal known
  - Lead to Large Latency
  - Require Large Storage
  - => Impractical in real-time applications
- Why FIR?
  - Maths. Stuffs, e.g. Stability
  - Practical Latency
  - East to implement

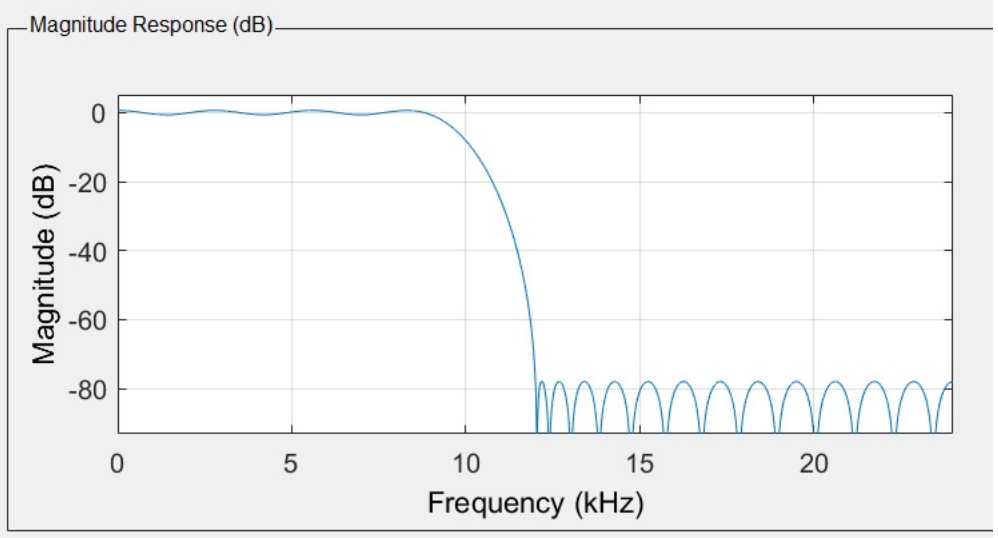
(Discrete) Convolution?

- Equivalent to Linear Time-Invariant System
  - In one word:
    - $Y\{\sum_m x[m]\delta[n-m]\}[n] = \sum_m x[m]h[n-m] = \sum_m h[m]x[n-m]$
- Be careful about Cyclic Convolution!
  - Discrete Convolution is performed with Z indexing
  - … but we have only finite samples so we only have cyclic convolution
  - … just by default (fortunately)!
  - ----------
  - Remember Zero Padding before calling ys = conv(hs, xs)
  - … pad both hs and xs to [length(xs) + length(hs) – 1]

So what is FIR Filter?

- Do you remember $h[n]$ (in last page)?
- FIR filter is an LTI system $h[n]$ with a finite support.
  - i.e. only finite entries are non-zero
- How to apply?
  - Just perform convolution.
- What to design?: Mag. Resp.

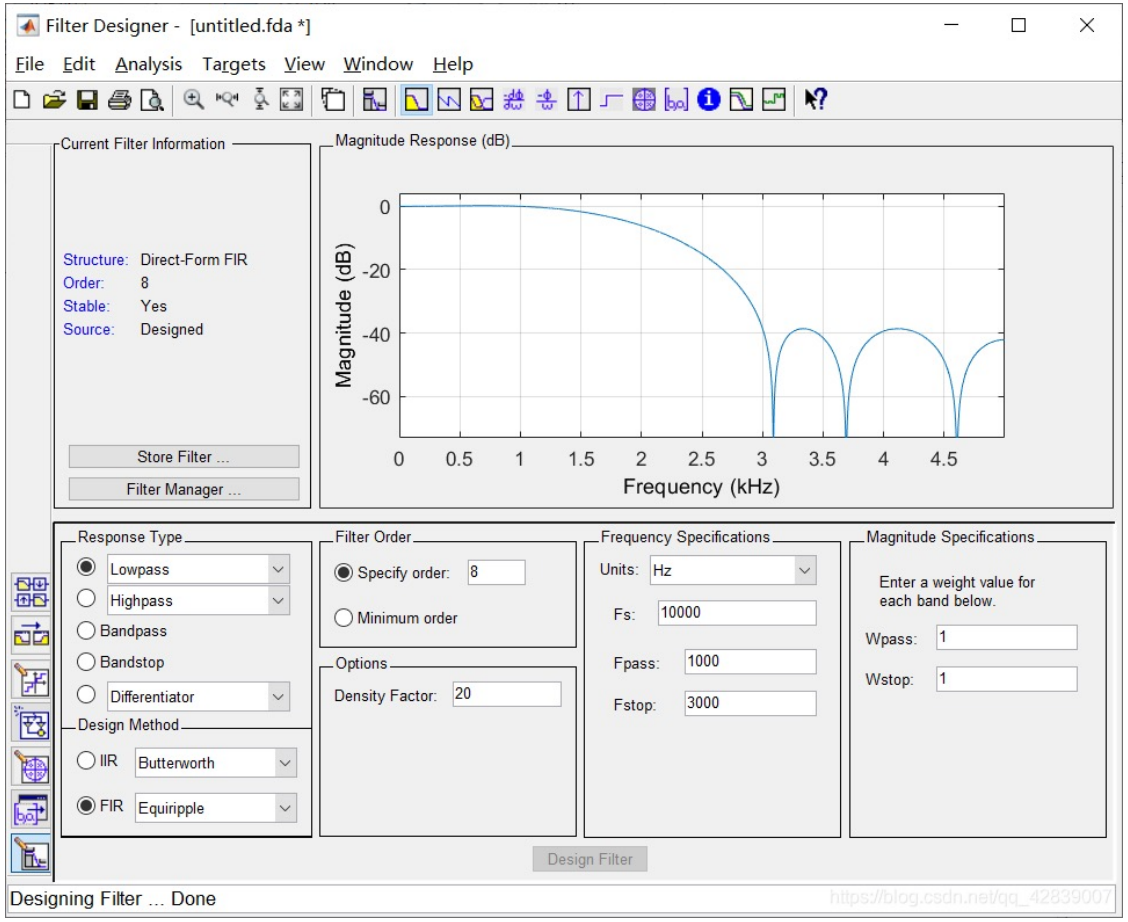




MATLAB Signal Processing Toolbox

Easy Usage with `fdatool`

fdatool is ALL You Need



Good Luck Have Fun!

Q&A