Procesamiento digital de señales de audio

STFT: síntesis y procesamiento

Instituto de Ingeniería Eléctrica
Facultad de Ingeniería

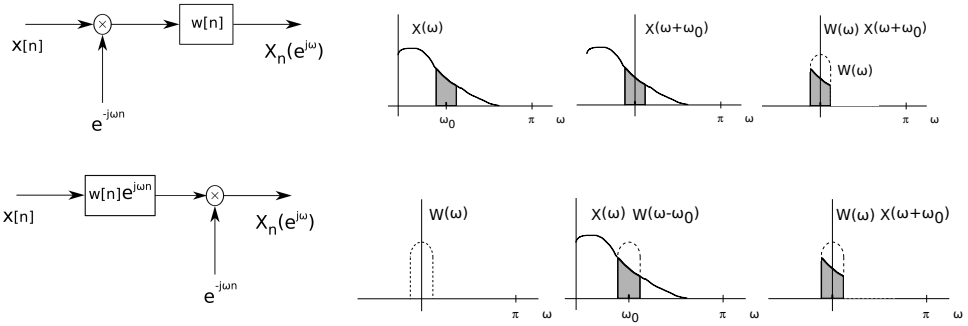
UNIVERSIDAD
DE LA REPÚBLICA
URUGUAY

Análisis de Fourier de tiempo corto (STFT)

Análisis STFT como filtro

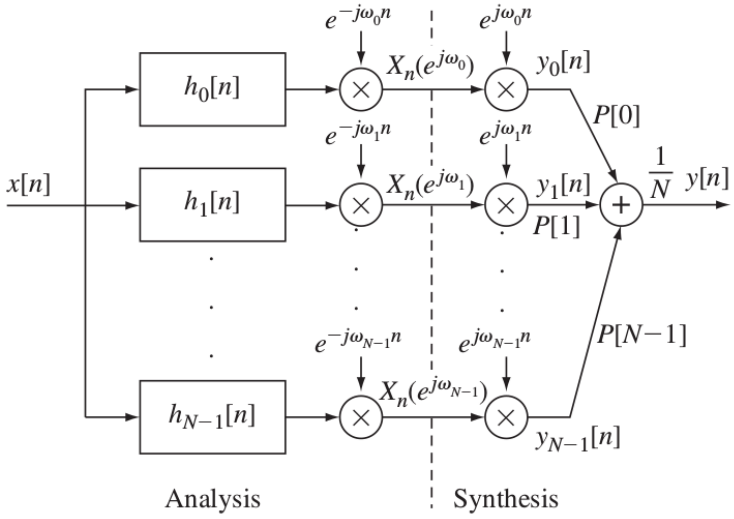
$$X_n(e^{j\omega}) = \sum_{m=-\infty}^{\infty} x[m]w[n-m]e^{-j\omega m} = \left(x[n]e^{-j\omega n}\right) * w[n]$$

$$X_n(e^{j\omega}) = e^{-j\omega n} \sum_{m=-\infty}^{\infty} x[n-m]w[m]e^{j\omega m} = e^{-j\omega n}\left(x[n] * w[n]e^{j\omega n}\right)$$



Análisis de Fourier de tiempo corto (STFT)

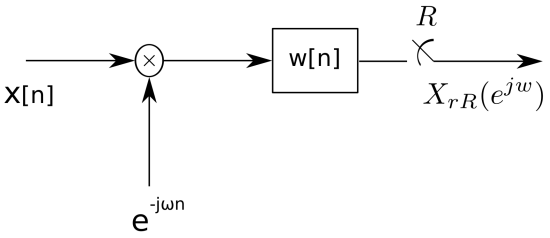
$$X_n(e^{j\omega_k}) = e^{-j\omega_k n} \sum_{m=-\infty}^{\infty} x[n-m]w[m]e^{j\omega_k m} = e^{-j\omega_k n}\Big(x[n] * \underbrace{w[n]e^{j\omega_k n}}_{h_k[n]}\Big)$$

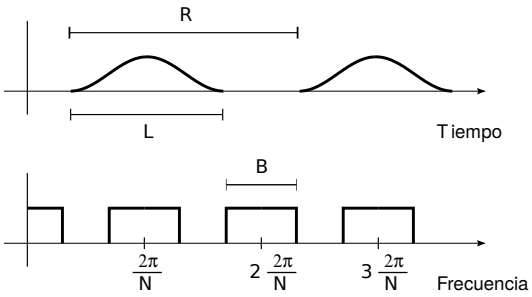


Análisis y síntesis de Fourier de tiempo corto (STFT)

- STFT tiempo-discreto: $x[n] = \frac{1}{2\pi w[0]} \int_{-\pi}^{\pi} X_n(e^{j\omega}) e^{j\omega n} d\omega$
  siempre invertible ($w[0] \neq 0$), muestra a muestra

- STFT discreta: $y[n] = \frac{1}{Nw[0]} \sum_{k=0}^{N-1} X_n(k) e^{j\frac{2\pi}{N}nk}$
  no siempre invertible





- decimación por un factor R, si $R > L \Rightarrow$ no es invertible
- dado $w[n]$ con ancho de banda B, si $\frac{2\pi}{N} > B \Rightarrow$ no es invertible

Análisis y síntesis de Fourier de tiempo corto (STFT)

STFT discreta:

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

no siempre invertible, es necesario derivar condiciones para $y[n] = x[n]$

Dos métodos clásicos de síntesis de la STFT discreta:

- Filter Bank Summation method (FBS)
- Overlap-Add method (OLA)

permiten establecer restricciones sobre el muestreo en frecuencia y la decimación temporal para que la STFT discreta sea invertible.

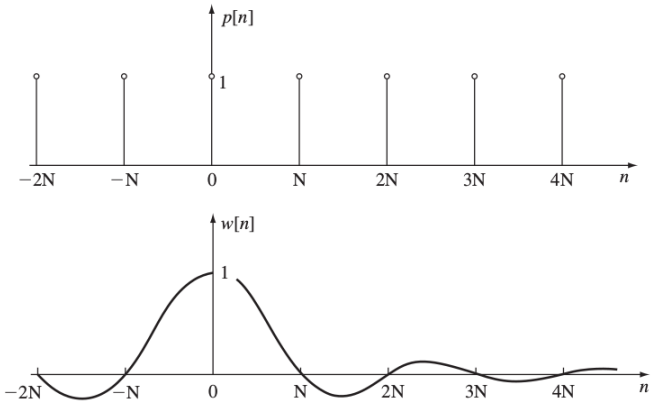
Filter Bank Summation method (FBS)

$$
\begin{aligned}
y[n] &= \frac{1}{Nw[0]} \sum_{k=0}^{N-1} X_n(k) e^{j\frac{2\pi}{N}nk} \\
&= \frac{1}{Nw[0]} \sum_{k=0}^{N-1} \left[\sum_{m=-\infty}^{\infty} x[n-m]w[m]e^{-j\frac{2\pi k}{N}(n-m)} \right] e^{j\frac{2\pi}{N}nk} \\
&= \frac{1}{Nw[0]} \sum_{k=0}^{N-1} \sum_{m=-\infty}^{\infty} x[n-m]w[m]e^{j\frac{2\pi}{N}mk} \\
&= \frac{1}{Nw[0]}\ x[n] * w[n] \sum_{k=0}^{N-1} e^{j\frac{2\pi}{N}nk} = \frac{1}{Nw[0]}\ x[n] * w[n]\ N \sum_{r=-\infty}^{\infty} \delta[n-rN] \\
y[n] &\stackrel{?}{=} x[n] \quad \Rightarrow \quad w[n] \sum_{r=-\infty}^{\infty} \delta[n-rN] = w[0]\delta[n] \quad \Leftarrow \quad L \leq N
\end{aligned}
$$

Filter Bank Summation method (FBS)

$$
L \leq N \quad \Rightarrow \quad w[n] \sum_{r=-\infty}^{\infty} \delta[n-rN] = w[0]\delta[n]
$$

Nota: condición suficiente pero no necesaria.



Filter Bank Summation method (FBS)

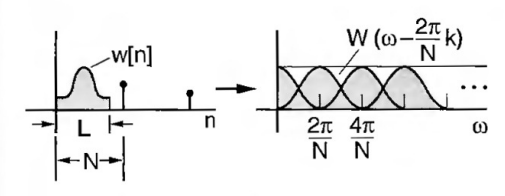
$$y[n] = \frac{1}{Nw[0]} \sum_{k=0}^{N-1} X_n(k) e^{j\frac{2\pi}{N}nk}$$

$$w[0]\delta[n] = w[n] \sum_{r=-\infty}^{\infty} \delta[n - rN] \quad \text{se cumple con } L \leq N$$

tomando la Transformada de Fourier

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

respuesta en frecuencia plana del banco de filtros



Overlap-add method (OLA)

Método IDFT

DFT inversa de cada trama y dividir resultado por la ventana

no es robusto en aplicaciones prácticas (e.g. factor de fase lineal $e^{j\omega_0}$)

Método OLA

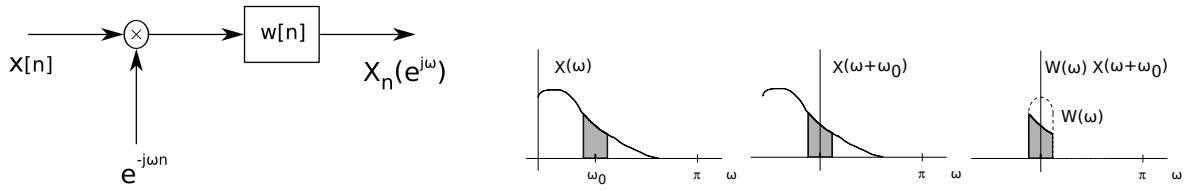
solapamiento y suma de bloques elimina el efecto de la ventana

promediado de muestras redundantes resulta ventajoso

$$x[n] = \frac{1}{2\pi W(e^{j0})} \int_{-\pi}^{\pi} \sum_{r=-\infty}^{\infty} X_r(e^{j\omega}) e^{j\omega n} d\omega \qquad \text{con } W(e^{j0}) = \sum_{n=-\infty}^{\infty} w[n]$$

se puede interpretar como: $x[n] = \frac{1}{2\pi w[0]} \int_{-\pi}^{\pi} X_n(e^{j\omega}) e^{j\omega n} d\omega$ promediado sobre muchos segmentos de tiempo corto y normalizado por $W(e^{j0})$

Overlap-add method (OLA)



$$X_n(e^{j\omega}) = \left(x[n]e^{-j\omega n}\right) * w[n]$$

tomando la Transformada de Fourier a ambos lados y evaluando en frecuencia 0,

$$TF\{X_n(e^{j\omega})\}|_{\theta=0} = X(e^{j(\theta+\omega)})W(e^{j\theta})|_{\theta=0} = X(e^{j\omega})W(e^{j0})$$

lo que corresponde a la suma de todas las muestras en el dominio del tiempo,

$$X(e^{j\omega}) = \frac{1}{W(e^{j0})}\sum_{r=-\infty}^{\infty} X_r(e^{j\omega})$$

Overlap-add method (OLA)

$$X(e^{j\omega}) = \frac{1}{W(e^{j0})}\sum_{r=-\infty}^{\infty} X_r(e^{j\omega})$$

tomando la Transformada de Fourier inversa,

$$x[n] = \frac{1}{2\pi W(e^{j0})}\int_{-\pi}^{\pi}\sum_{r=-\infty}^{\infty} X_r(e^{j\omega})e^{j\omega n}d\omega$$

el método OLA es una versión discretizada,

$$y[n] = \frac{1}{W(e^{j0})}\sum_{r=-\infty}^{\infty}\underbrace{\left[\frac{1}{N}\sum_{k=0}^{N-1} X_r(k)e^{j\frac{2\pi}{N}kn}\right]}_{y_r[n]=x[n]w[r-n]}$$

$$y[n] = \frac{1}{W(e^{j0})}\sum_{r=-\infty}^{\infty} x[n]w[r-n]$$

Overlap-add method (OLA)

$$y[n] = x[n] \; \frac{1}{W(e^{j0})} \sum_{r=-\infty}^{\infty} w[r-n]$$

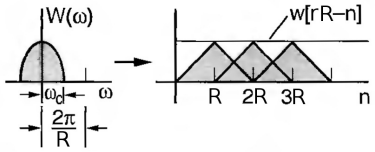
$$y[n] \stackrel{?}{=} x[n] \;\Rightarrow\; \sum_{r=-\infty}^{\infty} w[r-n] = W(e^{j0}) \quad \text{siempre se cumple!}$$

Pero si la STFT discreta está decimada por un factor R

$$y[n] = \frac{R}{W(e^{j0})} \sum_{r=-\infty}^{\infty} x[n]w[rR-n]$$

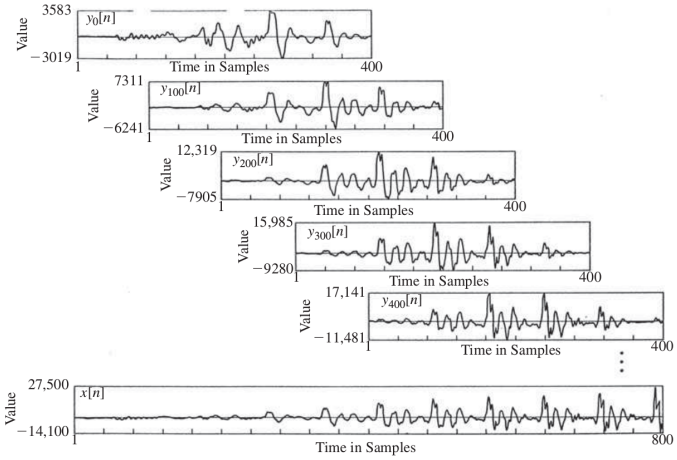
$$y[n] \stackrel{?}{=} x[n] \;\Rightarrow\; \sum_{r=-\infty}^{\infty} w[rR-n] = \frac{W(e^{j0})}{R} \quad \text{NO siempre se cumple!}$$

Para que se cumpla la suma de las ventanas en el tiempo debe ser igual a una constante



Overlap-add method (OLA)

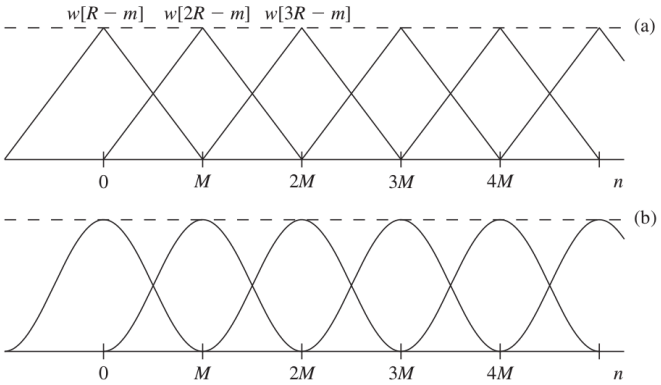
$$y[n] = \sum_{r=-\infty}^{\infty} \Big(\underbrace{\frac{1}{N} \sum_{k=0}^{N-1} X_{rR}(e^{j\omega_k}) e^{j\omega_k m}}_{y_r[m]} \Big) \Big|_{m=n}$$



$L = 400$, $R = 100$

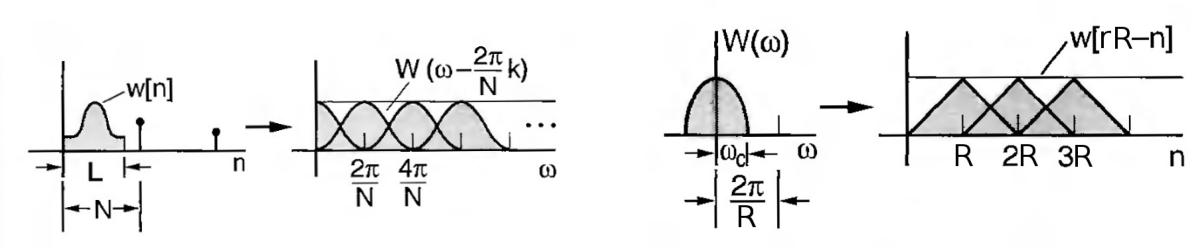
Overlap-add method (OLA)

$$\sum_{r=-\infty}^{\infty} w[rR - n] = \frac{W(e^{j0})}{R} = C$$



(a) Bartlett window (b) Hann window. $L = 2M + 1$, $R = M$, $C = 1$

Dualidad entre FBS y OLA



restricción FBS (muestreo en frecuencia):

$$\frac{1}{N}\sum_{k=0}^{N-1} W(e^{j(\omega - \frac{2\pi}{N}k)}) = w[0]$$

restricción OLA (muestreo en el tiempo):

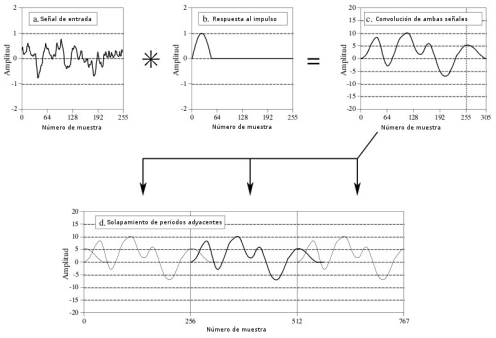
$$\sum_{r=-\infty}^{\infty} w[rR - n] = \frac{W(e^{j0})}{R}$$

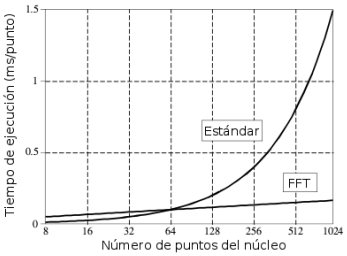
Convolución FFT

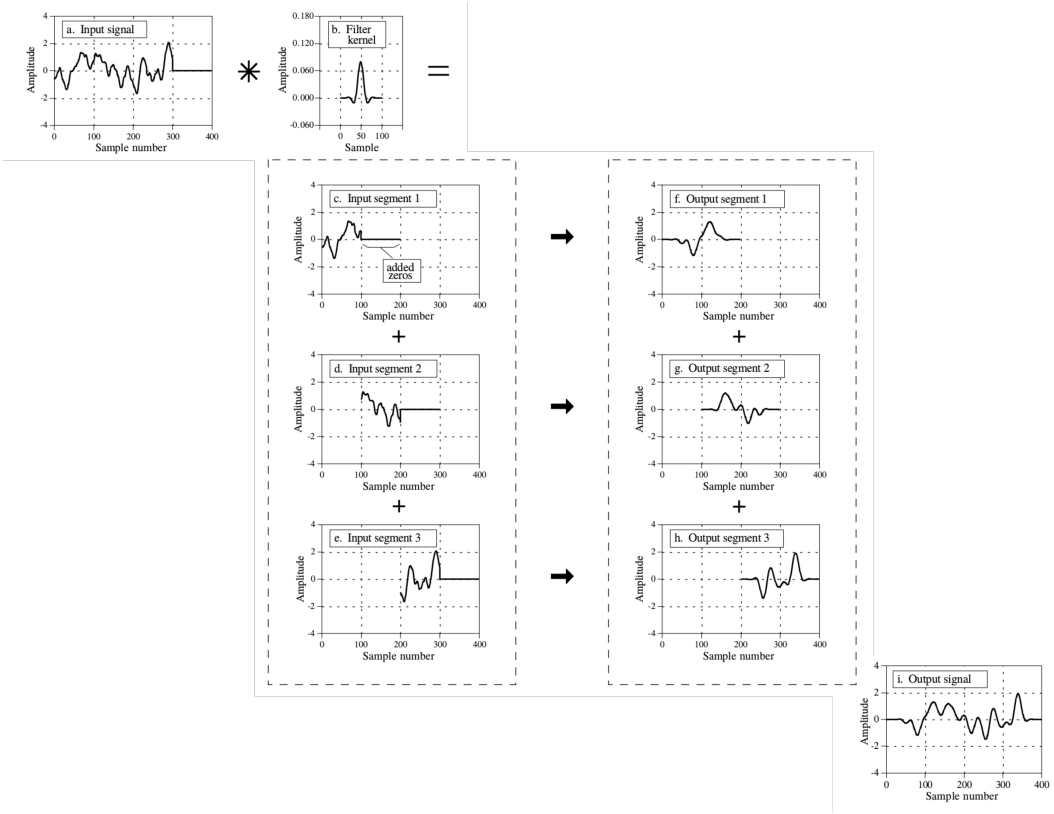
Convolución rápida

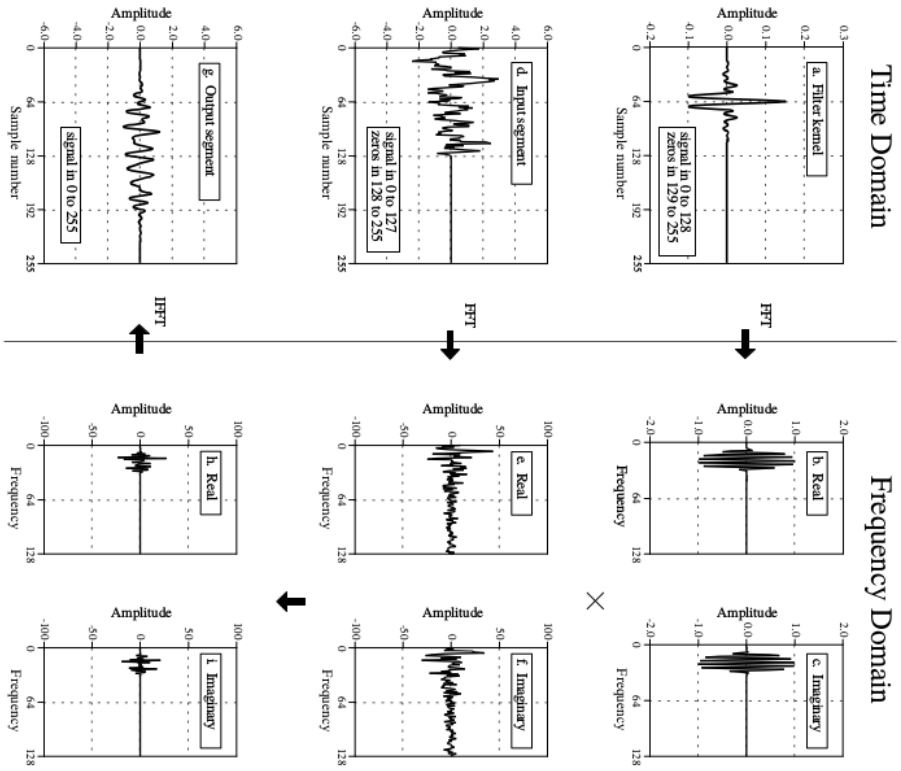
$$y[n] = x[n] * h[n] \iff Y(k) = X(k) \times H(k)$$

- sustituye convolución en el tiempo por producto en frecuencia
- hace uso de FFT y el método overlap-add
- N_{FFT} suficientemente largo para evitar aliasing temporal
- resulta más eficiente que la convolución en el tiempo





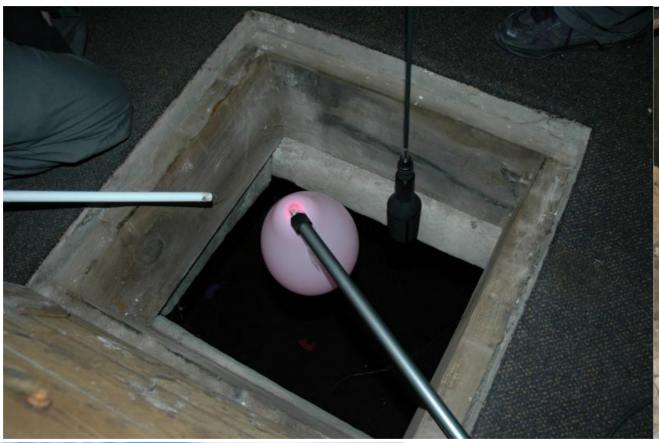


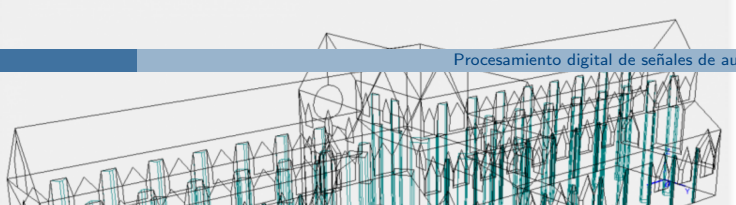


Convolución FFT

Aplicación en auralización

- respuesta al impulso de un espacio sonoro
- convolución rápida para calcular señal reverberada
- simulación de espacios acústicos, reconstrucción virtual, etc.



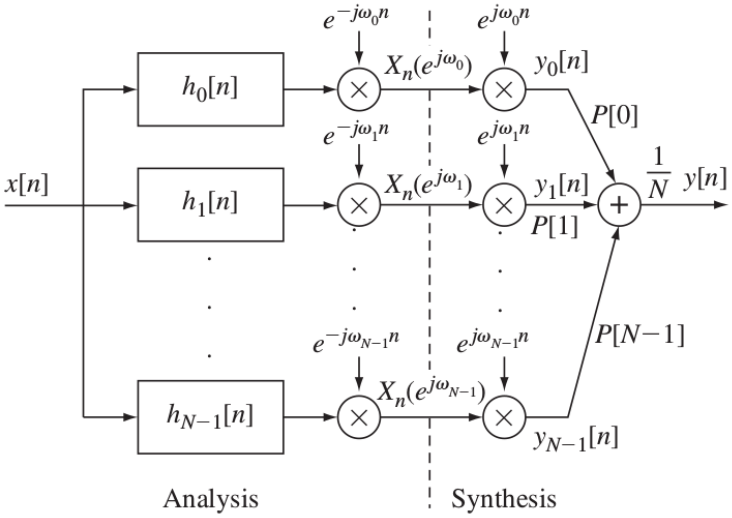
Convolución FFT

Aplicación en audio binaural

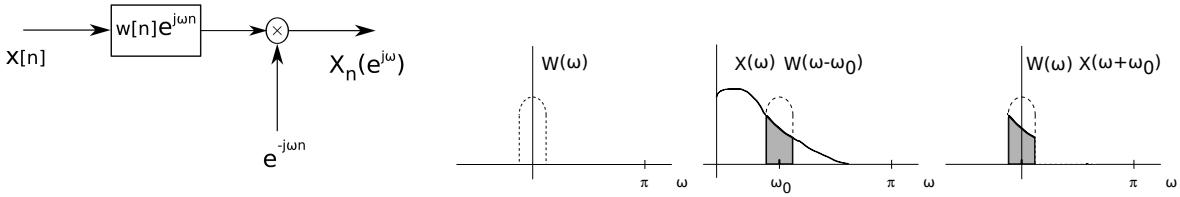
- funciones de transferencia de cabeza y torso (HRTFs)
- convolución rápida para calcular señal binaural (para auriculares)
- simulación de ubicación espacial de sonidos y campo sonoro
- aplicaciones en realidad virtual (VR), sonido 3D, videojuegos, etc.

Análisis de Fourier de tiempo corto (STFT)

$$X_n(e^{j\omega_k}) = e^{-j\omega_k n} \sum_{m=-\infty}^{\infty} x[n-m]w[m]e^{j\omega_k m} = e^{-j\omega_k n}\Big(x[n] * \underbrace{w[n]e^{j\omega_k n}}_{h_k[n]}\Big)$$



Phase vocoder



$$X_n(e^{j\omega}) = e^{-j\omega n}\left(x[n] * w[n]e^{j\omega n}\right)$$

salida de cada filtro antes de demodular

$$y_k[n] = x[n] * h_k[n] \qquad h_k[n] = w[n]e^{j\frac{2\pi}{N}kn}$$

lo que puede formularse como,

$$y_k[n] = a_k[n]e^{j\phi_k[n]} \qquad a_k[n] = |y_k[n]|, \quad \phi_k[n] = \tan^{-1}\left[\frac{Im\{y_k[n]\}}{Re\{y_k n\}}\right]$$

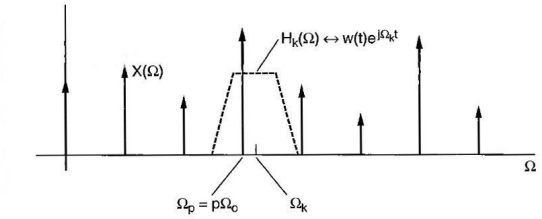
$y[n]$ corresponde a suma de sinusoides complejas,

$$y[n] = \frac{1}{Nw[0]}\sum_{k=0}^{N-1} a_k[n]e^{j\phi_k[n]}$$

Phase vocoder

respuesta del banco de filtros a un armónico:

$$x(t) = A_p\cos(\Omega_p t + \phi_p) \qquad \Omega_p = p\Omega_0, \quad \Omega_0 : \text{ frecuencia fundamental}$$



$$x(t) = \frac{A_p}{2}\left[e^{j(\Omega_p t+\phi_p)} + e^{-j(\Omega_p t+\phi_p)}\right]$$

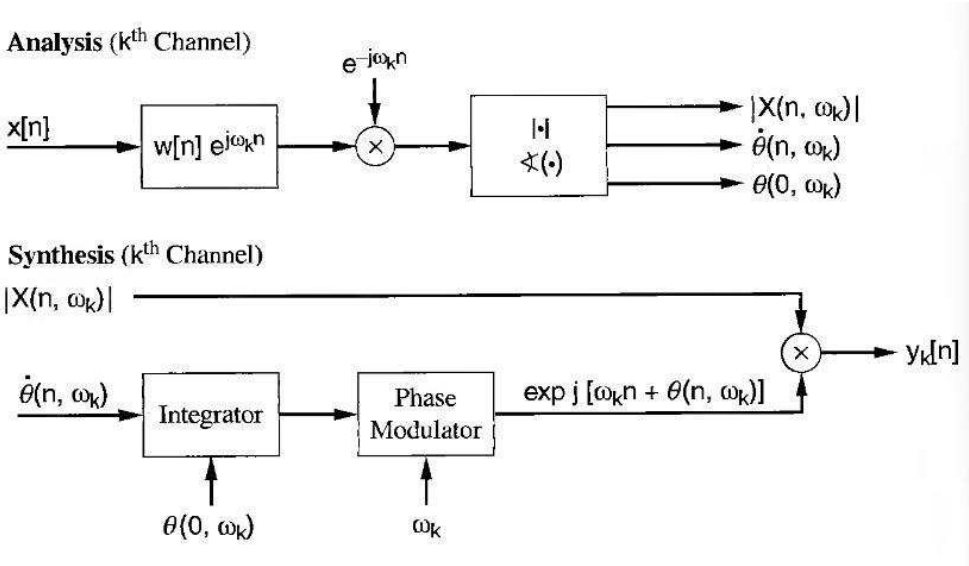
$$X_t(\Omega_k) = \frac{A_p}{2}\, e^{j(\Omega_p t+\phi_p)}\, e^{-j\Omega_k t}$$

$$|X_t(\Omega_k)| = \frac{A_p}{2}, \quad \theta_t(\Omega_k) = (\Omega_p - \Omega_k)t + \theta_0(\Omega_k), \quad \dot{\theta}_t(\Omega_k) = (\Omega_p - \Omega_k)$$

Phase vocoder

en el caso discreto y cuasi-periódico,

$$|X_n(\omega_k)| \approx \frac{A_p}{2}, \quad \dot{\theta}_n(\omega_k) \approx (\omega_p[n] - \omega_k)$$

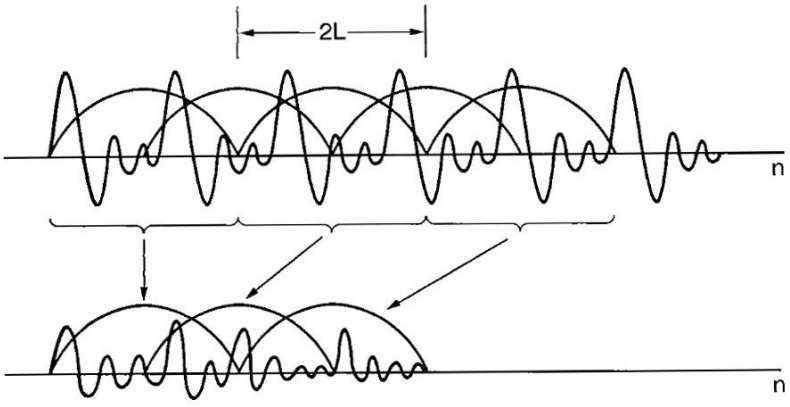


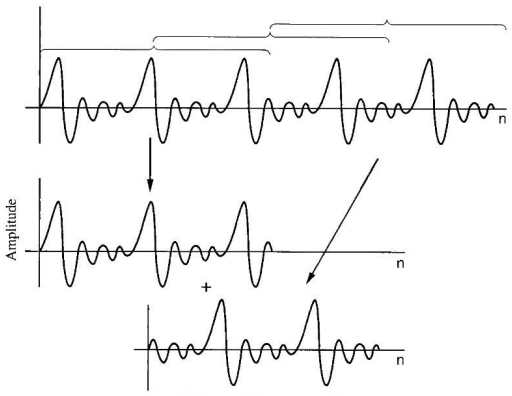
Phase vocoder

Aplicación: modificar escala temporal

Usando STFT tradicional:

- decimar tramas sucesivas y reconstruir con overlap-add
- inconsistencia de fase genera falta de sincronismo de pitch





Phase vocoder

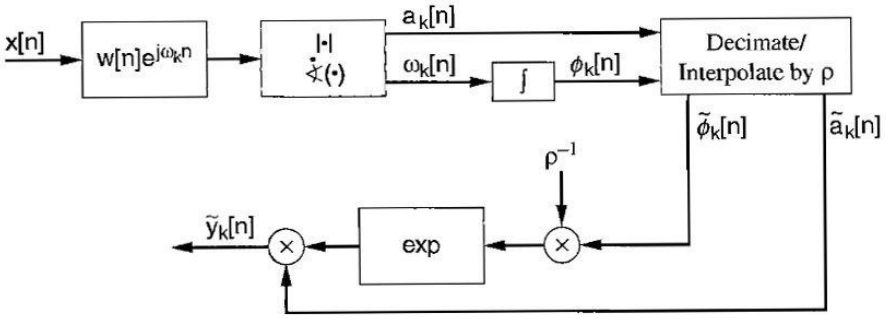
Aplicación: modificar escala temporal

Usando Phase vocoder:

- decimar/interpolar amplitud y fase a nueva escala temporal

$$\tilde{y}_k[n] = \tilde{a}_k[n] \cos(\rho\, \tilde{\Phi}_k[n])$$

- factor de escalado ρ normaliza fase para mantener misma frecuencia



Phase vocoder

Implementación mediante la STFT

- modificar la escala temporal usando un hop R_a de análisis y otro R_s de síntesis
- se impone coherencia de fase usando estimación de frecuencia instantánea

1. diferencia de fase entre tramas sucesivas:

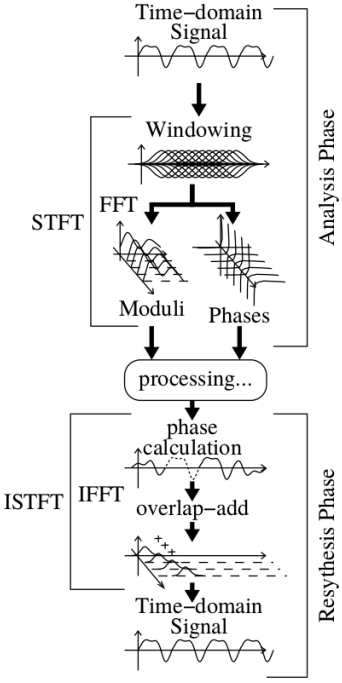
$$\Delta\Phi_k^u = \angle X(t_a^u, \Omega_k) - \angle X(t_a^{u-1}, \Omega_k) - R_a\, \Omega_k$$

2. argumento principal: $\Delta_p \Phi_k^u = \mathrm{princarg}\{\Delta\Phi_k^u\}$
3. estimación de la frecuencia instantanea:

$$\hat{\omega}_k(t_a^u) = \Omega_k + \frac{1}{R_a} \Delta_p \Phi_k^u$$

4. se calcula la fase en la reconstrucción

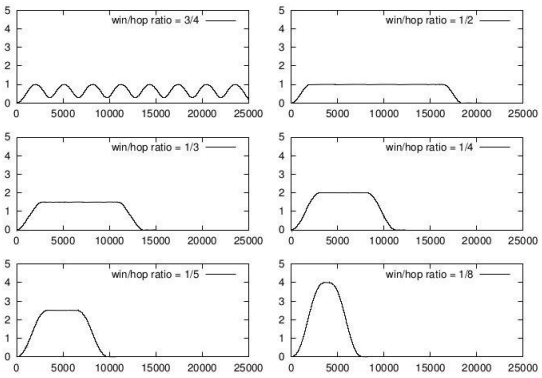
$$\angle Y(t_s^u, \Omega_k) = \angle Y(t_s^{u-1}, \Omega_k) + R_s\, \hat{\omega}_k(t_a^u)$$



Phase vocoder

Consideraciones prácticas

- varios detalles de implementación [Gotzen et al., 2000]:
  - selección de razón R/L de acuerdo a la ventana ($R \leq L$)
  - desdoblamiento de fase (detección de saltos de 2π)
  - escalamiento de amplitud, zero-padding, etc



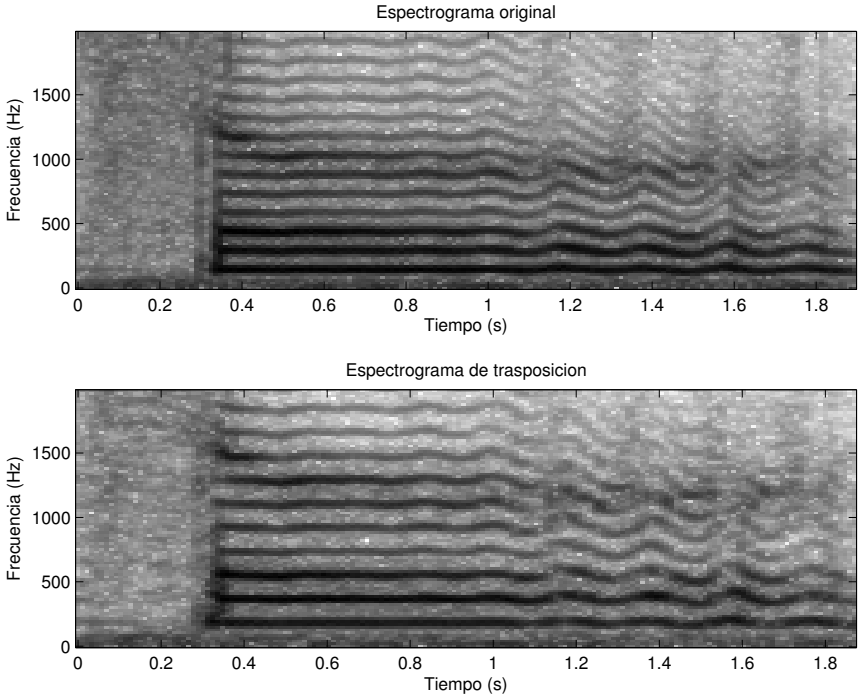
hop/window ratio (R/L) [Gotzen et al., 2000]

Phase vocoder

Consideraciones prácticas

- limitantes:
  - falla cuando no hay un único componente en cada banda
  - o cuando sus variaciones son demasiado rápidas
- numerosas aplicaciones:
  - codificación de voz hablada
  - estudios sobre percepción de timbre
  - efectos con fines musicales
    - time-stretching
    - pitch-shifting
    - harmonization
    - chorus, etc

Phase vocoder



Referencias

Gotzen, A., Bernardin, N., and Arfib., D. (2000).
Traditional implementations of a phase-vocoder: The tricks of the trade.
In *Int. Conf. on Digital Audio Effects, Italy*.

Quatieri, T. F. (2001).
Discrete-Time Speech Signal Processing: Principles and Practice.
Prentice Hall PTR, 1st edition.

Rabiner, L. R. and Schafer, R. W. (2011).
Theory and Applications of Digital Speech Processing.
Prentice Hall, 1st edition.
Chapter 7 - Frequency-domain representations.

Smith, S. W. (1997).
The Scientist and Engineer's Guide to Digital Signal Processing.
California Technical Publishing.