DISCRETE-TIME AND DISCRETE FOURIER TRANSFORM

CONTENTS (ADDITIONAL READING)

SPECTROGRAPHIC ANALYSIS OF SPEECH

SHORT TIME FOURIER TRANSFORM ANALYSIS

TIME-FREQUENCY RESOLUTION TRADEOFF

FAST FOURIER TRANSFORM

1. Spectrographic analysis of speech

- The Fourier transform of the windowed speech waveform, i.e. the short-time Fourier transform (STFT), is given by

$$X(\omega,\tau)=\sum_{n=-\infty}^{\infty}x[n,\tau]\exp(-jn\omega). \tag{1-1}$$

where $x[n,\tau]=w[n,\tau]\cdot x[n]$ is the windowed speech segments as a function of the window center at time τ.

- The spectrogram is a graphical display of the magnitude of the time-varying spectral characteristis and is given by

$$S(\omega,\tau)=|X(\omega,\tau)|^2, \tag{1-2}$$

which is a measure of the energy of the frequency component at frequency ω in the neighborhood of τ.

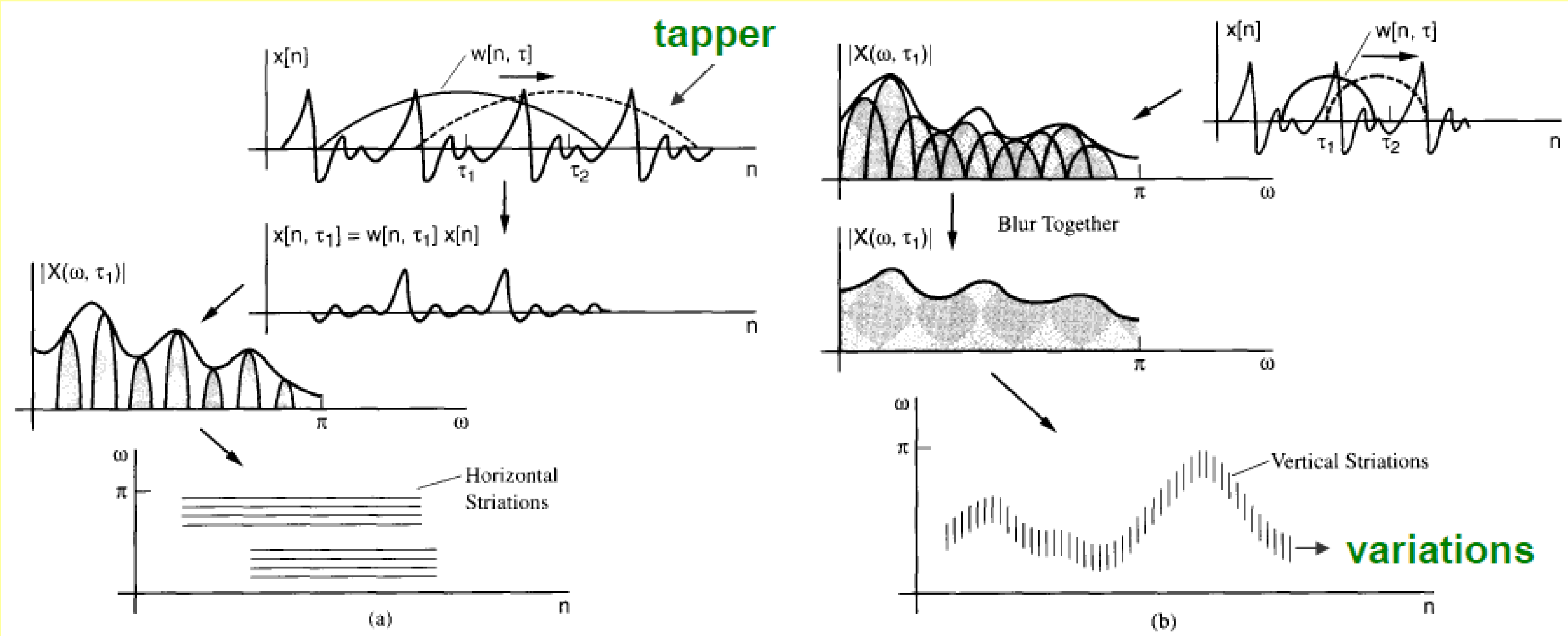

Fig. 3.14 Formation of (a) the narrowband and (b) the wideband spectrograms.

- The figure shows two types of spectrograms: narrowband (good spectral resolution with large window length, e.g. 20ms) and wideband (good time resolution with short window length – e.g. 4ms Hamming window).

2.1 Short-Time Fourier Transform (STFT) Analysis

- Given time-series $x[n]$, the **STFT** at time n is given as:

$$X(n,\omega) = \sum_{m=-\infty}^{\infty} x[m]w[n-m]e^{-j\omega m}, \qquad (2.1)$$

where $w[n]$ is the analysis window, which is assumed to be non-zero only in the interval $[0, N_w - 1]$.

- The discrete STFT is obtained by sampling $X(n,\omega)$ over the unit circle:

$$X(n,k) = X(n,\omega)\Big|_{\omega=\frac{2\pi}{N}k} = \sum_{m=-\infty}^{\infty} x[m]w[n-m]e^{-j\frac{2\pi}{N}km}, \qquad (2.2)$$

where N is the frequency sampling factor and $2\pi/N$ is the frequency sampling interval.

- **An alternate (filtering) view of the discrete STFT is:**

$$X(n,\omega_0) = \sum_{m=-\infty}^{\infty} (x[m]e^{-j\omega_0 m}) w[n-m] = (x[n]e^{-j\omega_0 n}) * w[n]. \quad \text{(2.3)}$$

That is, the signal x[n] is first modulated with $e^{-j\omega_0 n}$, and then passed through a filter with impulsive response $w[n]$.

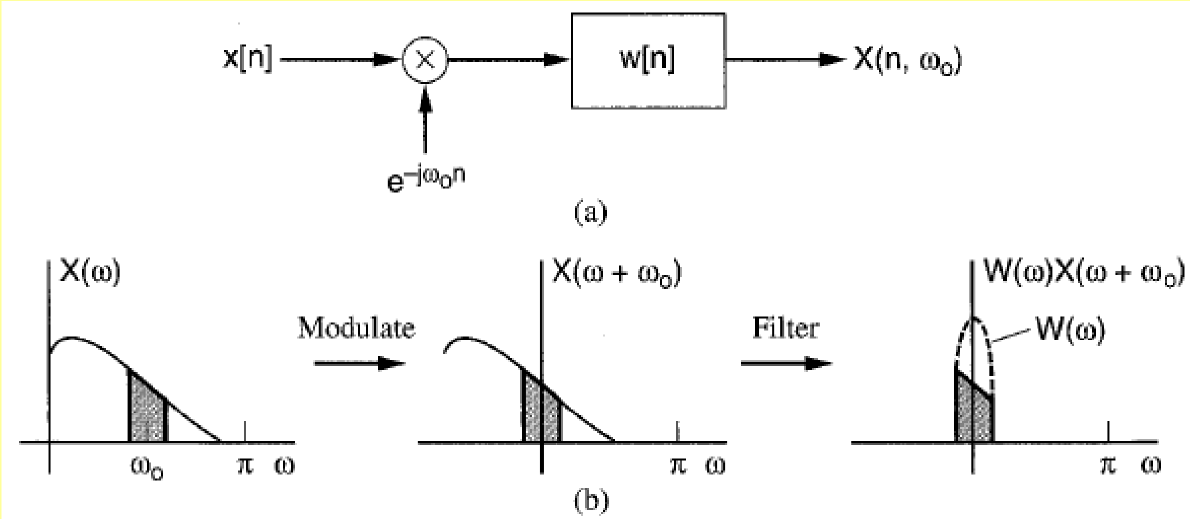

Fig. 7.3 Filtering view of STFT analysis at frequency ω_0: (a) block diagram of complex exponential modulation followed by a lowpass filter; (b)operations in the frequency domain.

- **An equivalent representation of (2.3) is:**

$$X(n,\omega_0) = e^{-j\omega_0 n}\left(x[n] * w[n]e^{j\omega_0 n}\right). \qquad \textbf{(2.4)}$$

That is, the sequence $x[n]$ is first passed through the filter $w[n]$ with a linear phase factor. The output is then modulated by $e^{-j\omega_0 n}$.

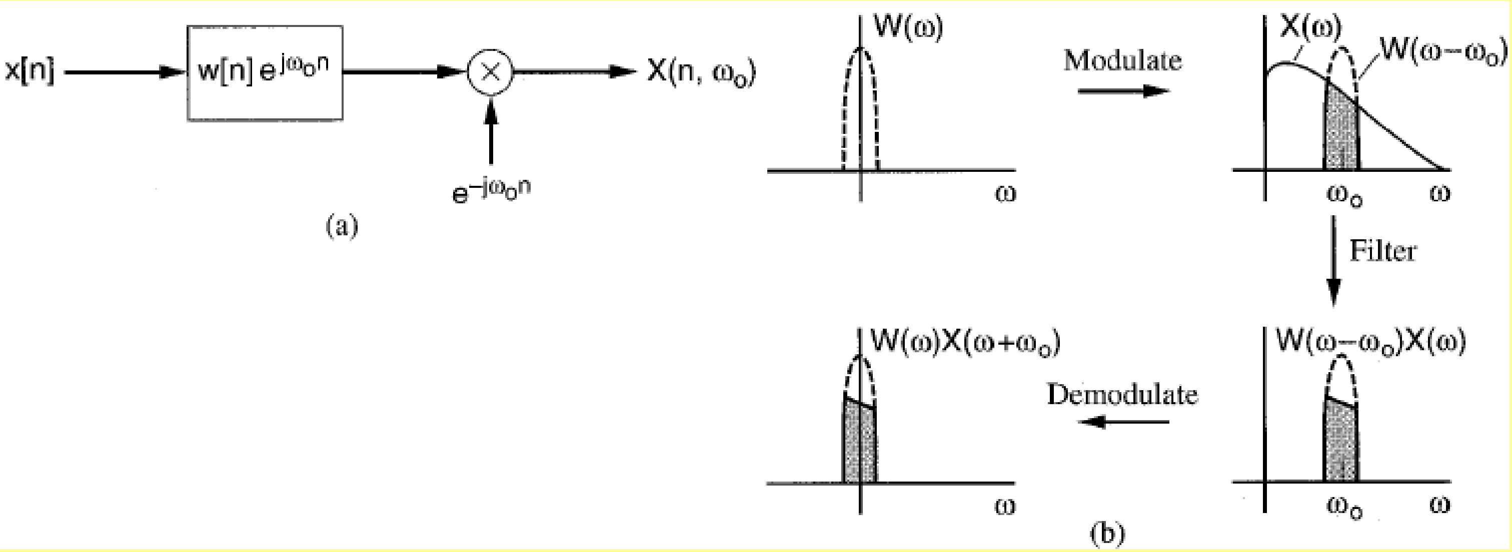

Fig. 7.4 Alternative filtering view of STFT analysis at frequency ω_0: (a) block diagram of bandpass filtering followed by complex exponential modulation; (b)operations in the frequency domain.

2.1.3 Time-Frequency Resolution Tradeoffs

- The STFT can be also written as

$$X(n,\omega) = \frac{1}{2\pi}\int_{-\pi}^{\pi} W(\theta)e^{j\theta n} X(\omega+\theta)d\theta, \qquad (2.5)$$

$X(\omega)$ is the Fourier transform of $x[m]$ and $W(-\omega)e^{j\omega n}$ as the Fourier transform of $w[n-m]$ with respect to m.

- The size of $w[n]$ affects the time-frequency resolution of STFT:

Window size of $w[n]$	Bandwidth of $W(\omega)$	Time resolution	Frequency resolution	Good for:
long	narrow	bad	good	sinusoidal components, (harmonic)
short	wide	good	bad	fast time-varying components, (rapid conversational speech)

A fundamental problem of STFT and other time-frequency analysis techniques is the selection of the windows to achieve a good tradeoff between time and frequency resolution.

Example:

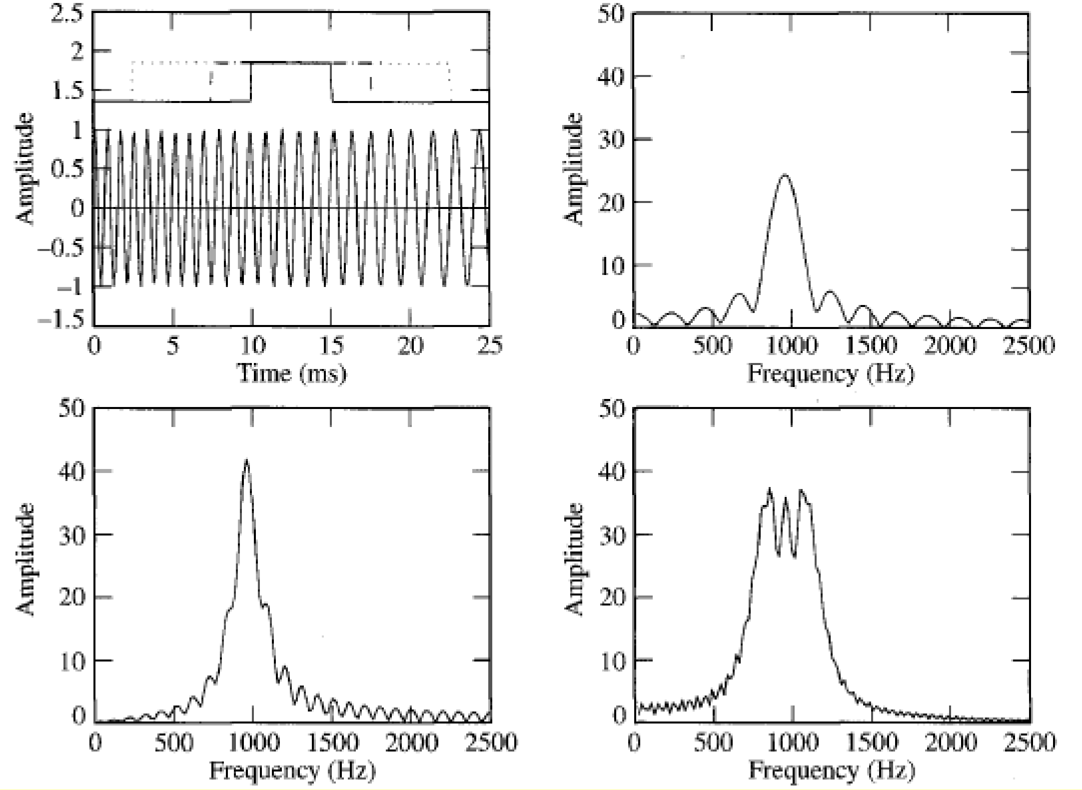

Fig. 7.8 Effect of the length of the analysis window on the discrete Fourier Transform of linearity frequency-modulated sinusoid of 25 ms whose frequency decreases from 1250 Hz to 625 Hz. The Fourier transform uses a rectangular window centered at 12.5 ms, as illustrated in (a). Transform are shown for different window lengths: (b) 5 ms [solid in (a)]; (c) 10 ms [dashed in (a)]; (d) 20 ms [dotted in (a)].

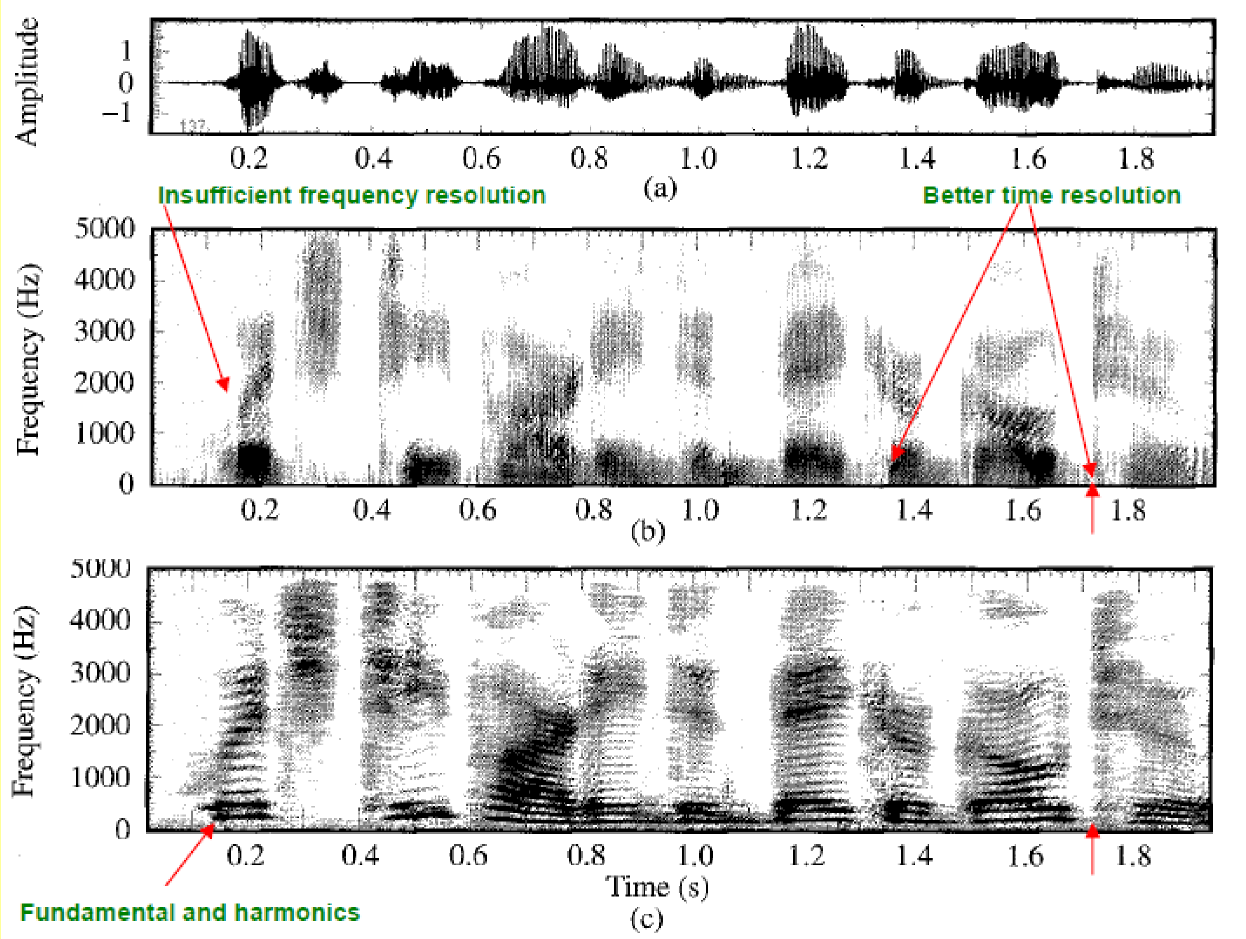

Fig. 3.15 Comparison of measured spectrograms for the utterance, "which tea party did Baker go to?": (a) speech waveform; (b) wideband spectrogram; (c) narrowband spectrogram.

3. EXAMPLES

Example 1: Power Quality Monitoring

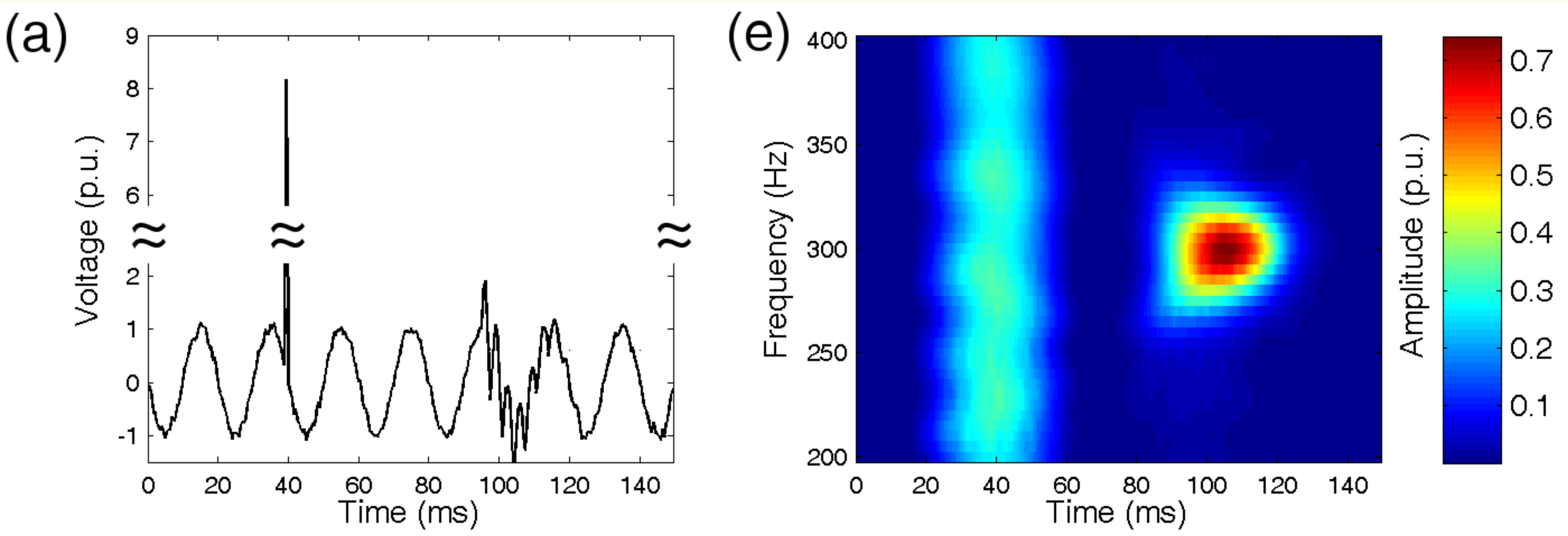

Fig. 3. Time-frequency representation and separation of impulsive and transient power transients. (a) Simulated 50-Hz power waveforms with an impulsive transient at 40 ms and an oscillatory transient from 95 ms to 125 ms in Time representation; (e) Time-frequency representation of (a) obtained from spectrogram.

Example 2: Electroencephalography (EEG) of an eye (event detection in Frequency domain)

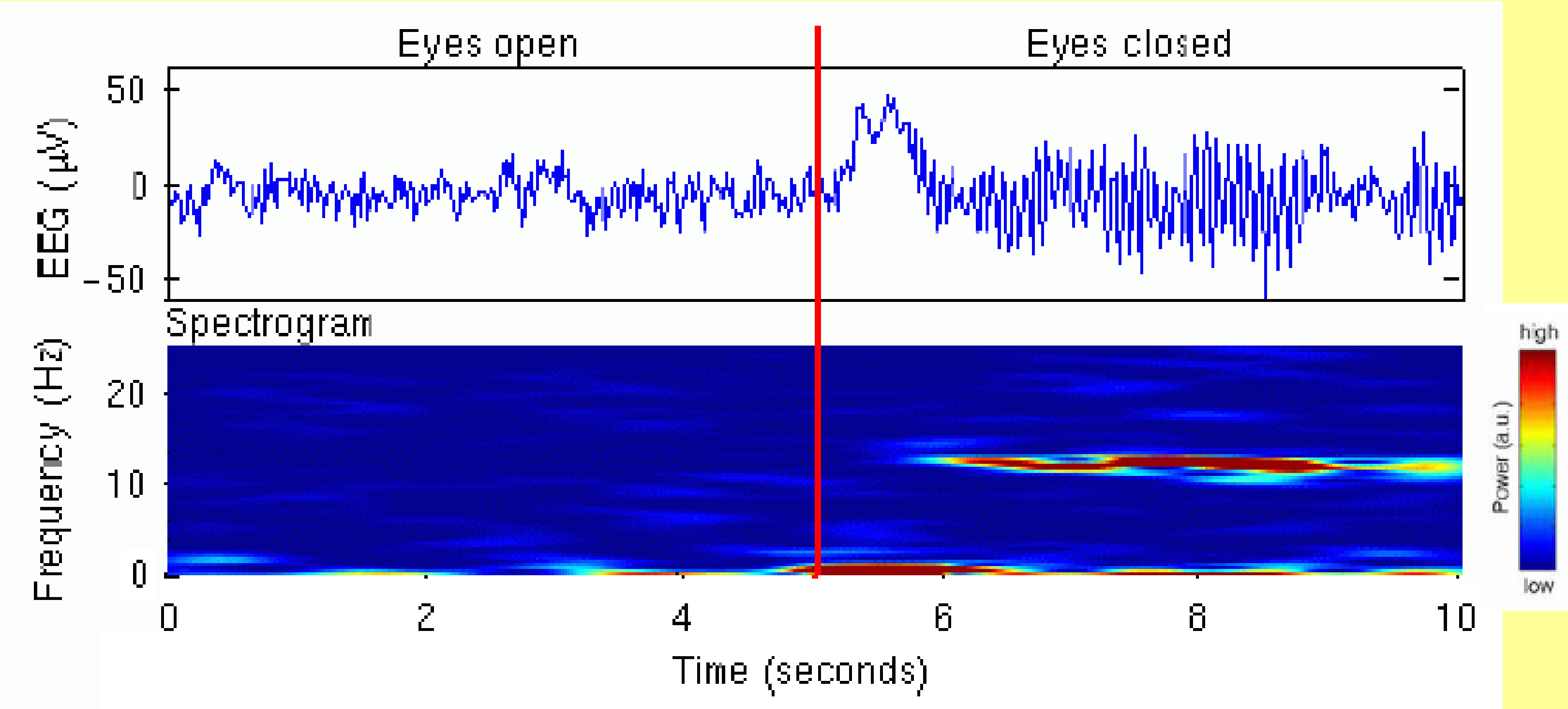

Fig. 4. Time-frequency representation of the EEG of an eye during opening and closing.

4. FAST FOURIER TRANSFORM

- Direct computation of the DFT requires N^2 complex multiplications and N^2 complex additions. Fast algorithms for DFT are called **Fast Fourier Transform** (FFT) with order O($N \log_2 N$) arithmetic complexity.

- In the FFT, a DFT of length $N = N_1 N_2$ can be decomposed into two or more smaller DFTs with simpler implementation and hence the complexity is reduced. It is based on a technique called **multi-dimensional index mapping**.

- Two classes of **index mapping** are

  Common factor map (CFM), which works for general N_1 and N_2.

  Prime Factor Map (PFM), which works for relative prime N_1 and N_2.

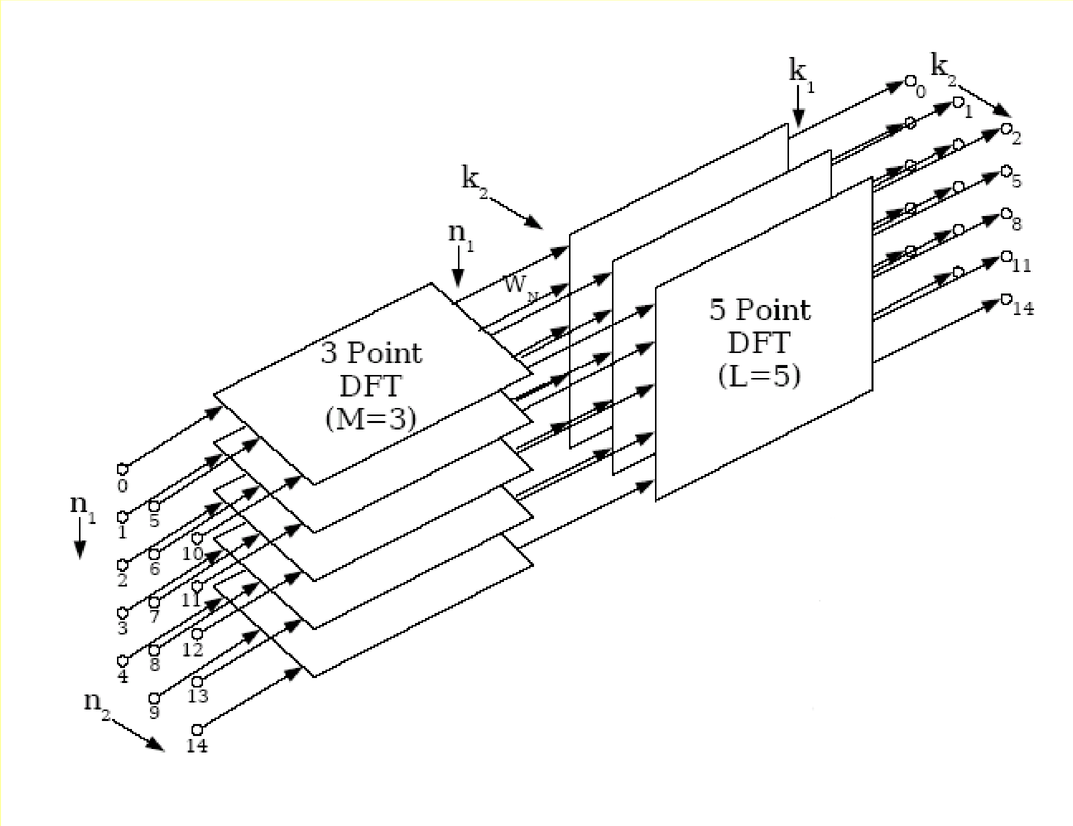

Fig. 5 An example of using Common Factor Mapping (4.5a and 4.5b) where the original 15-point DFT is decomposed to a series of row 3-point DFTs followed by twiddle multiplications and a series of column 5-point DFTs. $N = 15$ with $N_1 = 3$, $N_2 = 5$.

With PFM, the twiddle multiplications can be avoided. The input and output mapping will be given by (4.7a and 4.7b).

4.1 Common Factor Map

- **The index mapping of common factor map (CFM) is given as**

$$n = N_2 n_1 + n_2, \qquad n_1 = 0,..,N_1 - 1; n_2 = 0,..,N_2 - 1 \qquad \textbf{(4.1a)}$$

$$k = k_1 + N_1 k_2, \qquad k_1 = 0,..,N_1 - 1; k_2 = 0,..,N_2 - 1 \qquad \textbf{(4.1b)}$$

This is called decimation in frequency (DIF) form. Substituting into (9.15) gives:

$$X[k_1, k_2] = X[k_1 + N_1 k_2] = \sum_{n = N_2 n_1 + n_2} x[N_2 n_1 + n_2] W_N^{(N_2 n_1 + n_2)(k_1 + N_1 k_2)} \qquad \textbf{(4.2)}$$

$$= \sum_{n_2 - 1}^{N_2 - 1} \left[\sum_{n_1 - 1}^{N_1 - 1} x[N_2 n_1 + n_2] W_{N_1}^{n_1 k_1} \right] \left(W_N^{k_1 n_2} \right) \cdot W_{N_2}^{n_2 k_2}, \text{ where } W_N = e^{-j2\pi / N}.$$

If one treats $x[N_2 n_1 + n_2]$ as a 2D array $x[n_1, n_2]$, then

- **the operations in the square bracket is to perform an N_1-point DFT along n_1 for each n_2 (N_2 in total) . After transform, the row index is given by k_1.**
- **the term in curve bracket is to multiple each element of the array by $\left(W_N^{k_1 n_2} \right)$.**
- **the last summation is equivalent to perform an N_2-point DFT along n_2 for each k_1 (with number N_1 in total) . After transform, the row index is given by k_2.**

- In summary, the DFT is done by performing N_2 length-N_1 DFTs on the first dimension (row) of the input two dimension array following by the twiddle multiplications, $W_N^{k_1 n_2}$, and the N_1 length-N_2 DFTs along the other dimension (column).
- Notice that the order cannot be interchange due to the twiddle factor.
- If $N_1 = 2$ and $N_2 = N/2$, W_{N_1} becomes $W_{N_1} = e^{-j\pi} = -1$. Then the multiplication with W_{N_1} can be simplified to an addition in the 2-point DFT.
- If $N = 2^M$, the process can be repeated M times, leading to the DIF Radix-2 FFT.
- The radix r is related to the small DFT used.
- In the decimation-in-frequency (DIF) FFT, (4.2) reads

$$X[k_1,k_2] = \sum_{n_2=0}^{N/2-1}\left[\sum_{n_1=0}^{1} x[n_1,n_2](-1)^{n_1 k_1}\right]\left(W_N^{n_2 k_1}\right)W_{N/2}^{n_2 k_2} \qquad (4.3)$$

- **It is called DIF radix-2 FFT because it can also be obtained by decimating the frequency indices separately into the even and odd indexed outputs of the DFT as follows:**

$$X[2k] = \sum_{n=0}^{N/2-1} \left[x[n] + x[n+N/2] \right] W_{N/2}^{nk} \qquad \textbf{(4.4a)}$$

$$X[2k+1] = \sum_{n=0}^{N/2-1} \left[x[n] - x[n+N/2] \right] \left(W_N^n \right) W_{N/2}^{nk} \qquad \textbf{(4.4b)}$$

- **The length-*N* DFT is computed from two length-*N*/2 DFTs. The decomposition can repeatedly be applied to the smaller DFTs until eventually only trivial two-point DFTs remain.**

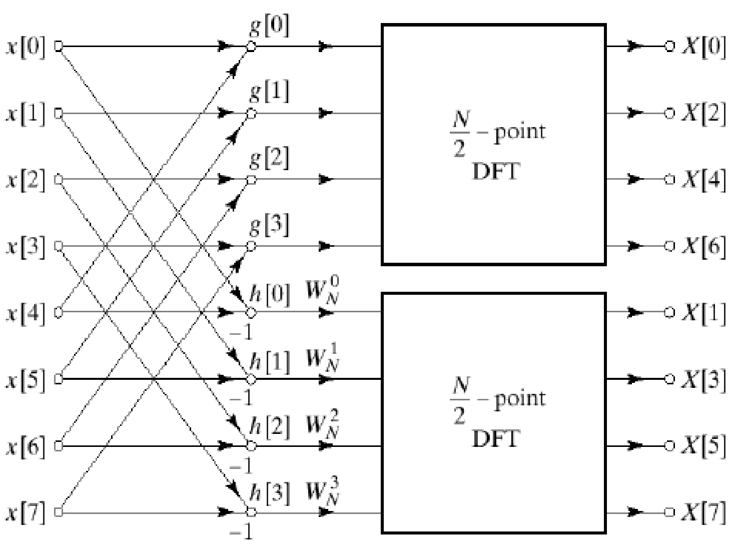

Fig. 7a. Decimation in frequency of a length N DFT into two length $N/2$ DFTs preceded by a preprocessing stage.

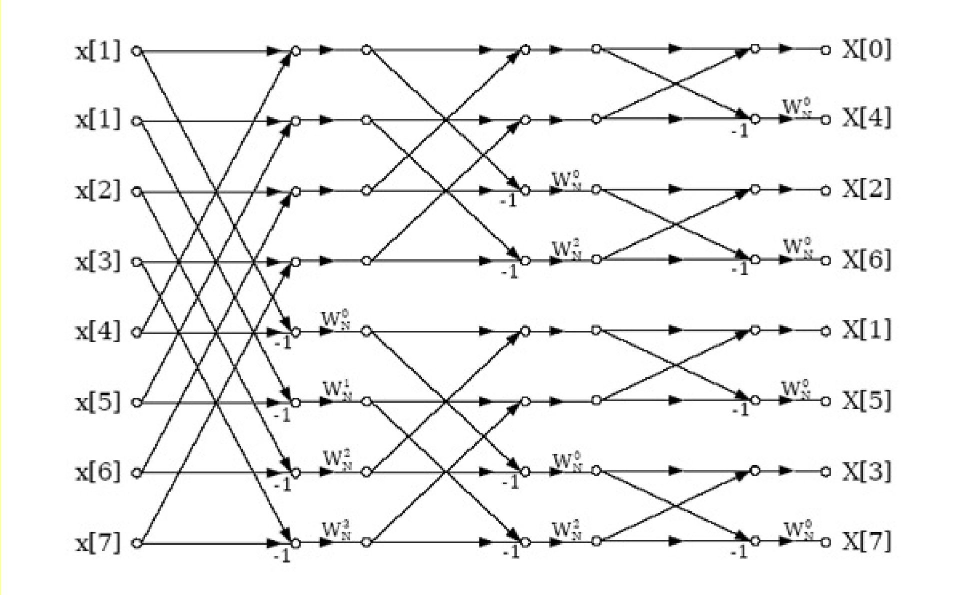

Fig. 7b. A length-8 Radix-2 DIF FFT Diagram.

- **Another equivalent form called decimation-in-time (DIT) form can be obtained by interchanging the role of (4.1a) and (4.1b).**

$$n = n_1 + N_1 n_2, \quad n_1 = 0,..,N_1 - 1; n_2 = 0,..,N_2 - 1 \qquad \textbf{(4.5a)}$$

$$k = N_2 k_1 + k_2, \quad k_1 = 0,..,N_1 - 1; k_2 = 0,..,N_2 - 1 \qquad \textbf{(4.5b)}$$

$$X[k_1,k_2] = X[N_2k_1+k_2] = \sum_{n=n_1+N_1n_2} x[n_1+N_1n_2] W_N^{(n_1+N_1n_2)(N_2k_1+k_2)} \quad \textbf{(4.5c)}$$

$$= \sum_{n_1-1}^{N_1-1}\left[\sum_{n_2-1}^{N_2-1} x[n_1+N_1n_2] W_{N_2}^{n_2k_2}\right]\left(W_N^{n_1k_2}\right)\cdot W_{N_1}^{n_1k_1}, \text{ where } W_N = e^{-j2\pi/N}.$$

- The DFT is done by performing N_1 length-N_2 DFTs on the second dimension (column) of the input two dimension array following by the **twiddle multiplications**, $W_N^{n_1k_2}$, and the N_2 length-N_1 DFTs along the other dimension (row).
- This can be viewed as the DIF algorithm with the row and column of the array interchanged. So they are actually equivalent.
- This can also be derived by decimating the time indices as follows:

$$X[k] = \sum_{n=0}^{N/2-1} x[2n] W_{N/2}^{nk} + W_N^k \sum_{n=0}^{N/2-1} x[2n+1] W_{N/2}^{nk} \quad \textbf{(4.5d)}$$

$$X[k+N/2]=\sum_{n=0}^{N/2-1}x[2n]W_{N/2}^{nk}-W_N^k\sum_{n=0}^{N/2-1}x[2n+1]W_{N/2}^{nk} \qquad \textbf{(4.5e)}$$

- **For $N=2^m$, the dividing process is repeated $m=\log_2 N$ times and requires $N/2$ multiplications each time. This gives an arithmetic complexity of only $(N/2)\log_2 N$ as compared with N^2 for direct calculation.**

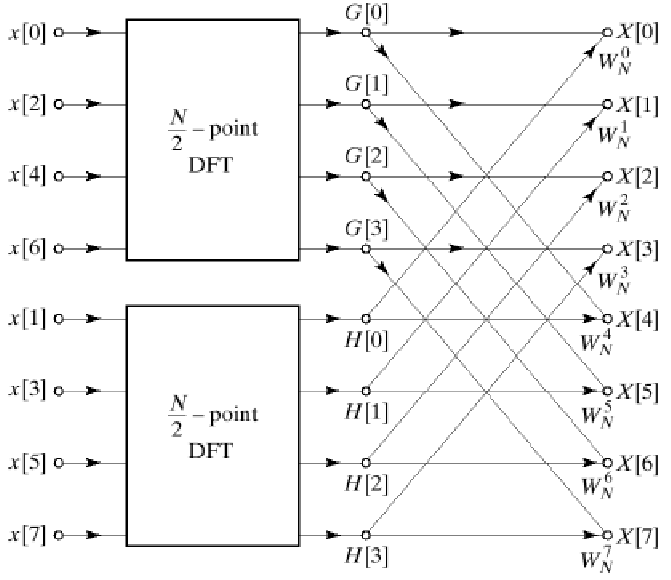

Fig. 8a. Decimation in time of a length N DFT into two length $N/2$ DFTs followed by a combining stage.

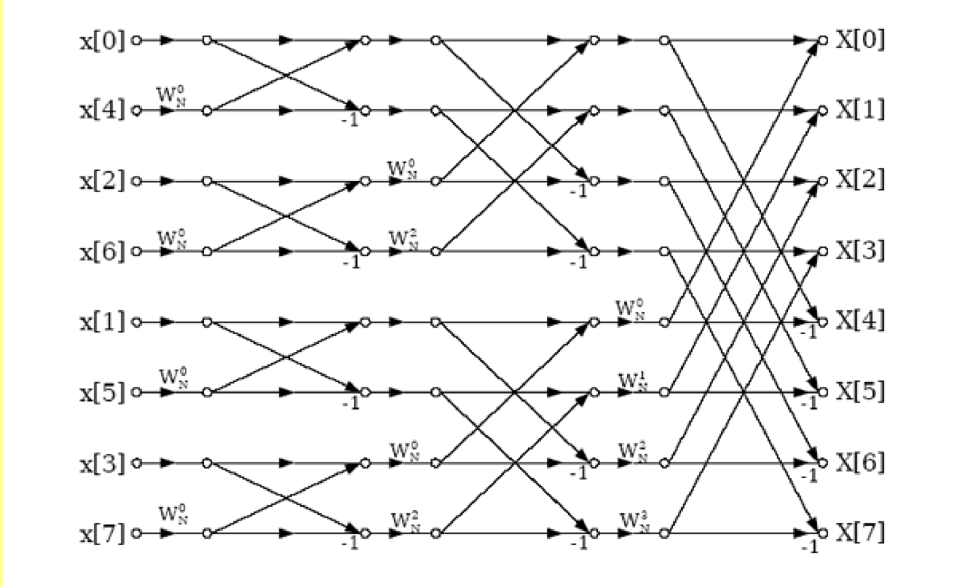

Fig. 8b. A length 8 Radix-2 DIT FFT Diagram.

- **It can be seen that Fig. 8b. is a transpose of Fig. 7b.**

- **Other commonly used Radix-r FFT are Radix-4 ($N_2 = N/4$) and Radix 8 ($N_2 = N/8$) FFT. Combination of them is possible.**

- **Increasing the Radix-r will reduce the number of stages and hence the twiddle factor multiplications, but the complexity for the length-r DFT will also increase. For small r, such as 2, 4, 8 and 16, the short DFTs usually can be implemented efficiently.**

- **Radix-2 has a simple structure while Radix-4 and Radix-8 are more efficient.**

4.2 Prime Factor Map

- **An advantage of the Prime Factor Algorithm (PFA) is the elimination of the twiddle factors.**

- **If N is a product of *m* relatively prime factors,**

$$N = N_1 N_2 \ldots N_m, \qquad \textbf{(4.6)}$$

then the general form of the Prime Factor Map (PFM) is

$$n = \langle r_1 M_1 n_1 + \ldots + r_m M_m n_m \rangle_N \qquad \textbf{(4.7a)}$$

$$k = \langle \tilde{r}_1 M_1 k_1 + \ldots + \tilde{r}_m M_m k_m \rangle_N \qquad \textbf{(4.7b)}$$

where $\langle r \rangle_N = r \bmod N$ **.** $M_i = N / N_i$, $\langle \tilde{r}_i r_i M_i \rangle_N = 1; \quad i = 1, \ldots, m.$

- **The DFT becomes:**

$$X[k_1,..,k_m] = \sum_{n_1=0}^{N_1-1} .. \sum_{n_m=0}^{N_m-1} x[n_1,..,n_m] W_{N_1}^{n_1 k_1} .. W_{N_m}^{n_m k_m} \quad (4.8)$$

$$0 \le n_i \le N_i - 1; \quad 0 \le k_i \le N_i - 1; \quad i = 1,..,m.$$

where $x[n_1,..,n_m] = x[\langle r_1 M_1 n_1 + ... + r_m M_m n_m \rangle_N]$, **and**

$$X[k_1,...,k_m] = X[\langle \tilde{r}_1 M_1 k_1 + ... + \tilde{r}_m M_m k_m \rangle_N].$$

- **The 1-D array is now mapped to a *m*-dimensional array. (4.8) is a *m*-dimensional DFT which corresponding to performing N_i-point DFT along the *i*-th dimension (N/N_i in total). Since the twiddle factor is missing, the order is immaterial and the arithmetic complexity can be greatly reduced.**
- **An example is given in figure 5 for $N = 15 = 3 \times 5$. In a Prime Factor Algorithm (PFA), it is common to use:**

$$r_i = 1, i = 1,...m. \qquad \text{(Ruritanian map)} \quad (4.9a)$$

$$\tilde{r}_i = M_i^{-1} \bmod N_i, i = 1,...,m \qquad \text{(Chinese Remainder Map)} \quad (4.9b)$$

4.3 Summary

- Two classes of index mapping are the common factor map (CFM) and the Prime Factor Map (PFM).
- Radix-2, Radix-4, and Radix-8 are some commonly used FFT algorithms relying on the CFM. DIT and DIF are two different configurations of the Radix-r FFT.
- Increasing the Radix-r will reduce the number of stages and hence the twiddle factor multiplications, but the complexity for the length-r DFT will also increase. For small *r*, such as 2, 4, 8 and 16, the short DFTs usually can be implemented efficiently.
- An advantage of the Prime Factor Algorithm (PFA) is the elimination of the twiddle factors. The structure however is slightly complicated with somewhat restricted lengths, though all both CFM and PFM can be combined to cover composite lengths.