Table of Content

1. Preliminary
2. Short Time Fourier Transformation: Analysis
3. A brief introduction of Finite Impose Response–window method
4. Short Time Fourier Transformation: Synthesis–Constant Overlapping Add
5. Pre-application Explanation
6. Simple FIR Filter Application
7. Reference

# @Preliminary

• **noncyclic-Convolution:** Given two sequence **x[n]** with length $N$ and **h[n]** with length M, the convolution of the two sequence

$$y[n] = \sum_{m=-\infty}^{\infty} x[m]h[n-m]$$

the length of which is $L = N + M - 1$ for each fixed n, the window $h[m]$ has the right end point located at n.
• **noncyclic-Convolution thereom(proof omitted):**

$$F(xv) = F(x) * F(v) \quad and \quad F(x * v) = F(x)F(v)$$

• **cyclic-Convolution:** Given **X** to be the DFT of**x**, and **Y** to be the DFT of**y**, and $Z[k] = X[k]. * Y[k]$, where k=0,1....,K-1, the DFT of **Z** will be

$$z[k] = x[k] \circledast y[k] = \sum_{l=0}^{l=K-1} x[l]y[n-l-cL]$$

where c is a constant. **proof:**

$$\begin{aligned} Z[k] = X[k]. * Y[k] &= \sum_{m=0}^{L-1} x[m]e^{-2i\pi km/L} \sum_{l=0}^{L-1} y[l]e^{-2i\pi kl/L} \\ &= \sum_{m=0}^{L-1} \sum_{l=0}^{L-1} x[m]y[l]e^{-2i\pi k(l+m)/L} \end{aligned}$$

then,

$$\begin{aligned} F^{-1}(Z)[n] =& 1/L \sum_{k=0}^{L-1} (\sum_{m=0}^{L-1} \sum_{l=0}^{L-1} x[m]y[l]e^{-2i\pi k(l+m)/L}) e^{2i\pi nk/L}) \\ =& 1/L \sum_{m=0}^{L-1} \sum_{l=0}^{L-1} x[m]y[l]) \sum_{k=0}^{L-1} e^{-2i\pi k(l+m-n)/L} \end{aligned}$$

noticed when (n-l-m)mod L=0 the last summation=L, otherwise 0; and $n - l - m = 2 - 2L, ...0, ..., L - 1$. Therefore, there are only 2 terms are not zero, that is $l = n - m$ and $l = n - m + L$. Thus

$$z[n] = \sum_{l=0}^{l=K-1} \sum_{m=0}^{m=K-1} x[m]y[l](L\delta_{l+m-n=cL}) = \sum_{m=0}^{m=K-1} x[m]y[n-m-cL]$$

, where c is constant and $l$ and decided by m and n
• **Time-Frequency Duality:**

$$\begin{aligned} F\{x[n]w[n]\} =& \frac{1}{2\pi} X(\omega) \circledast W(\omega) \\ where \quad X(\omega) \circledast W(\omega)[n] =& \int_{2\pi} x(u)w[n-u]du \end{aligned}$$

## @STFT: Analysis

There are two interpretation: Fourier view or Filtering view. The student presents only fourier view.
•**Window function**:

$$w(n) = \begin{cases} something & \text{if } 0 \leq n \leq L-1 \\ 0 & \text{otherwise} \end{cases}$$

•**discrete STFT:**

$$x[n,\omega)_{\omega=\frac{2i\pi k}{N}} = \sum_{m=-\infty}^{\infty} x[m]w[n-m]e^{-i\omega m}$$

the endpoint of the window located at time n. The window "sees" only the section that's selected. illustrated as below: Let the length of the signal $x[n]$ to be N, the length of the window to be $N_w$, the temporal decimator (hop size) to be L

$$x[nL,\omega)_{\omega=\frac{2i\pi k}{N_w}} = \sum_{m=0}^{N_w-1} x[m]w[nL-m]e^{-i\omega m}$$

•**Invertibility:**
recall that frequency $f = \frac{2\pi}{N}$. And stft can be written as

$$\begin{aligned} x[n,\omega)_{\omega=\frac{2i\pi k}{N}} &= \sum_{m=-\infty}^{\infty} x[m]w[n-m]e^{-i\omega m} \\ &= F(x[m]w[n-m]) \\ &= F(x[m]) * F(w[n-m]) \\ &= F(x[m]) * F(w[-m])e^{-i\omega n} \\ &= X(\omega) * W(-\omega)e^{-i\omega n} \\ (byconvthereom) &= \frac{1}{2\pi}\int_{2\pi} X(\theta+\omega)W(\theta)e^{i\theta n}d\theta \end{aligned}$$

which is the convolution in frequency domain, or between the frequency response. if the bandwidth of the window is less than the sampling frequency of the whole signal, then there will be frequencies that does not pass through any portion of the filter, thus non invertible.

from a fourier point of view, if the hop size is greater than the window length, there will be signals that do not pass through the window,which means the signal can't be reconstruct. Thus, Hop size should be less or equal to window size.

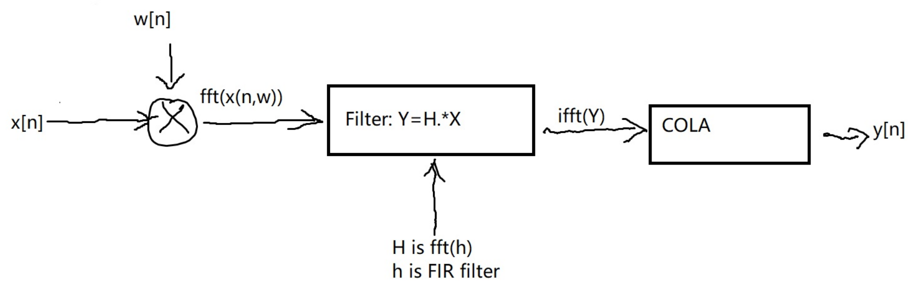
## @A Brief Summary of Finite Impulse Response–window method

This section is not intended to present the matematical core of FIR, instead, the student use FIR merely as a tool to demonstrate how analysis and synthesis of discrete-STFT work.

•**objective of a filter:** let the wanted frequencies pass or reject unwanted frequencies.
•**Filter Design:** Since a finite linear phase filter is desired, a window function w[n] is taken to multiply the ideal frequency response in time, i.e. $h[k] = w[n]h_{id}[n]$. This is equivalent to convolve $W(\omega)$ and $H(\omega)$ in frequency domian, where $W$ and $H$ are the DFTs.
the proof involves least-square approx using DFT which is omitted here, however the illustration is drawn bellow:
•**How does a filter work?**: (citation) Given a filter $h[k]$ and a signal $x[k]$, the output signal is given by:

$$y[n] = x[n] * h[n] \longleftrightarrow Y(\omega) = X(\omega). * H(\omega)$$

**•illustration of filtering procedure:**




## @ STFT Synthesis–constant overlapping add

recall that, according to Fourier view, a window function is first used to decimate the signal, therefore, define each block of segmented signal $x_m(n)$ can be expressed as:

$$x_m(n) =: x(n)w(n - mL)$$

where L is the hop size. and because $w(n)$ is useally symmetric, $w(n - mL) = w(L - n)$

**•Constant overlapping add:** with overlapping, one observation is:

$$\begin{aligned} x(n) &= \sum_m x_m(n) \\ &= \sum_m x(n)w(mL - n) \\ &= x(n) \underbrace{\sum_m w(mL - n)}_{=1(COLA)} \end{aligned}$$

the sum of all window functions, i.e. $g(n) = \sum_m w(mL - n) = 1$ should hold for x(n) to be perfectly reconstructed. The illustration is attached below:

**•Constant overlapping add for STFT:** given the DFT of the short time section is:

$$\begin{aligned} \because X_m(\omega) &= \sum_n x(n)w(n - mL)e^{-i\omega n/N} \\ \therefore X(\omega) &= \sum_m X_m(\omega) \\ &= \sum_n x(n)e^{-i\omega n/N} \underbrace{\sum_m w(n - mL)}_{=1} = \underbrace{DTFT\{x(n)\}}_{Discrete-timeFT} \end{aligned}$$

therefore the reconstruction of STFT is simply taking the $DTFT^{-1}$ of $X\{\omega\}$

**•Remarks1:**

As have mentioned in FIR filtering section above, the following two approaches are the same:

1. $y_m[n] = x_m[n] * h[n] \to DFT \to Y_m[n] \to COLA \to IDFT \to y[n]$

2. $h[n], x_m[n] \rightarrow DFT \rightarrow H(\omega), W(\omega) \rightarrow IDFT(Y(\omega) = H. * W) \rightarrow COLA \rightarrow y[n]$

the two approaches are both viable, however the second method is more preferable and usually how people implement the STFT algorithms, due to the fact that it is faster and can be implemented in real-time. The fflops and justification for real-time are not difficult to show(omitted here).

**•Remarks2:**
the same overlap rule follows the second approach:

$$y[n] = \sum_m y_m[n] = \sum_m IDTFT_n\{Y_m\} = \sum_m x_m * h_m$$

which is the similar answer to the first approach

## @ Some Explanation for the Application

**•Objectives:** this project is intended to design a simple finite impulse response filter to filter out unwanted frequencies in single-channel audio file. The analysis and synthesis is done with short time fourier transformation.

**•procedures:**
1. To start with, I have a 6-second audio who contains a persistant tone in the background and a couple of octives that were played by a piano. The note is C.

2. using the matlab built in to plot the time-frequency spectrum of the audio. one can either look at the frequency intesity at a fixed time, or the intensity of a certain frequency across time. Decibels(dB) calculated as $log_{10}\frac{\text{engergy}}{\text{reference energy}}$, the smaller it is the lower energy of the signal, the blue-er it is on the graph

3. It can be observed that the constant tone in the background is lightened up the whole duration, and it has relatively high energy. It can also be observed that on the spectrum different fundamental frequencies are lighted up in time at for the octives.

4. some parameter:
sampling frequency: 44100Hz
total duration/Length: 7s/44100*7
window:Length=512 hann window with 50% overlap
filter: length=256 highpass filter with passband frequency [2000Hz,$+\infty$)
Length of fourier transformation: NFFT=1024 to avoid time domain aliasing

**•some more justifications:**
1. with filter length=256 and window length=512, I choose the length of my short time fourier block NFFT=1024.
**justification for time domain aliasing:**

why not using fft() on length=512? As have been proved in **@Preliminary**,

$$z[n] = x[n] \circledast y[n] = \sum_{l=0}^{l=K-1} x[l]y[n-l-cL]$$

if not enough space for a convolution length 768, matlab will perform cyclic convolution on the filter and the signal, which means the last 255 elements will be add on top of the first 255 elements, i.e. Time Domain Aliasing. for example take a look at $z[0]$:

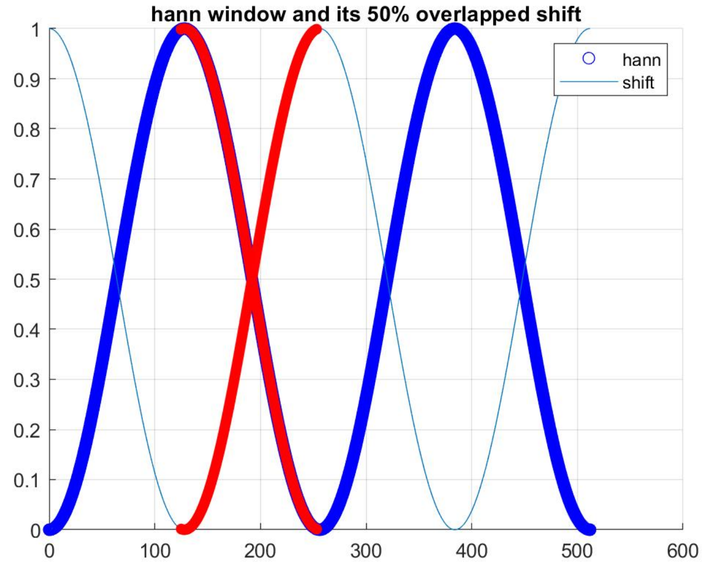
$$y[0] = \sum_{l=0}^{l=511} x[l]h[0-l] + \sum_{l=0}^{l=511} x[l]h[512-l]$$

**To avoid aliasing**, as has been proved in the **@Preliminary**, the convolution between the filter and the short-time section :

$$\underbrace{x_m[n]}_{length=512} * \underbrace{h[n]}_{length=256} = \underbrace{y[n]}_{length=768}$$

which means each Short time Fourier section $\mathbf{Y} = \mathbf{H}.*\mathbf{X}$ should also have length 768. To make fft() efficient, I zero padded $\mathbf{x}$ and $\mathbf{w}$ to $2^{10} = 1024$.

**justification for window selection–Hann window:**




the thick blue line is hann function with length 256, and the thin blue line is hann function shifted 128. And the red part is the overlapping section between any two 50% shift

Hann function

$$W[n] = sin^2(n*\pi/N)$$

and it is easy to observe that after a $\frac{\pi}{2}$ shift, the function becomes $W[n] = cos^2(n*\pi/N)$. Therefore all the windows add up to 1, which is in line rule of constant overlapping add.

## @ Reference

[1] S. Hamid Nawab,Thomas F. Quatieri, Advanced topics in signal processing. Prentice-Hall, Inc. Upper Saddle River, NJ, USA 1987,ISBN:0-13-013129-6

[2] Alan V. Oppenheim. Application of Digital Signal Processing. Prentice-Hall, Inc., Englewood Cliffs, New Jersey.

[3] Julius O. Smith III, Spectral Audio Signal Processing, W3K Publishing, 2011, ISBN 978-0-9745607-3-1. https:ccrma.stanford.edu jossaspMathematical_Definition_STFT.html

[4] Yi-wen Liu,Overlap-Add (OLA) Systems for Audio Analysis and Synthesis, `http://ocw.nthu.edu.tw/ocw/upload/130/1608/10410%E5%8A%89%E5%A5%95%E6%96%87%E6%95%99%E6%8E%88%E6%95%B8%E4%BD%8D%E6%95%B8%E4%BD%8D%E8%81%B2%E8%A8%8A%E5%88%86%E6%9E%90%E8%88%87%E5%90%88%E6%88%90L4_Lecture-4-OLA-2015.pdf`, ,2015,[Online; accessed 19-November-2008]