

Análisis Tiempo-Frecuencia y Descomposición de Señales

Clase 6

Dr. Marcelo Alejandro Colominas
macolominas@conicet.gov.ar

Lab. Señales y Dinámicas no Lineales
Instituto de Investigación y Desarrollo en Bioingeniería y Bioinformática (IBB)
Facultad de Ingeniería, Universidad Nacional de Entre Ríos, Argentina

2019



Motivación
oo

Detección de crestas
ooooooo

Reassignment vector
ooo

Extracción de modos
ooo

Umbralado
oooo

Model Fitting
ooooo

Contenidos

- 1 Motivación
- 2 Detección de crestas
- 3 Reassignment vector
- 4 Extracción de modos
- 5 Umbralado
- 6 Model Fitting

Contenidos

1 Motivación

2 Detección de crestas

3 Reassignment vector

4 Extracción de modos

5 Umbralado

6 Model Fitting

- Vimos que para una señal de la forma $x(t) = e^{i2\pi\phi(t)}$, con $|\phi''(t)| < \epsilon$, se tenía que $F_x^g(t, f) \approx x(t)\hat{g}(f - \phi'(t))$.

- Vimos que para una señal de la forma $x(t) = e^{i2\pi\phi(t)}$, con $|\phi''(t)| < \epsilon$, se tenía que $F_x^g(t, f) \approx x(t)\hat{g}(f - \phi'(t))$.
- También, para $x(t) = e^{i2\pi(at+bt^2)}$ (chirp lineal), tenemos que $F_x^g(t, f) = x(t)\hat{g}_{\phi''}(f - \phi'(t))$.

- Vimos que para una señal de la forma $x(t) = e^{i2\pi\phi(t)}$, con $|\phi''(t)| < \epsilon$, se tenía que $F_x^g(t, f) \approx x(t)\hat{g}(f - \phi'(t))$.
- También, para $x(t) = e^{i2\pi(at+bt^2)}$ (chirp lineal), tenemos que $F_x^g(t, f) = x(t)\hat{g}_{\phi''}(f - \phi'(t))$.
- Esto indica que el módulo de la STFT tiene una «cresta» (máximo local) en la frecuencia instantánea.

- Vimos que para una señal de la forma $x(t) = e^{i2\pi\phi(t)}$, con $|\phi''(t)| < \epsilon$, se tenía que $F_x^g(t, f) \approx x(t)\hat{g}(f - \phi'(t))$.
- También, para $x(t) = e^{i2\pi(at+bt^2)}$ (chirp lineal), tenemos que $F_x^g(t, f) = x(t)\hat{g}_{\phi''}(f - \phi'(t))$.
- Esto indica que el módulo de la STFT tiene una «cresta» (máximo local) en la frecuencia instantánea.
- Encontrar la cresta lleva entonces a una estimación de la frecuencia instantánea.

- Vimos que para una señal de la forma $x(t) = e^{i2\pi\phi(t)}$, con $|\phi''(t)| < \epsilon$, se tenía que $F_x^g(t, f) \approx x(t)\hat{g}(f - \phi'(t))$.
- También, para $x(t) = e^{i2\pi(at+bt^2)}$ (chirp lineal), tenemos que $F_x^g(t, f) = x(t)\hat{g}_{\phi''}(f - \phi'(t))$.
- Esto indica que el módulo de la STFT tiene una «cresta» (máximo local) en la frecuencia instantánea.
- Encontrar la cresta lleva entonces a una estimación de la frecuencia instantánea.
- Además, la detección de crestas sirve para la separación en modos (descomposición).

Motivación
oo

Detección de crestas
●oooooooo

Reassignment vector
ooo

Extracción de modos
ooo

Umbralado
oooo

Model Fitting
ooooo

Contenidos

1 Motivación

2 Detección de crestas

3 Reassignment vector

4 Extracción de modos

5 Umbralado

6 Model Fitting

El modelo multicomponente

$$x(t) = \sum_{k=1}^K x_k(t) = \sum_{k=1}^K A_k(t) \cos(2\pi\phi_k(t)),$$

El modelo multicomponente

$$x(t) = \sum_{k=1}^K x_k(t) = \sum_{k=1}^K A_k(t) \cos(2\pi\phi_k(t)),$$

con:

- $|A'_k(t)| < \epsilon$

El modelo multicomponente

$$x(t) = \sum_{k=1}^K x_k(t) = \sum_{k=1}^K A_k(t) \cos(2\pi\phi_k(t)),$$

con:

- $|A'_k(t)| < \epsilon$
- $|\phi''_k(t)| < \epsilon$ (a veces)

El modelo multicomponente

$$x(t) = \sum_{k=1}^K x_k(t) = \sum_{k=1}^K A_k(t) \cos(2\pi\phi_k(t)),$$

con:

- $|A'_k(t)| < \epsilon$
- $|\phi''_k(t)| < \epsilon$ (a veces)
- $\phi'_k(t) > 0$

El modelo multicomponente

$$x(t) = \sum_{k=1}^K x_k(t) = \sum_{k=1}^K A_k(t) \cos(2\pi\phi_k(t)),$$

con:

- $|A'_k(t)| < \epsilon$
- $|\phi''_k(t)| < \epsilon$ (a veces)
- $\phi'_k(t) > 0$
- $|\phi'_l(t) - \phi'_k(t)| > 2B$, para $k \neq l$ ($\text{supp}\{\hat{g}\} = [-B, B]$).

Motivación
○○

Detección de crestas
○○●○○○○

Reassignment vector
○○○

Extracción de modos
○○○

Umbralizado
○○○○

Model Fitting
○○○○○

STFT

Si $x(t)$ cumple con el modelo, entonces

$$F_x^g(t, f) = \sum_{k=1}^K F_{x_k}^g(t, f) \approx \sum_{k=1}^K A_k \cos(2\pi\phi(t))\hat{g}(f - \phi'(t))$$

Motivación
○○

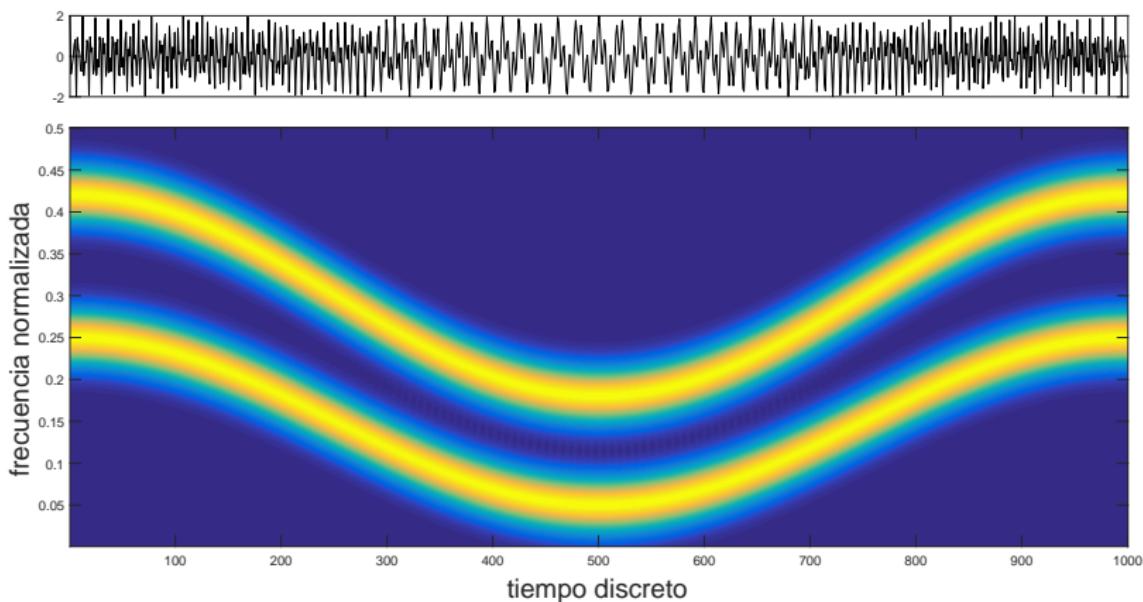
Detección de crestas
○○○●○○○○

Reassignment vector
○○○

Extracción de modos
○○○

Umbralado
○○○○

Model Fitting
○○○○○



Motivación
○○

Detección de crestas
○○○○●○○○

Reassignment vector
○○○

Extracción de modos
○○○

Umbralado
○○○○

Model Fitting
○○○○○

Detección de cresta

$$\max_{\mathcal{C}} \sum_{k=1}^K \int (|F_x^g(t, c_k(t))|^2 - \alpha(c'_k(t))^2 - \beta(c''_k(t))^2) dt$$

Detección de cresta

$$\max_c \sum_{k=1}^K \int (|F_x^g(t, c_k(t))|^2 - \alpha(c'_k(t))^2 - \beta(c''_k(t))^2) dt$$

Se pueden usar otras funciones monótonas del módulo de la STFT.

Detección de cresta

$$\max_{\mathcal{C}} \sum_{k=1}^K \int (|F_x^g(t, c_k(t))|^2 - \alpha(c'_k(t))^2 - \beta(c''_k(t))^2) dt$$

Se pueden usar otras funciones monótonas del módulo de la STFT.

Solución voraz

Considerando que para n ya se tiene una solución óptima,

Detección de cresta

$$\max_c \sum_{k=1}^K \int (|F_x^g(t, c_k(t))|^2 - \alpha(c'_k(t))^2 - \beta(c''_k(t))^2) dt$$

Se pueden usar otras funciones monótonas del módulo de la STFT.

Solución voraz

Considerando que para n ya se tiene una solución óptima,

$$c_k[n+1] = \arg \max_p |F_x^g[n+1, p]|^2 - \alpha(p - c_k[n])^2 - \beta(p - 2c[n] + c[n-1])^2,$$

Detección de cresta

Otro enfoque

$$\max_{\mathcal{C}} \sum_{k=1}^K \int |F_x^g(t, c_k)|^2 dt$$

s.a. $|c'_k(t)| < \epsilon_k$

Detección de cresta

Otro enfoque

$$\max_{\mathcal{C}} \sum_{k=1}^K \int |F_x^g(t, c_k)|^2 dt$$

s.a. $|c'_k(t)| < \epsilon_k$

Solución voraz

Considerando que para n ya se tiene una solución óptima,

Detección de cresta

Otro enfoque

$$\max_{\mathcal{C}} \sum_{k=1}^K \int |F_x^g(t, c_k)|^2 dt$$

$$\text{s.a. } |c'_k(t)| < \epsilon_k$$

Solución voraz

Considerando que para n ya se tiene una solución óptima,

$$c_k[n+1] = \arg \max_{p \in I_k} |F_x^g[n+1, p]|, \quad \text{con } I_k = [c[n] - \epsilon_k, c[n] + \epsilon_k]$$

Detección de cresta

Más enfoques

- También pueden combinarse ambos enfoques: mantener las constantes α y β , limitando el tamaño del «salto».

Detección de cresta

Más enfoques

- También pueden combinarse ambos enfoques: mantener las constantes α y β , limitando el tamaño del «salto».
- Debido a la no-causalidad del método, la cresta puede recorrerse tanto de izquierda a derecha como viceversa.

Detección de cresta

Más enfoques

- También pueden combinarse ambos enfoques: mantener las constantes α y β , limitando el tamaño del «salto».
- Debido a la no-causalidad del método, la cresta puede recorrerse tanto de izquierda a derecha como viceversa.
- En la práctica, se resuelven P problemas con inicialización aleatoria. Para cada «semilla», se recorre la cresta en ambas direcciones. Se logra un conjunto $\{c_{k,p}\}_{p=1}^P$, y se elige la $c_{k,\tilde{p}}$ que maximice el problema original.

Detección de cresta

Más enfoques

- También pueden combinarse ambos enfoques: mantener las constantes α y β , limitando el tamaño del «salto».
- Debido a la no-causalidad del método, la cresta puede recorrerse tanto de izquierda a derecha como viceversa.
- En la práctica, se resuelven P problemas con inicialización aleatoria. Para cada «semilla», se recorre la cresta en ambas direcciones. Se logra un conjunto $\{c_{k,p}\}_{p=1}^P$, y se elige la $c_{k,\tilde{p}}$ que maximice el problema original.
- En el caso multicomponente, se resuelve el problema de manera «deflacionaria». Encontramos primero una cresta c_1 , y luego seguimos con la diferencia entre $F_x^g[n, k]$ y la franja $\{c_1[n] - b, c_1[n] + b\}$.

Motivación
○○

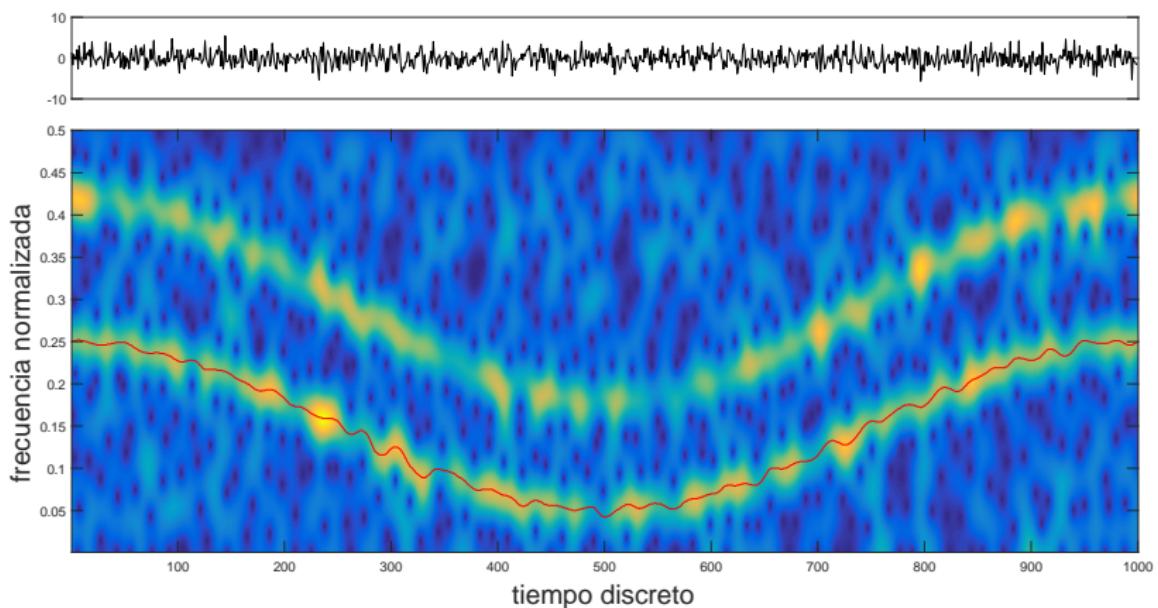
Detección de crestas
○○○○○○●

Reassignment vector
○○○

Extracción de modos
○○○

Umbralizado
○○○○

Model Fitting
○○○○○



Motivación
○○

Detección de crestas
○○○○○○○○

Reassignment vector
●○○

Extracción de modos
○○○

Umbralado
○○○○

Model Fitting
○○○○○

Contenidos

1 Motivación

2 Detección de crestas

3 Reassignment vector

4 Extracción de modos

5 Umbralado

6 Model Fitting

Vector de *reassignment*

$$\vec{v}(t, f) = \begin{bmatrix} f - \tilde{f}(t, f) \\ t - \tilde{t}(t, f) \end{bmatrix}$$

Motivación
○○

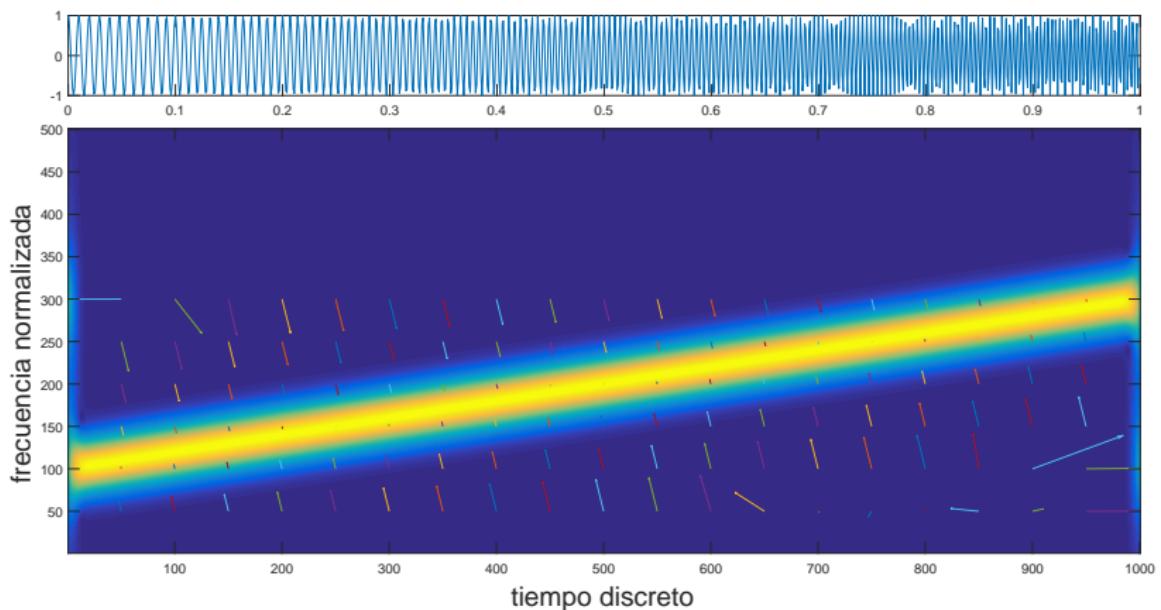
Detección de crestas
○○○○○○○○

Reassignment vector
○○●

Extracción de modos
○○○

Umbralado
○○○○

Model Fitting
○○○○○



Motivación
○○

Detección de crestas
○○○○○○○○

Reassignment vector
○○○

Extracción de modos
●○○

Umbralado
○○○○

Model Fitting
○○○○○

Contenidos

1 Motivación

2 Detección de crestas

3 Reassignment vector

4 Extracción de modos

5 Umbralado

6 Model Fitting

Motivación
○○

Detección de crestas
○○○○○○○○

Reassignment vector
○○○

Extracción de modos
○●○

Umbralizado
○○○○

Model Fitting
○○○○○

Una vez obtenida la cresta c_k , podemos extraer el modo:

Una vez obtenida la cresta c_k , podemos extraer el modo:

$$x_k[n] \approx \frac{1}{g(0)} \sum_{p=c_k[n]-b_k^-[n]}^{c_k[n]+b_k^+[n]} F_x^g[n, p]$$

Una vez obtenida la cresta c_k , podemos extraer el modo:

$$x_k[n] \approx \frac{1}{g(0)} \sum_{p=c_k[n]-b_k^-[n]}^{c_k[n]+b_k^+[n]} F_x^g[n, p]$$

- Podemos definir $b_k^-[n] = b_k^+[n] = b$ (constante) basándonos en el soporte de \hat{g} .

Una vez obtenida la cresta c_k , podemos extraer el modo:

$$x_k[n] \approx \frac{1}{g(0)} \sum_{p=c_k[n]-b_k^-[n]}^{c_k[n]+b_k^+[n]} F_x^g[n, p]$$

- Podemos definir $b_k^-[n] = b_k^+[n] = b$ (constante) basándonos en el soporte de \hat{g} .
- O podemos intentar estimarlas de alguna manera.

Motivación
○○

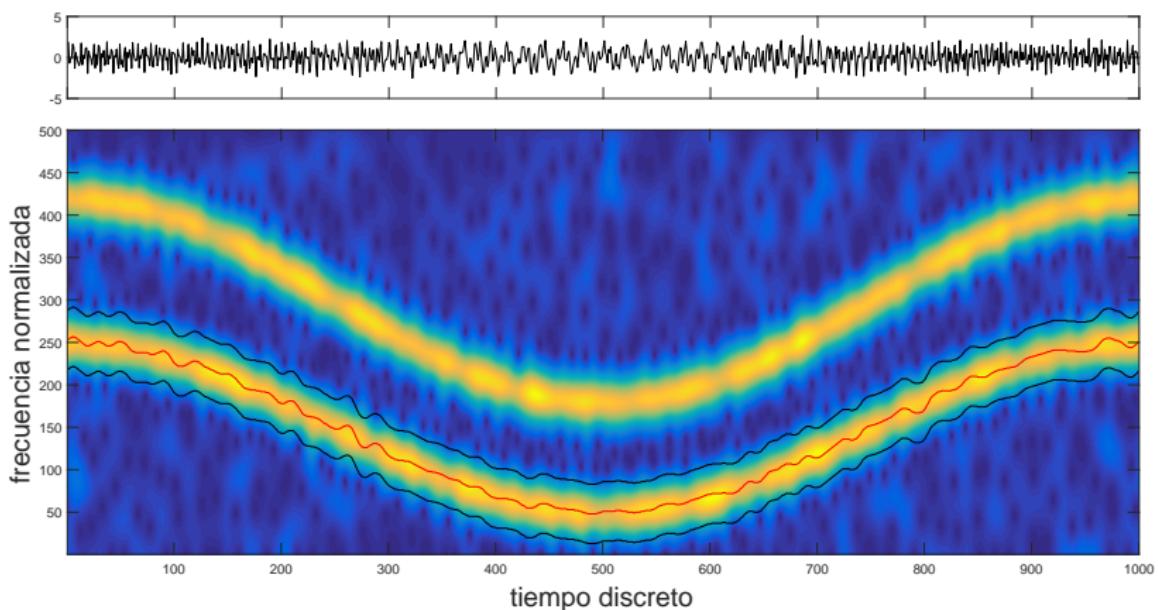
Detección de crestas
○○○○○○○○

Reassignment vector
○○○

Extracción de modos
○○●

Umbralado
○○○○

Model Fitting
○○○○○



Motivación
○○

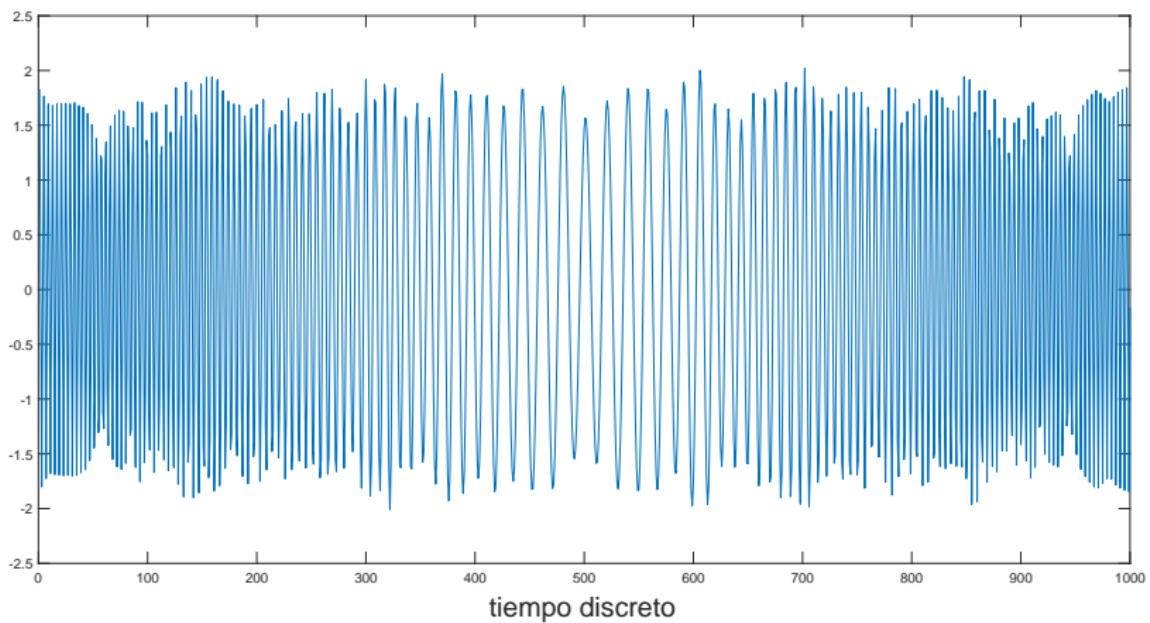
Detección de crestas
○○○○○○○○

Reassignment vector
○○○

Extracción de modos
○○●

Umbralizado
○○○○

Model Fitting
○○○○○



Contenidos

1 Motivación

2 Detección de crestas

3 Reassignment vector

4 Extracción de modos

5 Umbralado

6 Model Fitting

Motivación
○○

Detección de crestas
○○○○○○○○

Reassignment vector
○○○

Extracción de modos
○○○

Umbralado
○●○○

Model Fitting
○○○○○

Sea $x[n] = x_R[n] + x_I[n]$, con $x_R \sim \mathcal{N}(0, \sigma)$ y $x_I \sim \mathcal{N}(0, \sigma)$.

Sea $x[n] = x_R[n] + x_I[n]$, con $x_R \sim \mathcal{N}(0, \sigma)$ y $x_I \sim \mathcal{N}(0, \sigma)$.
Entonces:

$$\text{var}\{Re(F_x^g[n, k])\} = \text{var}\{Im(F_x^g[n, k])\} = \sigma^2 \|g\|_2^2$$

Sea $x[n] = x_R[n] + x_I[n]$, con $x_R \sim \mathcal{N}(0, \sigma)$ y $x_I \sim \mathcal{N}(0, \sigma)$.
Entonces:

$$\text{var}\{Re(F_x^g[n, k])\} = \text{var}\{Im(F_x^g[n, k])\} = \sigma^2 \|g\|_2^2$$

Si las partes real e imaginaria tienen distribución normal, entonces el módulo de la STFT tiene distribución χ (chi).

Sea $x[n] = x_R[n] + x_I[n]$, con $x_R \sim \mathcal{N}(0, \sigma)$ y $x_I \sim \mathcal{N}(0, \sigma)$.
Entonces:

$$\text{var}\{Re(F_x^g[n, k])\} = \text{var}\{Im(F_x^g[n, k])\} = \sigma^2 \|g\|_2^2$$

Si las partes real e imaginaria tienen distribución normal, entonces el módulo de la STFT tiene distribución χ (chi). Podemos umbralizar la STFT del siguiente modo:

$$\widetilde{F}_x^g(t, f) = \begin{cases} F_x^g(t, f) & \text{si } |F_x^g(t, f)| > Q\sigma \|g\|_2 \\ 0, & \text{en otro caso} \end{cases}$$

Sea $x[n] = x_R[n] + x_I[n]$, con $x_R \sim \mathcal{N}(0, \sigma)$ y $x_I \sim \mathcal{N}(0, \sigma)$.
Entonces:

$$\text{var}\{Re(F_x^g[n, k])\} = \text{var}\{Im(F_x^g[n, k])\} = \sigma^2 \|g\|_2^2$$

Si las partes real e imaginaria tienen distribución normal, entonces el módulo de la STFT tiene distribución χ (chi). Podemos umbralizar la STFT del siguiente modo:

$$\widetilde{F}_x^g(t, f) = \begin{cases} F_x^g(t, f) & \text{si } |F_x^g(t, f)| > Q\sigma \|g\|_2 \\ 0, & \text{en otro caso} \end{cases}$$

Eligiendo $Q = 3$, tenemos un 99 % de confianza.

Sea $x[n] = x_R[n] + x_I[n]$, con $x_R \sim \mathcal{N}(0, \sigma)$ y $x_I \sim \mathcal{N}(0, \sigma)$.
 Entonces:

$$\text{var}\{Re(F_x^g[n, k])\} = \text{var}\{Im(F_x^g[n, k])\} = \sigma^2 \|g\|_2^2$$

Si las partes real e imaginaria tienen distribución normal, entonces el módulo de la STFT tiene distribución χ (chi). Podemos umbralizar la STFT del siguiente modo:

$$\widetilde{F}_x^g(t, f) = \begin{cases} F_x^g(t, f) & \text{si } |F_x^g(t, f)| > Q\sigma \|g\|_2 \\ 0, & \text{en otro caso} \end{cases}$$

Eligiendo $Q = 3$, tenemos un 99 % de confianza.

Con $Q = 3,4086$, tenemos un 99,7 % de confianza.

($3.4086 = \text{sqrt}(2 * \text{gammaincinv}(.997, 1))$ en MATLAB).

Sea $x[n] = x_R[n] + x_I[n]$, con $x_R \sim \mathcal{N}(0, \sigma)$ y $x_I \sim \mathcal{N}(0, \sigma)$.
Entonces:

$$\text{var}\{Re(F_x^g[n, k])\} = \text{var}\{Im(F_x^g[n, k])\} = \sigma^2 \|g\|_2^2$$

Si las partes real e imaginaria tienen distribución normal, entonces el módulo de la STFT tiene distribución χ (chi). Podemos umbralizar la STFT del siguiente modo:

$$\widetilde{F}_x^g(t, f) = \begin{cases} F_x^g(t, f) & \text{si } |F_x^g(t, f)| > Q\sigma \|g\|_2 \\ 0, & \text{en otro caso} \end{cases}$$

Eligiendo $Q = 3$, tenemos un 99 % de confianza.

Con $Q = 3,4086$, tenemos un 99,7 % de confianza.

($3,4086 = \text{sqrt}(2 * \text{gammaincinv}(.997, 1))$ en MATLAB).

En la práctica, estimamos el desvío estándar como

$$\sigma \|g\|_2 = \text{median}|(Re(F_x^g))| / 0,6475$$

- Podemos usar el umbralado para limpieza de ruido.

- Podemos usar el umbralado para limpieza de ruido.
- También podemos usarlo para extracción de modos.

- Podemos usar el umbralado para limpieza de ruido.
- También podemos usarlo para extracción de modos.
- Podemos definir el «dominio» de cada modo como la región que rodea a una cresta, y se mantiene por encima del umbral.

Motivación
○○

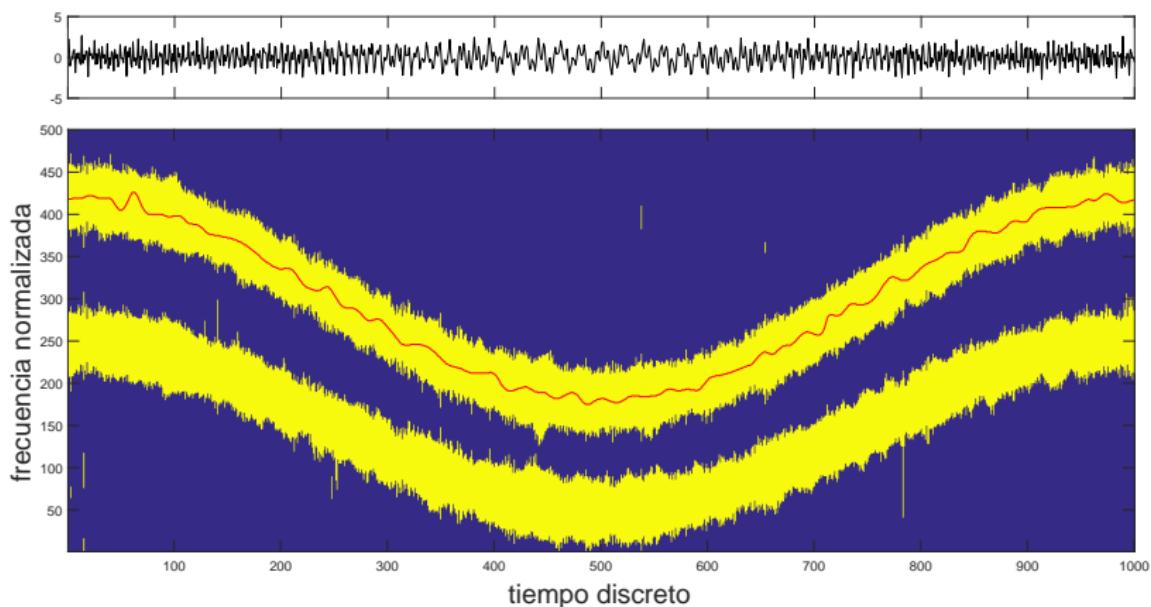
Detección de crestas
○○○○○○○○

Reassignment vector
○○○

Extracción de modos
○○○

Umbralizado
○○○●

Model Fitting
○○○○○



Motivación
oo

Detección de crestas
ooooooo

Reassignment vector
ooo

Extracción de modos
ooo

Umbralado
oooo

Model Fitting
●oooo

Contenidos

1 Motivación

2 Detección de crestas

3 Reassignment vector

4 Extracción de modos

5 Umbralado

6 Model Fitting

Motivación
○○

Detección de crestas
○○○○○○○○

Reassignment vector
○○○

Extracción de modos
○○○

Umbralizado
○○○○

Model Fitting
○●○○○

La extracción de cresta tiene una desventaja.

La extracción de cresta tiene una desventaja.

Se hace sobre una «grilla» predeterminada que es inevitable debido a la implementación en frecuencias discretas.

La extracción de cresta tiene una desventaja.

Se hace sobre una «grilla» predeterminada que es inevitable debido a la implementación en frecuencias discretas.

Por lo tanto, la estimación de frecuencia instantánea que se obtiene resulta una función constante a trozos.

La extracción de cresta tiene una desventaja.

Se hace sobre una «grilla» predeterminada que es inevitable debido a la implementación en frecuencias discretas.

Por lo tanto, la estimación de frecuencia instantánea que se obtiene resulta una función constante a trozos.

Ocurre que estamos tratando de estimar una función que toma valores reales (la frecuencia instantánea) mediante una función que toma valores naturales (la cresta).

Model Fitting

$$\rho(f, \tilde{\phi}'(t), \tilde{\phi}''(t)) = \sqrt{\frac{\pi}{\sqrt{\sigma^2 + \pi^2 \tilde{\phi}''(t)^2}}} e^{\frac{-\sigma\pi^2(f - \tilde{\phi}'(t))^2}{\sigma^2 + \pi^2 \tilde{\phi}''(t)^2}}, \quad (1)$$

Model Fitting

$$\rho(f, \tilde{\phi}'(t), \tilde{\phi}''(t)) = \sqrt{\frac{\pi}{\sqrt{\sigma^2 + \pi^2 \tilde{\phi}''(t)^2}}} e^{\frac{-\sigma\pi^2(f - \tilde{\phi}'(t))^2}{\sigma^2 + \pi^2 \tilde{\phi}''(t)^2}}, \quad (1)$$

$$\max_{\Phi} \sum_{l=1}^L \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} |F_x^g(t, f)| \rho(f, \tilde{\phi}_l'(t), \tilde{\phi}_l''(t)) df dt, \quad (2)$$

Model Fitting

$$\rho(f, \tilde{\phi}'(t), \tilde{\phi}''(t)) = \sqrt{\frac{\pi}{\sqrt{\sigma^2 + \pi^2 \tilde{\phi}''(t)^2}}} e^{\frac{-\sigma\pi^2(f - \tilde{\phi}'(t))^2}{\sigma^2 + \pi^2 \tilde{\phi}''(t)^2}}, \quad (1)$$

$$\max_{\Phi} \sum_{l=1}^L \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} |F_x^g(t, f)| \rho(f, \tilde{\phi}_l'(t), \tilde{\phi}_l''(t)) df dt, \quad (2)$$

donde $\Phi := (\tilde{\phi}_l', \tilde{\phi}_l'')_{l=1,\dots,L}$ son las IFs y valores absolutos de las CRs, respectivamente.

Model Fitting

$$\rho(f, \tilde{\phi}'(t), \tilde{\phi}''(t)) = \sqrt{\frac{\pi}{\sqrt{\sigma^2 + \pi^2 \tilde{\phi}''(t)^2}}} e^{\frac{-\sigma\pi^2(f - \tilde{\phi}'(t))^2}{\sigma^2 + \pi^2 \tilde{\phi}''(t)^2}}, \quad (1)$$

$$\max_{\Phi} \sum_{l=1}^L \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} |F_x^g(t, f)| \rho(f, \tilde{\phi}_l'(t), \tilde{\phi}_l''(t)) df dt, \quad (2)$$

donde $\Phi := (\tilde{\phi}_l', \tilde{\phi}_l'')_{l=1,\dots,L}$ son las IFs y valores absolutos de las CRs, respectivamente.

En el caso discreto, el conjunto Φ está compuesto de $\tilde{\phi}_l', \tilde{\phi}_l'' : \{1, \dots, N\} \mapsto \mathbb{R}$.

Motivación
○○

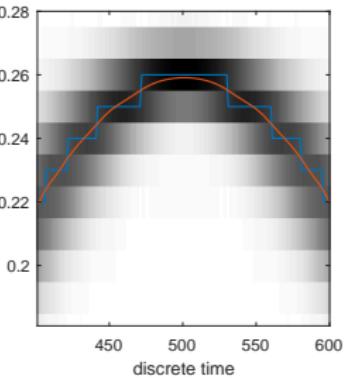
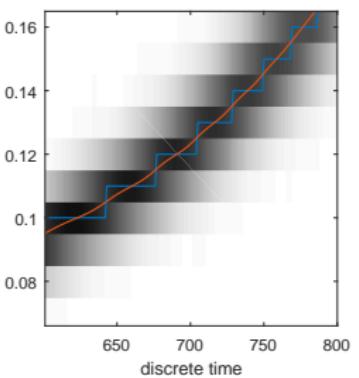
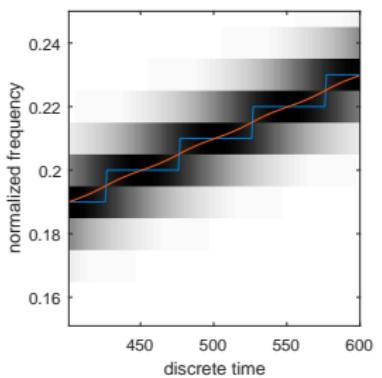
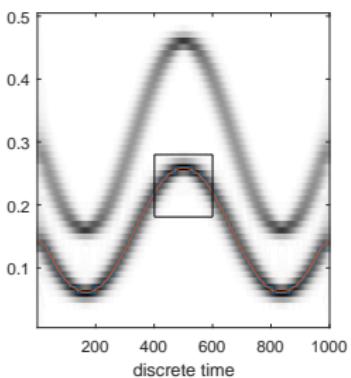
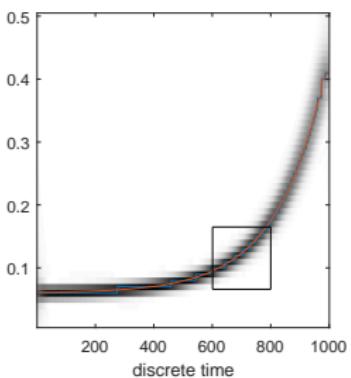
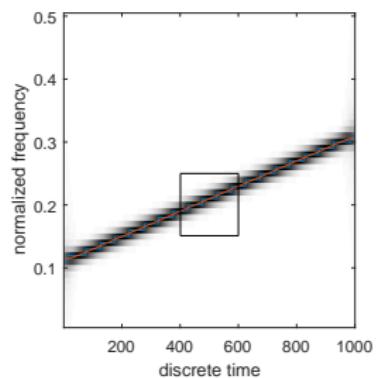
Detección de crestas
○○○○○○○○

Reassignment vector
○○○

Extracción de modos
○○○

Umbralizado
○○○○

Model Fitting
○○○●○



Motivación
○○

Detección de crestas
○○○○○○○○

Reassignment vector
○○○

Extracción de modos
○○○

Umbralizado
○○○○

Model Fitting
○○○○●

