

**Signal Processing Techniques and Its Applications**  
**Dr. Shyamal Kumar Das Mandal**  
**Advanced Technology Development Centre**  
**Indian Institute of Technology, Kharagpur**

**Lecture - 29**  
**STFT Synthesis**

So, now, in the last class, we talk about STFT Analysis. Now, we talk about STFT Synthesis.

(Refer Slide Time: 00:32)



(Refer Slide Time: 00:33)

STFT Synthesis

For a given value of  $n$ ,  $X(n, \omega)$  has the same properties as a normal Fourier transform, we can recover the input sequence exactly.

For each  $n$ , we take the inverse Fourier transform of  $X(n, \omega)$  from the STFT.

Then obtain  $f[m] = x[m]w[n-m]$  →  $x[m] = \frac{f[m]}{w[n-m]}$


Evaluating  $f[m]$  at  $m = n$ , obtain  $x[n]w[0]$ . Assuming  $w[0] \neq 0$

Then  $x[n] = f[n] / w[0]$

$$x[n] = \frac{f[n]}{w[0]} = \frac{\int_{-\pi}^{\pi} X(n, \omega) w[n-m] e^{j\omega n} d\omega}{w[0]}$$

$$x[n] = \int_{-\pi}^{\pi} \left( \frac{X(n, \omega) w[n-m]}{w[0]} \right) e^{j\omega n} d\omega$$

$$x[n] = \int_{-\pi}^{\pi} X(n, \omega) e^{j\omega n} d\omega$$



So, what I have done is I have a signal  $x[m]$ , and I have multiplied by the  $w[n-m]$  ok. Then I did that DFT discrete Fourier transform, and I got  $X(n, k)$ . What do I have to do if I want to get the signal back? I have to take the IDFT of this one. So, that is what I will get back: I will get back  $x[m]$  multiplied by  $w[n-m]$ .

But, my intention is to get back  $x[m]$ , not that  $x[m]$  multiplied by  $w[n-m]$ . So, if this is my  $f[m]$ , this is my  $f[m]$ . So,  $f[m]$  when it will be equal to  $x[m]$ . So, I can say the  $f[m]$ , if I want to get back the  $x[m]$  is here, then  $x[m]$  is equal to  $f[m]$  divided by  $w[n-m]$ . So, if I say that  $w$ , how do I eliminate  $w[n-m]$ ? If I say that ok,  $f[m]$  is  $f[m]$  is great because every  $m$  equals  $n$ . What is the  $n$ ?  $n$  is the index of the window.

So, if it is a 160-sample window,  $n$  is equal to 160. So, if I evaluate the window for every  $m$  equal to  $m$ , that means, if my window is shifted by one sample, then what we will get at  $m$  equal to  $m$  is nothing but a  $f[m]$  divided by  $w[0]$ . So, if the  $w[0]$  is not equal to 0, then I can say that  $f[m]$  equals  $x[m]$  for every  $m$  equal to  $m$ . So, at  $m$  equal to  $n$ , this will be  $w[0]$ , which is ok. Now, if the  $w[0]$  is not 1 and  $w[0]$  is a constant, then I can get the  $x$  back with no problem. So, that is my limitation in STFT synthesis. So, what do I have to do?

(Refer Slide Time: 03:17)

The slide content is as follows:

$$f_n[m] = x[m] w[n - m]$$

Diagram showing a window function  $w[n-m]$  (red lines) and a signal  $x[m]$  (red line) being multiplied to form  $f_n[m]$ .

Flowchart:

- $f_n[m]$  (with a red arrow pointing to it from the diagram) → DFT →  $X(n, \omega)$
- $X(n, \omega)$  → IDFT at  $m=n$  →  $x[m] w[0]$

Text: If the STFT is unique representation of  $x[n]$  then it is invertible.

$$x[n] = \frac{1}{2\pi w[0]} \int_{-\pi}^{\pi} X(n, \omega) e^{j\omega n} d\omega$$

Synthesis equation for discrete-time STFT

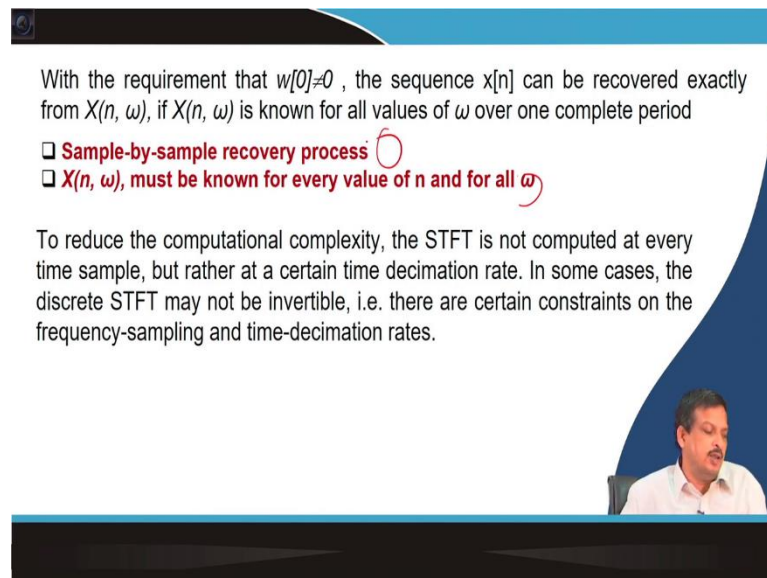
When I suppose I have a signal, I have to analyze it. Let us say 160 samples have been analyzed, but I have to know it for every sample. So, I can only shift the window by one

sample only. So, for every sample, I have to compute inverse DFT, and only then do I get back the signal.

So, that is called sample-by-sample recovery. But, it is very difficult; suppose I have an 8-kilo hertz signal; I take the window size of 160, and then for 160 times, I have to compute the IDFT, which is very time-consuming and is not usable. So, what should I do?

So, what should be the shift of the window? So that, although it has been shifted, I can also get back my  $x[m]$ . So, I have to find out the algorithm STFT synthesis what the allowable shift is possible. So that I can get the signal back. So, that is called STFT synthesis constant ok.

(Refer Slide Time: 04:51)



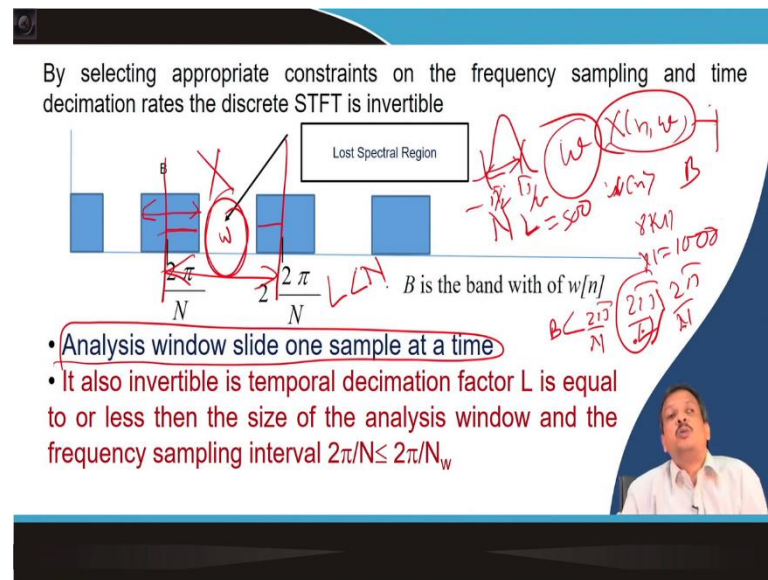
With the requirement that  $w[0] \neq 0$ , the sequence  $x[n]$  can be recovered exactly from  $X(n, \omega)$ , if  $X(n, \omega)$  is known for all values of  $\omega$  over one complete period

- ❑ Sample-by-sample recovery process
- ❑  $X(n, \omega)$ , must be known for every value of  $n$  and for all  $\omega$

To reduce the computational complexity, the STFT is not computed at every time sample, but rather at a certain time decimation rate. In some cases, the discrete STFT may not be invertible, i.e. there are certain constraints on the frequency-sampling and time-decimation rates.

So, let us say what I have said is that I have to know all  $\omega$ s sample by sample, which is not possible. So, I have to find out how much is shifting. Though I have shifted, I also got the signal back.

(Refer Slide Time: 05:08)



What is that, and what is that limitation? So, this is what I have already explained, one sample by one sample, one sample by sample, but the other also. So, I have to know all  $\omega$   $X[n]$   $\omega$  has to know for all  $\omega$ . Now, suppose I have chosen a window  $w[n]$  whose bandwidth is  $B$ . So, this is the bandwidth of the window; when I put the window there, and I have chosen the DF length of the DFT, it is  $N$ .

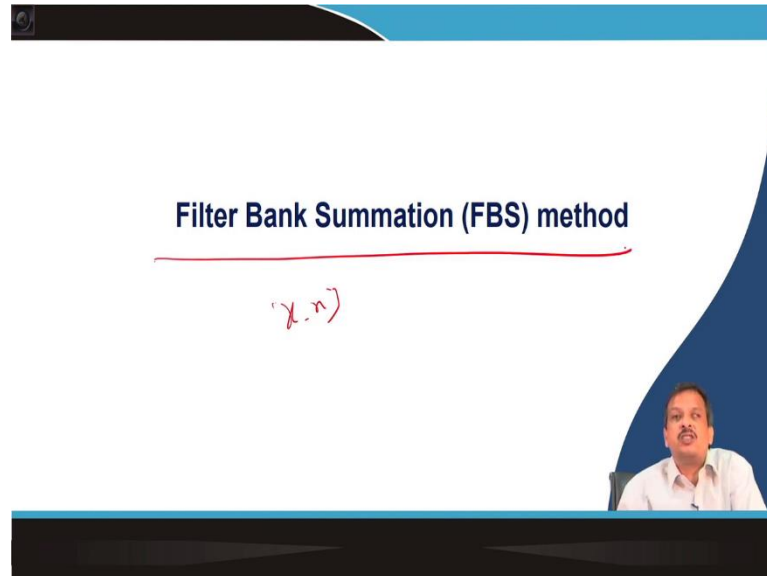
So, the separation between this one and this one is  $2\pi/N$ . Now, the bandwidth is less than; so, this is the bandwidth, and this is the  $B$  by  $2$   $B$  by  $2$ . So, it is nothing but a  $B$ . So, if  $B$  is less than  $2\pi/N$ , this thing will happen. So, this portion of the  $\omega$  I do not know after the filtering because the signal is limited by the filter. Is it clear? For example, suppose I said I have a signal 8-kilo hertz sampling frequency and  $N$  equal to 1000.

Now, say I have chosen an  $L$  window length equal to 500 I a window length equal to  $L$  500. Then what is the main lobe; this is nothing but a, this is  $\pi$  by  $L$ , this is minus  $\pi$  by  $L$ . So, it is  $2\pi/L$  is the bandwidth; here it is  $2\pi/N$ . Now I said, so, what I am doing, I am passing the signal  $2\pi/N$ , and I restricted the bandwidth of  $2\pi/L$ . So,  $2\pi/L$  is larger,  $L$  is smaller than  $N$ . So, this will be larger than this one. So, there will be an overlap.

On the other hand, I can say I have found out that some of the portions of the signal are not. Some of the portion of the frequency is not represented at the output of the  $X$  at the  $X(n, \omega)$ . So, I have not known all  $X[n]$  for all  $\omega$ . Now, if this is happen if the  $L$  is less than capital  $N$ . What is happening in time domain? I choose a signal 1000, but I only multiply

by 1 as 500 and for the rest of the 500, I make 0. So, when I choose the next window, I am not choosing the signal. So, what should I do?

(Refer Slide Time: 08:31)



So, there are two methods for STFT synthesis; one is called the filter bank summation method, and one is called the overlap-add method. Since I will not discuss details here of filter bank summation methods that are already available in another YouTube video when I talk about speech signal processing, you can search on YouTube and find out where you can get the details of filter bank summation methods.

So, I just touched it here because this is a signal processing course. So, I am not emphasising a particular application, filter bank summation method or OLA method. You can read it if you are interested, and then you can go to the YouTube video and read it again. So, filter banks summation methods, such as the DFT or STFT, can be viewed as a summation of a bandpass filter. So, I use that principle to get back the signal  $x[n]$ , and then I call it the filter bank summation method.

(Refer Slide Time: 09:39)

The filter bank interpretation of the STFT shows that for any frequency  $\omega_k$ ,  $X(n, \omega_k)$  is a low-pass representation of the signal in a band centered at  $\omega_k$

$$X(n, \omega_k) = e^{-j\omega_k n} (x[n] * (w[n] e^{j\omega_k n}))$$

$w[n] \longleftrightarrow W(\omega)$

$h[n] = w[n] e^{j\omega_k n} \longrightarrow H[\omega] = W[\omega] \otimes FT(e^{j\omega_k n})$

$$FT(e^{j\omega_k n}) = \sum_{n=-\infty}^{\infty} e^{j\omega_k n} e^{-j\omega n} = \sum_{n=-\infty}^{\infty} e^{-j(\omega - \omega_k)n} = \delta(\omega - \omega_k)$$

$$H(e^{j\omega}) = W(e^{j\omega}) \otimes \delta(\omega - \omega_k) = W(e^{j(\omega - \omega_k)})$$

So, what is mathematics, or what is the analogy behind it? So,  $X(n, \omega_k)$ ;  $\omega_k$  is a particular frequency when I take the  $\omega_0, \omega_1, \omega_2, \omega_k$ . So,  $\omega_k$  is a particular frequency, and this is my STFT analysis  $\omega_k$ , and I said the frequency response of  $\omega_n$  window function is this one.

So, it is nothing but a low pass filter whose bandwidth is this is  $\omega_p$ , this is minus  $\omega_p$ , this is the bandwidth  $B$ . Then the cut-off frequency is  $p$ , upper cut-off frequency and lower cut-off frequency and bandwidth is  $B$  ok or not. So, now, if I say what is, let us say a  $w[n]$  as an  $h$  as a system. The window is nothing but a system which is called a filter, whose frequency impulse response is  $h[n]$ .

So, I can say  $H[\omega]$  is nothing but a  $W[\omega]$ , and Fourier transform of  $e^{j\omega_k n}$ . This is my response to  $h[n]$ . So, what is the Fourier transform of  $e^{j\omega_k n}$  for a particular frequency? So, it will be again multiplied by  $e^{-j\omega n}$ . So, I get  $\omega$  minus  $\omega_k$ . So, it is nothing but a  $\delta$  function  $\omega$  minus  $\omega_k$ .

Now,

$$H(e^{j\omega}) = W(e^{j\omega}) \otimes \delta(\omega - \omega_k) = W(e^{j(\omega - \omega_k)})$$

(Refer Slide Time: 11:45)

$$v_k[n] = X_n(e^{j\omega_k}) = e^{-j\omega_k n} \cdot [x[n] * h[n]]$$

$$V_k(e^{j\omega}) = [H(e^{j\omega}) \cdot X(e^{j\omega})] \otimes FT(e^{-j\omega_k n})$$

$$= [H(e^{j\omega}) \cdot X(e^{j\omega})] \otimes \delta(\omega + \omega_k)$$

$$= [X(e^{j\omega}) \cdot W(e^{j(\omega - \omega_k)})] \otimes \delta(\omega + \omega_k)$$

$$= X(e^{j(\omega + \omega_k)}) \cdot W(e^{j\omega})$$

The diagram illustrates the STFT filtering process:
 

- Modulate:** The input spectrum  $X(\omega)$  is shifted by  $\omega_k$  to  $X(\omega)W(\omega - \omega_k)$ .
- Filter:** A lowpass filter  $W(\omega)$  is applied to the modulated signal.
- Demodulate:** The filtered signal is shifted back to the original frequency.

Let us say  $V_k[n]$  is my recovery signal. So,  $V_k e^{j\omega}$  is nothing but this one, and then I just follow that step, and I get this one. So, this is  $W[\omega]$  modulated to here, then filter, then demodulated, and I get back this one low pass filter. So, this is the process I also mentioned during the filtering view of STFT.

(Refer Slide Time: 12:28)

### Filter Bank Summation (FBS) method

$$y[n] = \frac{1}{Nw[0]} \sum_{k=0}^{N-1} X(n, k) e^{j \frac{2\pi nk}{N}}$$

$$y[n] = \frac{1}{Nw[0]} \sum_{k=0}^{N-1} \left( \sum_{m=-\infty}^{\infty} x[m] w[n-m] e^{-j \frac{2\pi km}{N}} \right) e^{j \frac{2\pi nk}{N}}$$

Since

$$y[n] = \frac{1}{Nw[0]} x[n] * \sum_{k=0}^{N-1} (w[n] e^{j \frac{2\pi nk}{N}})$$

Finite sum over the complex exponential reduce to an impulse train with period N

$$y[n] = \frac{1}{Nw[0]} x[n] * w[n] \sum_{r=-\infty}^{\infty} \delta[n - rN]$$

Now, I have come to the recovery process here. So, what I want is  $y[n]$ ; I want to recover  $y[n]$  from  $x[n]$  from  $X[n, \omega]$  or  $\omega_k$  for a particular frequency. So, if it is discrete DFT, then



it is discrete frequency  $k$ . So,  $y$  said 1 by  $N$  into  $\omega 0$ . So, in that case, it will be 1 by  $\omega 0$ . This one is the recovery of  $y[n]$ .

Inverse Fourier transform of  $X(n, k)$ . Now, I said  $X[n]$  now what is  $X(n, k)$  I put the value  $ok$ ;  $k$  equal to 0 to  $N$  minus 1. So, this I put. So, what I get is that I already know it is, so, if you see, this is nothing but a convolution between the signal and this portion,  $m$  equal to minus infinity to infinity is a convolution function. So, I can say  $y[n]$  is a 1 by  $N\omega[0]$  and convolution of  $x[n]$  and  $w[n]$ .

Because if you see the  $k$  equal to 0 to  $N$  minus 1, this does not depend on  $k$ . So, I take it outside, and this is nothing but a  $\delta[n - rN]$ ;  $r$  equal to minus infinity to infinity. Now, if it is that, then I can say  $y[n]$  is equal to  $x[n]$  when this portion becomes  $N$  into  $w[0]$ ; because  $N w[0]$ ,  $N w[0]$  will be cancelled, and  $y[n]$  will be equal to  $x[n]$ .

(Refer Slide Time: 14:38)

if  $w[n] \sum_{r=-\infty}^{\infty} \delta[n - rN] = w[0] \delta[n]$   $N$

then  $y[n] = x[n]$

Let  $N_w$  is the window length

$N_w$  should be less than or equal to the number of analysis filter  $N$

$\sum_{k=0}^{N-1} W\left(\omega - \frac{2\pi}{N}k\right) = Nw[0]$

Shifted version of the Fourier transform of the analysis window were required to add up to a constant

$N_w > N$  provided the following  $w[rN] = 0$  for  $r = -1, 1, -2, 2, \dots$

Handwritten notes on the slide:

- $N_w \leq N$
- $\frac{2\pi}{Nw} > \frac{2\pi}{N}$

The diagrams show a time-domain window  $w[n]$  of length  $N_w$  and its frequency-domain representation  $W(\omega)$  as a series of shifted rectangular pulses.

So, I can say

$$\sum_{k=-\infty}^{\infty} w[n] \delta[n - rN]$$

Then only I get  $y[n]$  is equal to  $x[n]$ . So, this is the requirement. So, how do I make this portion is  $w[0]$  into  $\delta[n]$ . Let  $N w$  be the window length. So, if  $N w$  is the window length, then this is the frequency response of the window  $ok$ .

And  $N w$  should be less than or equal to the number of analysis filters; then I can say

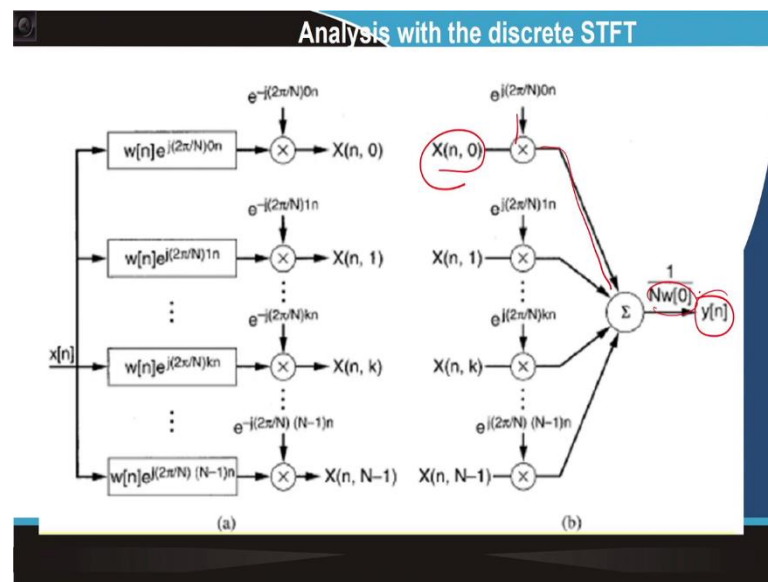


$$\sum_{k=0}^{N_w-1} W\left(\omega - \frac{2\pi k}{N}\right) = N_w W[0]$$

When it is possible, I can say there  $\omega$  if there is a low frequency that is left over. So,  $N_w$  should be less than or equal to the number of analysis windows. So, I can say that I can say what the structure of the  $W \omega$ . So, all the filters will be overlapped in such a way that when I add them up, they become one.

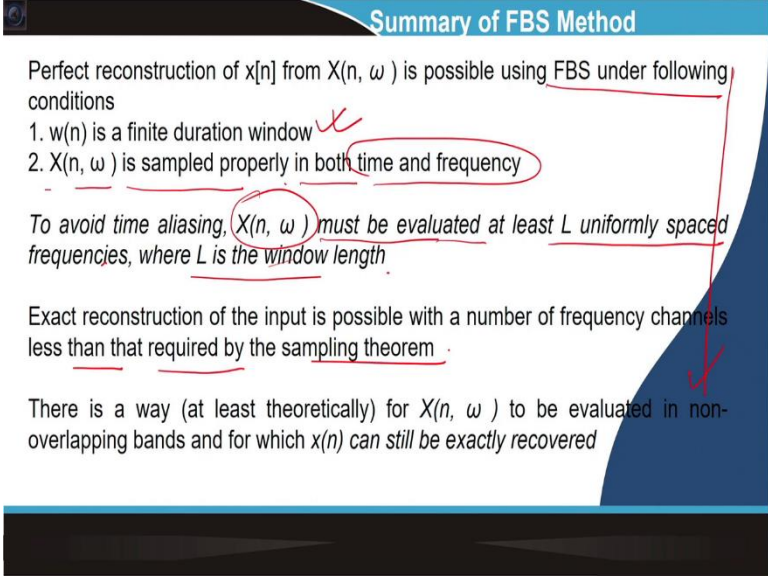
So, that is the filter band summation method. So,  $N_w$  should be less than or equal to a number of  $N_w$  should be less than equal to  $N$  because what is required  $2\pi/N_w$  must be greater than  $2\pi/N$  then only can I know all the frequencies.

(Refer Slide Time: 17:01)



So, that is the filter bank summation method. So, what we will do is get an  $X[n]_0$ , then we demodulate and then again I divide by  $Nw[0]$ , and I get  $y[n]$ . Ok, these are syntheses.

(Refer Slide Time: 17:15)



**Summary of FBS Method**

Perfect reconstruction of  $x[n]$  from  $X(n, \omega)$  is possible using FBS under following conditions

1.  $w(n)$  is a finite duration window ✓
2.  $X(n, \omega)$  is sampled properly in both time and frequency

To avoid time aliasing,  $X(n, \omega)$  must be evaluated at least  $L$  uniformly spaced frequencies, where  $L$  is the window length

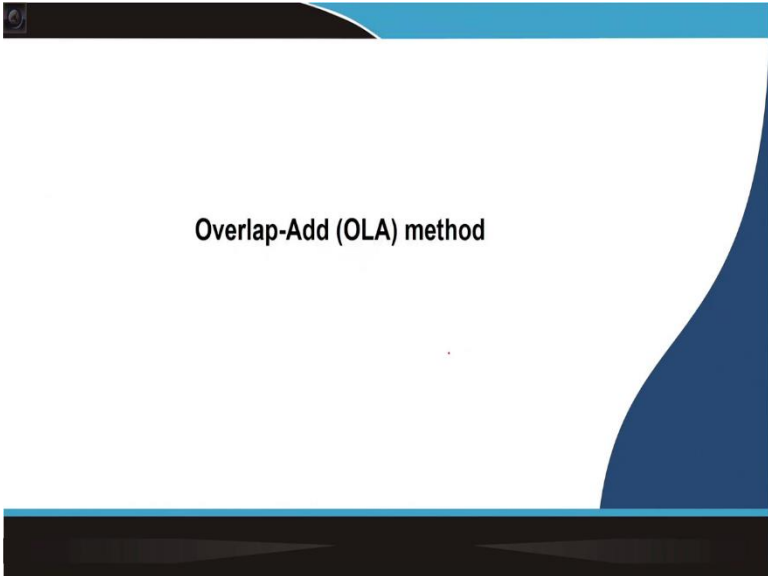
Exact reconstruction of the input is possible with a number of frequency channels less than that required by the sampling theorem

There is a way (at least theoretically) for  $X(n, \omega)$  to be evaluated in non-overlapping bands and for which  $x(n)$  can still be exactly recovered

So, this is a summary of FBS methods. You can read the perfect reconstruction of  $x[n]$  from  $X[n] \omega$  is possible using the FBS method. So,  $\omega n$  is a finite duration window, and  $X(n, \omega)$  is sampled properly in both time and frequency. To avoid time aliasing, in the  $n$  w  $X[n] w[n] \omega$  must be evaluated at least  $L$  uniformly spaced frequency, where  $L$  is the length of the window.

Exact reconstruction of the input is possible with the number of frequency channels less than that required by the sampling theorem. So, that is the summary of FBS methods.

(Refer Slide Time: 18:15)



**Overlap-Add (OLA) method**

Then there is a call Overlap-Add methods.

(Refer Slide Time: 18:18)

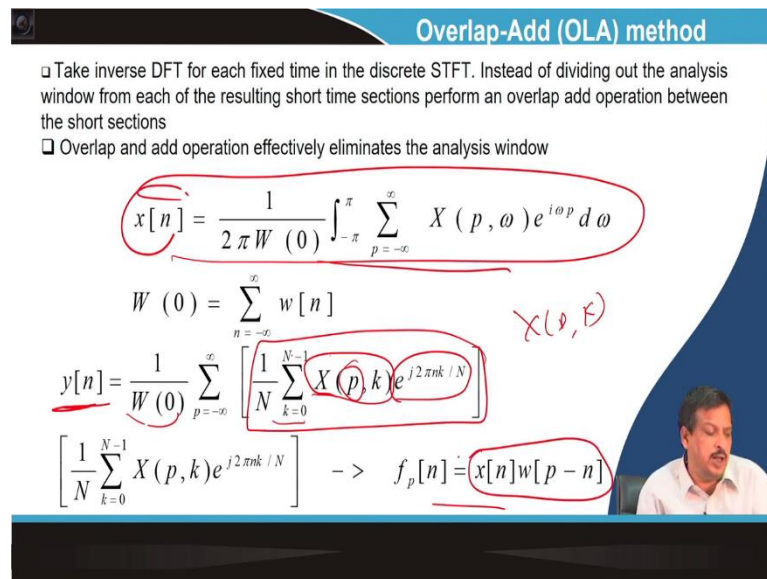
**Overlap-Add (OLA) method**

- Take inverse DFT for each fixed time in the discrete STFT. Instead of dividing out the analysis window from each of the resulting short time sections perform an overlap add operation between the short sections
- Overlap and add operation effectively eliminates the analysis window

$$x[n] = \frac{1}{2\pi W(0)} \int_{-\pi}^{\pi} \sum_{p=-\infty}^{\infty} X(p, \omega) e^{i\omega p} d\omega$$

$$W(0) = \sum_{n=-\infty}^{\infty} w[n]$$

$$y[n] = \frac{1}{W(0)} \sum_{p=-\infty}^{\infty} \left[ \frac{1}{N} \sum_{k=0}^{N-1} X(p, k) e^{j2\pi nk/N} \right]$$

$$\left[ \frac{1}{N} \sum_{k=0}^{N-1} X(p, k) e^{j2\pi nk/N} \right] \rightarrow f_p[n] = x[n]w[p-n]$$


I am not going to detail every method because it is already available on YouTube in another lecture on speech processing.

So, again, I can say my purpose is the same: I have to find out  $x[n]$ . So, if I take this as the DFT DTFT view, let us say I want to, and I will start from here. So,

$$y[n] = \frac{1}{W(0)} \sum_{p=-\infty}^{\infty} \left[ \frac{1}{N} \sum_{k=0}^{N-1} X(p, k) e^{j\frac{2\pi nk}{N}} \right]$$

So, this is nothing but this one  $x[n]$  multiply  $w$  and  $p$  minus  $n$ .

(Refer Slide Time: 19:16)

$$y[n] = \frac{1}{W(0)} \sum_{p=-\infty}^{\infty} x[n] w[p-n]$$

$$y[n] = x[n] \left( \frac{1}{W(0)} \sum_{p=-\infty}^{\infty} w[p-n] \right)$$

$$y[n] = x[n] \quad \text{if} \quad \sum_{p=-\infty}^{\infty} w[p-n] = W(0)$$

Sum of values of a sequence must be the first value of its Fourier Transform

$$\sum_{p=-\infty}^{\infty} w[p-n] = \frac{W(0)}{L}$$

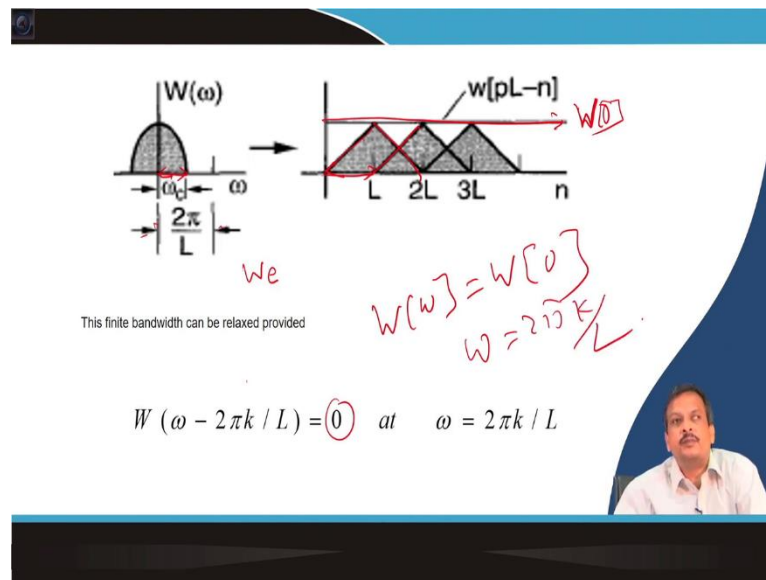
Sum of all the analysis windows to a constant

$$x[n] = \frac{L}{W(0)} \sum_{p=-\infty}^{\infty} \left[ \frac{1}{N} \sum_{k=0}^{N-1} X(pL, k) e^{j2\pi nk/N} \right]$$

So, if you see that  $y[n]$  is equal to 1 by  $w[0]$ , this is nothing but an I said  $x[n]$  is outside. So, this is nothing but a  $w[p-n]$  and  $y[n]$  equal to  $x[n]$  if this one is equal to  $w[0]$ . What is the meaning? The sum of the value of a sequence must be the first value of its Fourier transform. The sum of the value of a sequence  $w[p-n]$  must equal its first energy value. The first value of this is the DC component of the Fourier transform.

So,  $p$  is equal to minus infinity to infinity  $p$  into  $L$ , and  $L$  is the dissipation factor if that. So this is for every sample, so I put  $L$  instead of every sample. So, I also have to take this sample instead of every sample, and this sample's dissipation is  $L$ , which must be  $w[0]$  by  $L$ . So, what is there? The sum of all analysis windows must be constant.

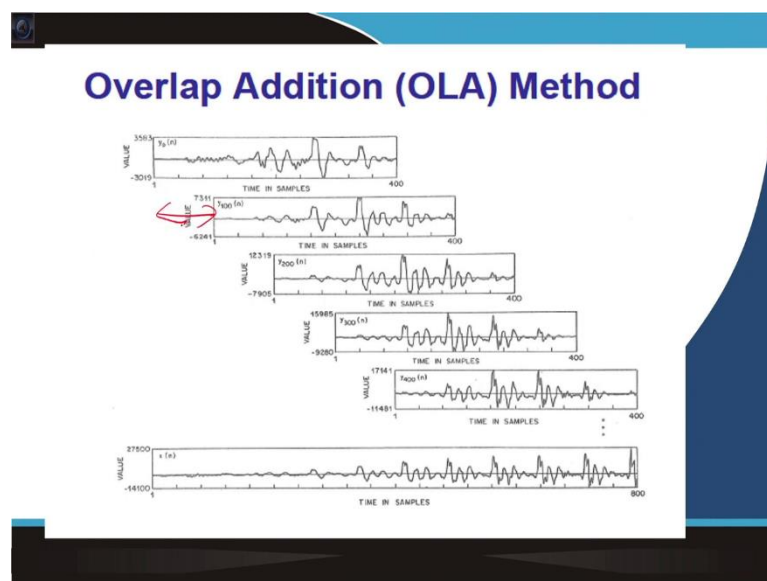
(Refer Slide Time: 20:39)



So, what is the pictorial view? So, if this is my  $\omega$  c. So, I can say the  $\omega$  c, and this is my  $\omega$  L; so, here, if you see that  $\omega$  0 by L. So, I can say this is my window impulse response, and this amount of L shifting is possible. Where if I take the sum, I get  $w[0]$  first value of the window frequency transform, which is ok.

So, where only  $k$  is equal to 0,  $W[\omega]$  is equal to  $W[\omega]$  will be  $W[\omega]$  0, but  $k$  is equal to everywhere it will be 0. So, I can say  $\omega$  must be  $2\pi$  by  $k$  by  $L$  ok.

(Refer Slide Time: 21:48)



So, this is the number of shifts that is possible. So, how much shift is possible that depends on the bandwidth of the window function also. So, there is a summary comparison. I think the summary slide is here or not.

(Refer Slide Time: 22:11)

FBS and OLA Comparisons

FBS Method	OLA Method
$y[n] = \frac{1}{Nw[0]} \sum_{k=0}^{N-1} X(n, k) e^{j \frac{2\pi}{N} kn}$ <p>Adding Frequency component for each n</p> $\sum_{k=0}^{N-1} W\left(\omega - \frac{2\pi}{N}k\right) = Nw[0]$ <p style="text-align: center;">Constraint</p> <p><math>N_w &lt; N \rightarrow y[n] = x[n]</math></p> <p style="color: red; font-size: small;">Sampling relation in frequency</p>	$y[n] = \frac{1}{W[0]} \sum_{p=-\infty}^{\infty} x[n] w[pL - n]$ <p>Adding time component for each n</p> $\sum_{p=-\infty}^{\infty} w[pL - n] = \frac{W[0]}{L}$ <p style="text-align: center;">Constraint</p> <p><math>\omega_c &lt; \frac{2\pi}{N} \rightarrow y[n] = x[n]</math></p> <p style="color: red; font-size: small;">Sampling relation in time</p>

This is the summary. So, this is the FBS method; this is the OLA method. FBS method constraint is that this portion should be equal to this one, which says that the sampling relation in frequency  $N w$  must be less than  $N$ . Here is the constraint is this one that means  $\omega_c$  off frequency of the off frequency of the window must be less than  $2\pi/N$ .

So, that is the time-frequency trade-off between the shift of  $L$ , and how much shift is possible you can get from that things.

(Refer Slide Time: 23:00)

### Short-Time Fourier Transform Magnitude

- The squared STFT magnitude is called the spectrogram or short time Fourier transform magnitude (STFTM).
- Although the phase information is removed in STFT, many signals can still be uniquely synthesized from STFTM.

The relation between STFTM and the short-time autocorrelation  $r[n, m]$  is:

$$r[n, m] = \frac{1}{2\pi} \int_{-\pi}^{\pi} |X(n, \omega)|^2 e^{j\omega m} d\omega$$

$$|X(n, \omega)|^2 = \sum_{m=-\infty}^{\infty} r[n, m] e^{-j\omega m}$$

where  $m$  is the autocorrelation "lag",  $r[n, m] = f[n] * f[n-m]$   
 $f[n] = x[n]w[n]$

I think I have a slide that has an example; I am not going into details on this amplitude magnitude. So, it is also possible that only from the spectral magnitude of the analysis STFT analysis can the signal recovery be proved.

(Refer Slide Time: 23:18)

### Sampling Rate in Time

- to determine the sampling rate in time, we take a linear filtering view
  1.  $X_n(e^{j\omega})$  is the output of a filter with impulse response  $w(n)$
  2.  $W(e^{j\omega})$  is a lowpass response with effective bandwidth of  $B$  Hertz
- thus the effective bandwidth of  $X_n(e^{j\omega})$  is  $B$  Hertz  $\Rightarrow X_n(e^{j\omega})$  has to be sampled at a rate of  $2B$  samples/second to avoid aliasing

Example: Hamming Window

$$w(n) = 0.54 - 0.46 \cos(2\pi n / (L-1)) \quad 0 \leq n \leq L-1$$

$$= 0 \quad \text{otherwise}$$

$\Rightarrow B \approx \frac{2F_s}{L}$  (Hz); for  $L = 100$ ,  $F_s = 10,000$  Hz  $\Rightarrow B = 200$  Hz  $\Rightarrow$  need rate of 400/sec (every 25 samples) for sampling rate in time

Here, I can say that there is an example. So,  $X e^{j\omega}$  is the output of the filter with impulse response  $w[n]$ . And  $w[n]$  is the low pass response with the effective bandwidth  $B$  hertz. So, I have a window whose effective bandwidth is  $B$  hertz. Does the effective bandwidth

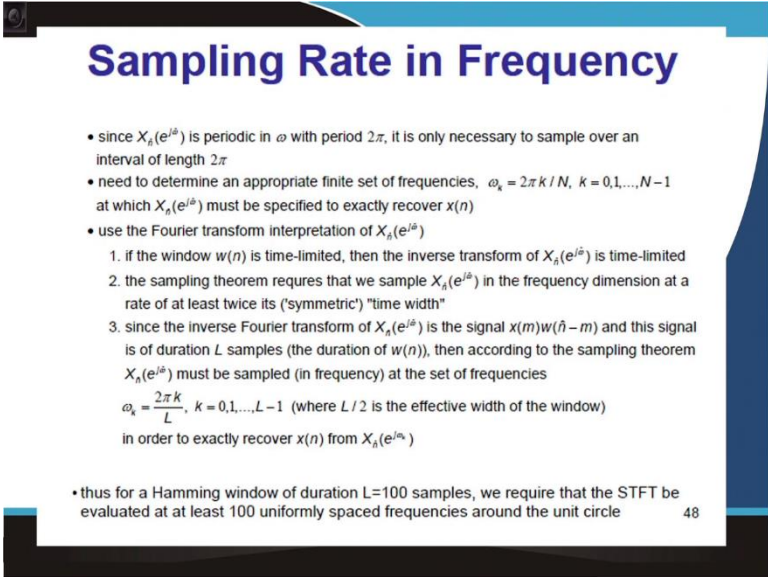


of  $e$  to the power  $B$  hertz; because the bandwidth is the filter, we will define at the bandwidth of the frequency analysis, ok?

This has to be sampled at  $2B$  samples per second. For example, let us say I have taken a hamming window. So, bandwidth is twice  $F_s$  by  $L$  twice  $F_s$  by  $L$ . So, for  $L$  equal to 100 and  $F_s$  equal to 10 kilohertz,  $B$  is equal to 200 hertz; that means it requires 400 needs the rate sampling rate time sampling rate is 400 sample per second ok.

So, that means here I can shift the window. If the length of the window is 100, I can shift the window at 25 samples. Twice  $F_s$  by  $L$  bandwidth is 200 hertz; so, I require  $2B$  sample per second ok rate in time.

(Refer Slide Time: 25:02)



## Sampling Rate in Frequency

- since  $X_d(e^{j\omega})$  is periodic in  $\omega$  with period  $2\pi$ , it is only necessary to sample over an interval of length  $2\pi$
- need to determine an appropriate finite set of frequencies,  $\omega_k = 2\pi k / N$ ,  $k = 0, 1, \dots, N-1$  at which  $X_d(e^{j\omega})$  must be specified to exactly recover  $x(n)$
- use the Fourier transform interpretation of  $X_d(e^{j\omega})$ 
  1. if the window  $w(n)$  is time-limited, then the inverse transform of  $X_d(e^{j\omega})$  is time-limited
  2. the sampling theorem requires that we sample  $X_d(e^{j\omega})$  in the frequency dimension at a rate of at least twice its ('symmetric') "time width"
  3. since the inverse Fourier transform of  $X_d(e^{j\omega})$  is the signal  $x(m)w(\hat{n}-m)$  and this signal is of duration  $L$  samples (the duration of  $w(n)$ ), then according to the sampling theorem  $X_d(e^{j\omega})$  must be sampled (in frequency) at the set of frequencies
 
$$\omega_k = \frac{2\pi k}{L}, \quad k = 0, 1, \dots, L-1 \quad (\text{where } L/2 \text{ is the effective width of the window})$$
 in order to exactly recover  $x(n)$  from  $X_d(e^{j\omega_k})$
- thus for a Hamming window of duration  $L=100$  samples, we require that the STFT be evaluated at at least 100 uniformly spaced frequencies around the unit circle

48

So, I will supply this material to you.

(Refer Slide Time: 25:07)

## “Total” Sampling Rate of STFT

- the “total” sampling rate for the STFT is the product of the sampling rates in time and frequency, i.e.,  
 $SR = SR(\text{time}) \times SR(\text{frequency})$   
 $= 2B \times L \text{ samples/sec}$   
 $B = \text{frequency bandwidth of window (Hz)}$   
 $L = \text{time width of window (samples)}$
- for most windows of interest,  $B$  is a multiple of  $F_s/L$ , i.e.,  
 $B = C F_s/L \text{ (Hz)}$ ,  $C=1$  for Rectangular Window  
 $C=2$  for Hamming Window  
 $SR = 2C F_s \text{ samples/second}$
- can define an ‘oversampling rate’ of  
 $SR/F_s = 2C = \text{oversampling rate of STFT as compared to}$   
 $\text{conventional sampling representation of } x(n)$   
for RW,  $2C=2$ ; for HW  $2C=4 \Rightarrow$  range of oversampling is 2–4  
this oversampling gives a very flexible representation of the speech signal

If you are interested in the details, read it. You can read it for the details of the STFT analysis, which will be available on YouTube. You can also read it from this slide. So, because I am not loading the course with the particular analysis because of STFT again, it can take another week to explain the details of STFT analysis.

So, thank you.