4RBI08

Traitement du signal audio

Henri Boutin
boutin@ircam.fr

2020/2021

Algorithme d'inversion de la TFCT : OLA

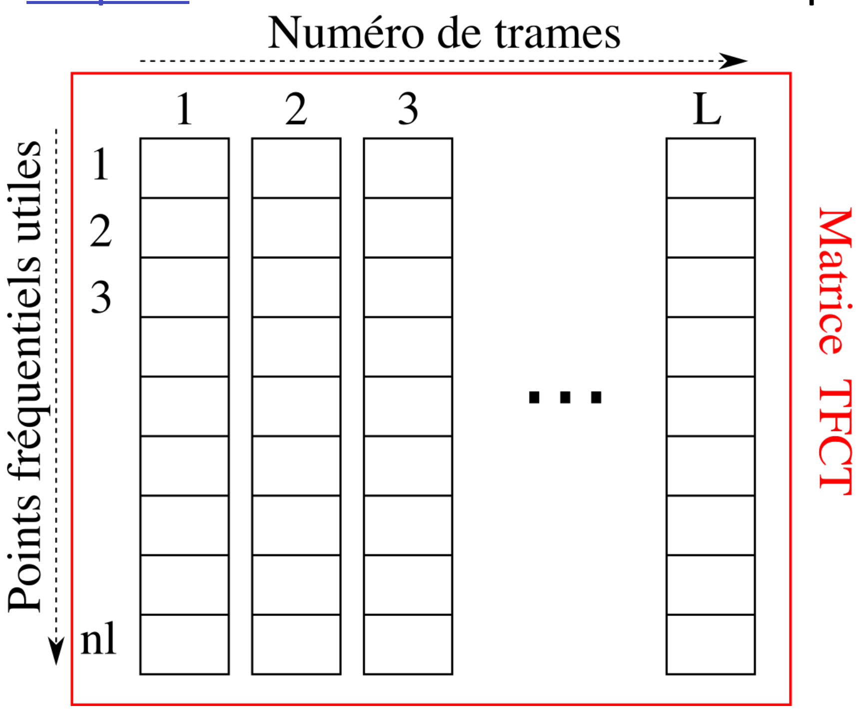
Objectif : reconstruire le signal temporel à partir de la matrice obtenue par TFCT (STFT)

Principe:

1- reconstruire la TFD de chaque trame (domaine fréquentiel)
2- reconstruire chaque trame fenêtrée (domaine temporel)
3- décaler chaque trame dans le temps, avec **chevauchement** entre les trames : ***O****ver****L****ap*
4- sommer les trames décalées : ***A****dd*
5- normaliser le signal temporel

Algorithme d'inversion de la TFCT : OLA

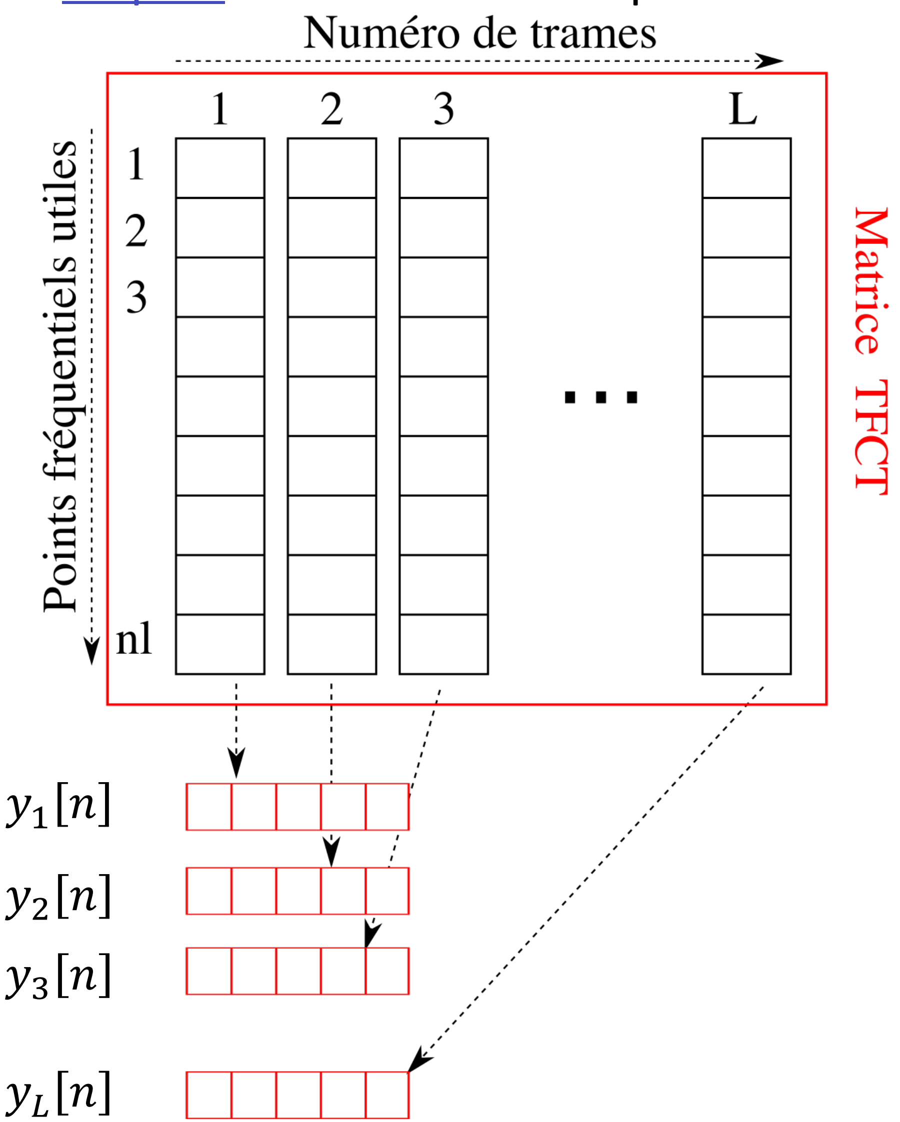
Etape 1: Reconstruire la TFD de chaque trame (domaine fréquentiel)



Où $n_l = floor\left(\frac{M}{2} + 1\right)$

Algorithme d'inversion de la TFCT : OLA

Etape 2: Reconstruire chaque trame fenêtrée (domaine temporel)

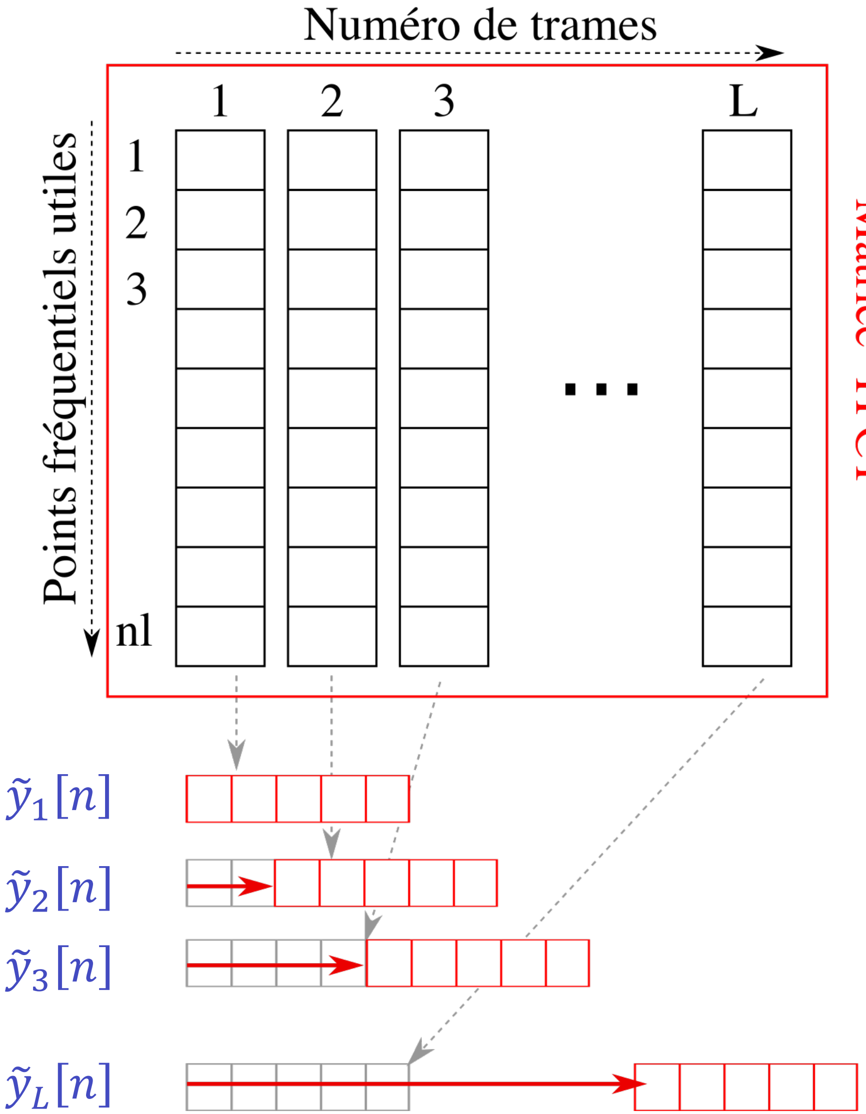


Où

$$y_1[n] = x[n] \times w[n]$$
$$y_2[n] = x\left[n + N_{hop}\right] \times w[n]$$
$$\dots$$
$$y_L[n] = x\left[n + (L-1)N_{hop}\right] \times w[n]$$

Algorithme d'inversion de la TFCT : OLA

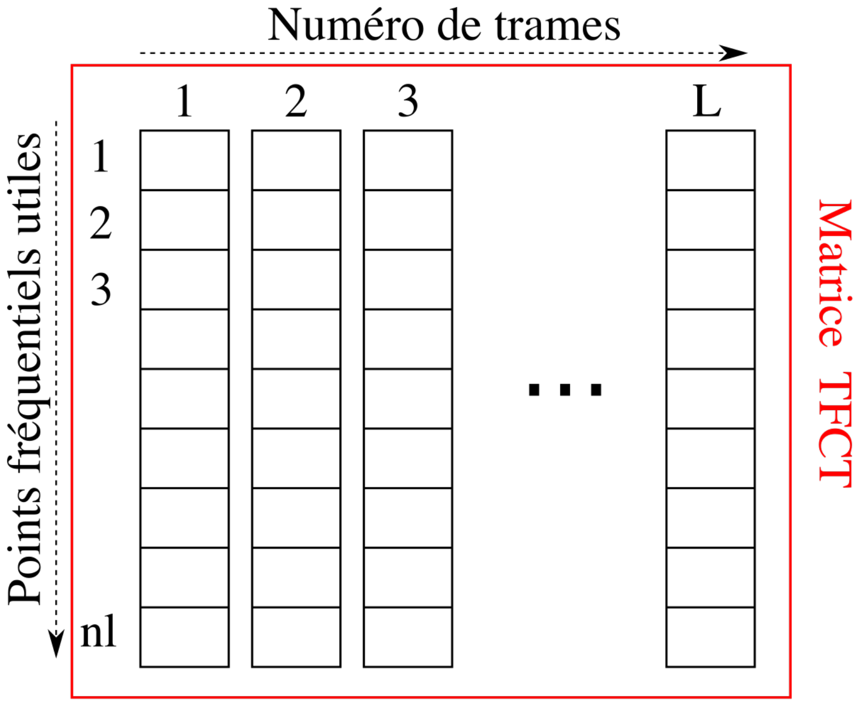
Etape 3: Décaler chaque trame dans le temps, avec **chevauchement** entre les trames : ***OverLap***



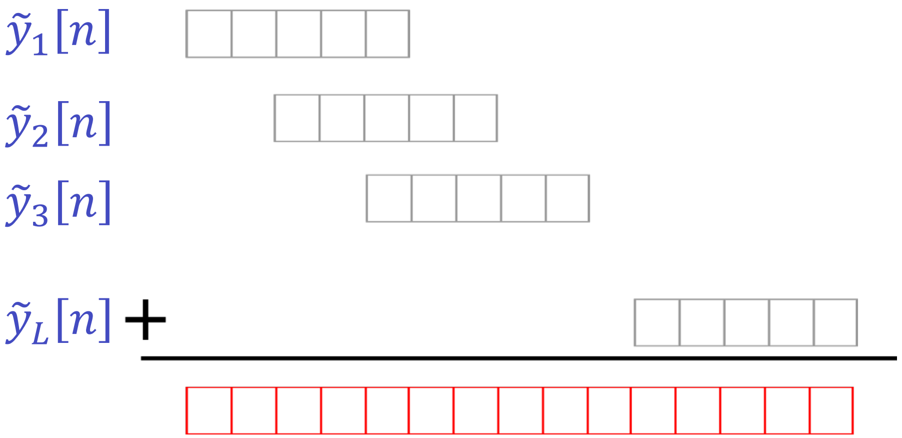
$$\tilde{y}_1[n] = y_1[n] = x[n] \times w[n]$$
$$\tilde{y}_2[n] = y_2\left[n - N_{hop}\right] = x[n] \times w\left[n - N_{hop}\right]$$
…
$$\tilde{y}_L[n] = y_L\left[n - (L-1)N_{hop}\right] = x[n] \times w\left[n - (L-1)N_{hop}\right]$$

Algorithme d’inversion de la TFCT : OLA

Etape 4: Sommer les trames décalées : *Add*

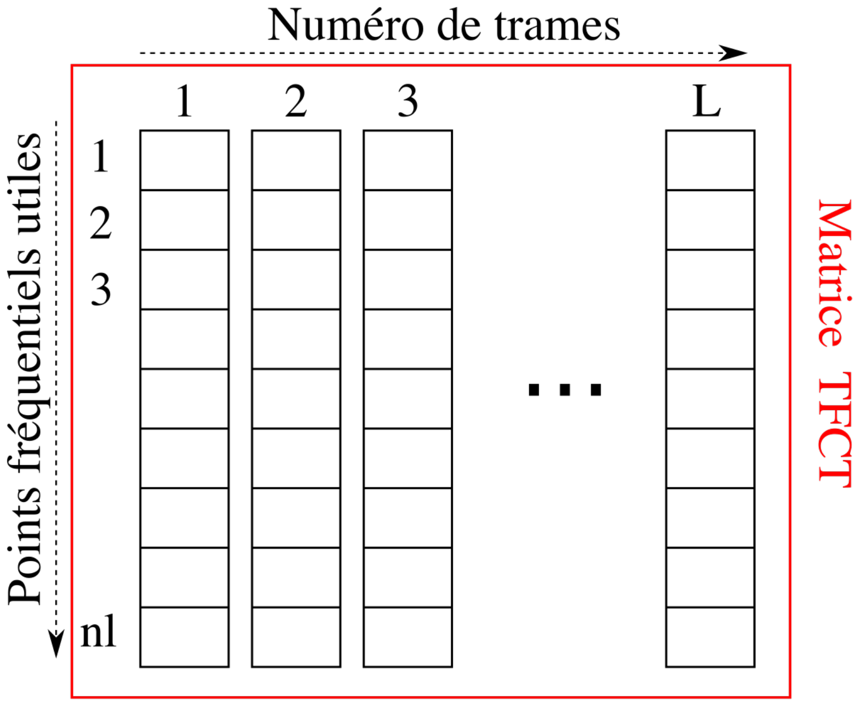


$$
\begin{array}{rl}
 & \tilde{y}_1[n] = y_1[n] = x[n] \times w[n] \\
 & \tilde{y}_2[n] = y_2\left[n - N_{hop}\right] = x[n] \times w\left[n - N_{hop}\right] \\
 & \dots \\
+ & \tilde{y}_L[n] = y_L\left[n - (L-1)N_{hop}\right] = x[n] \times w\left[n - (L-1)N_{hop}\right] \\
\hline
 & y[n] = x[n] \times \sum_{k=0}^{L-1} w\left[n - kN_{hop}\right]
\end{array}
$$

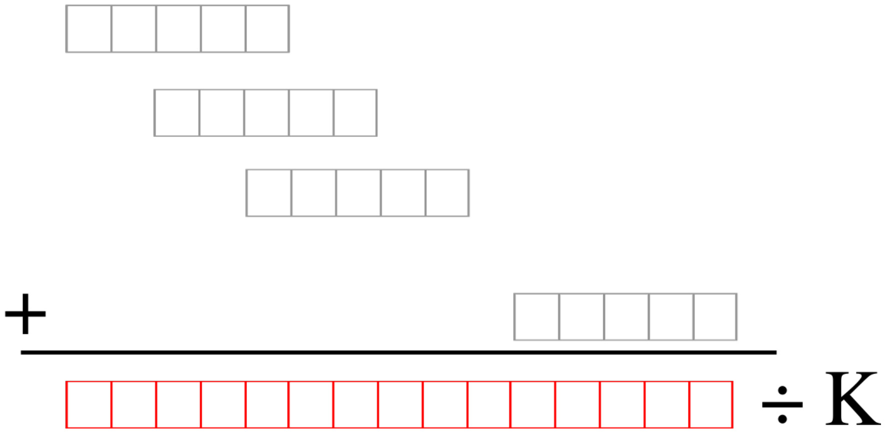


Algorithme d’inversion de la TFCT : OLA

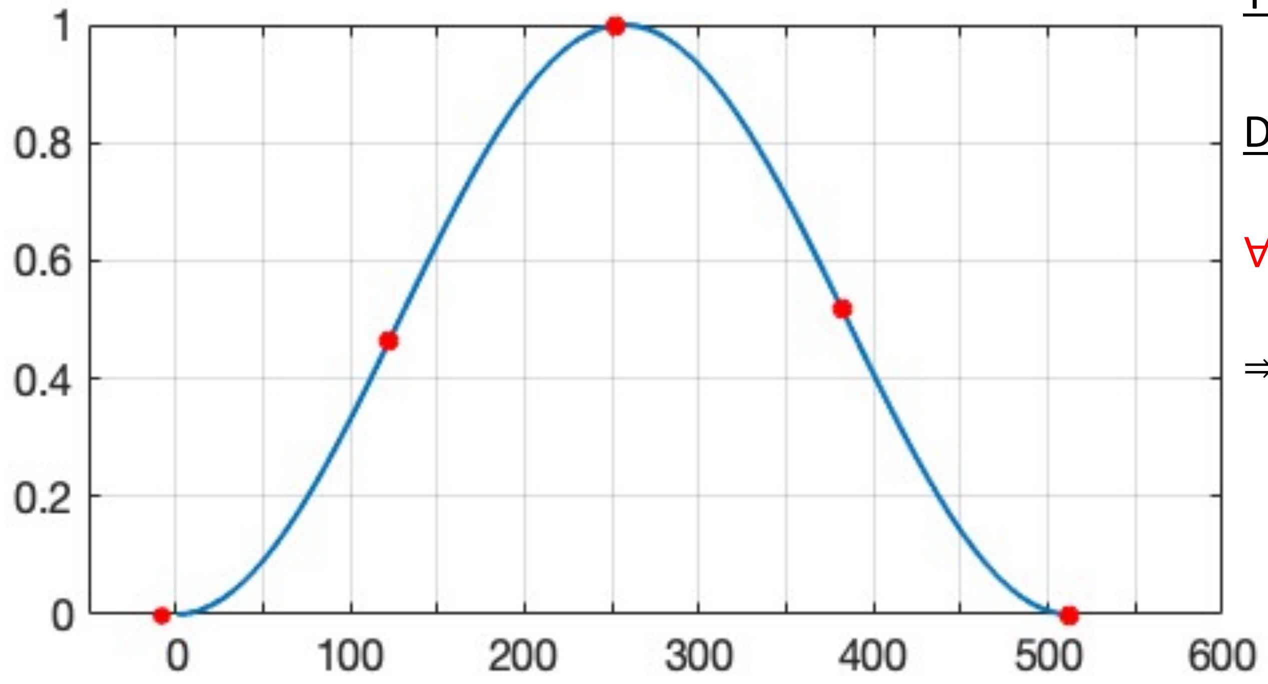
Etape 5: Normaliser le signal temporel



$$
\begin{array}{ll}
 & \tilde{y}_1[n] = y_1[n] = x[n] \times w[n] \\
 & \tilde{y}_2[n] = y_2\left[n - N_{hop}\right] = x[n] \times w\left[n - N_{hop}\right] \\
 & \dots \\
+ & \tilde{y}_L[n] = y_L\left[n - (L-1)N_{hop}\right] = x[n] \times w\left[n - (L-1)N_{hop}\right] \\
\hline
 & y[n] = x[n] \times \underbrace{\sum_{k=0}^{L-1} w\left[n - kN_{hop}\right]}_{= K}
\end{array}
$$



Etape 5: Normaliser le signal temporel



Théorème : si K indépendant de n alors $K = \frac{\sum_{k=0}^{N_{win}-1} w[n]}{N_{hop}}$

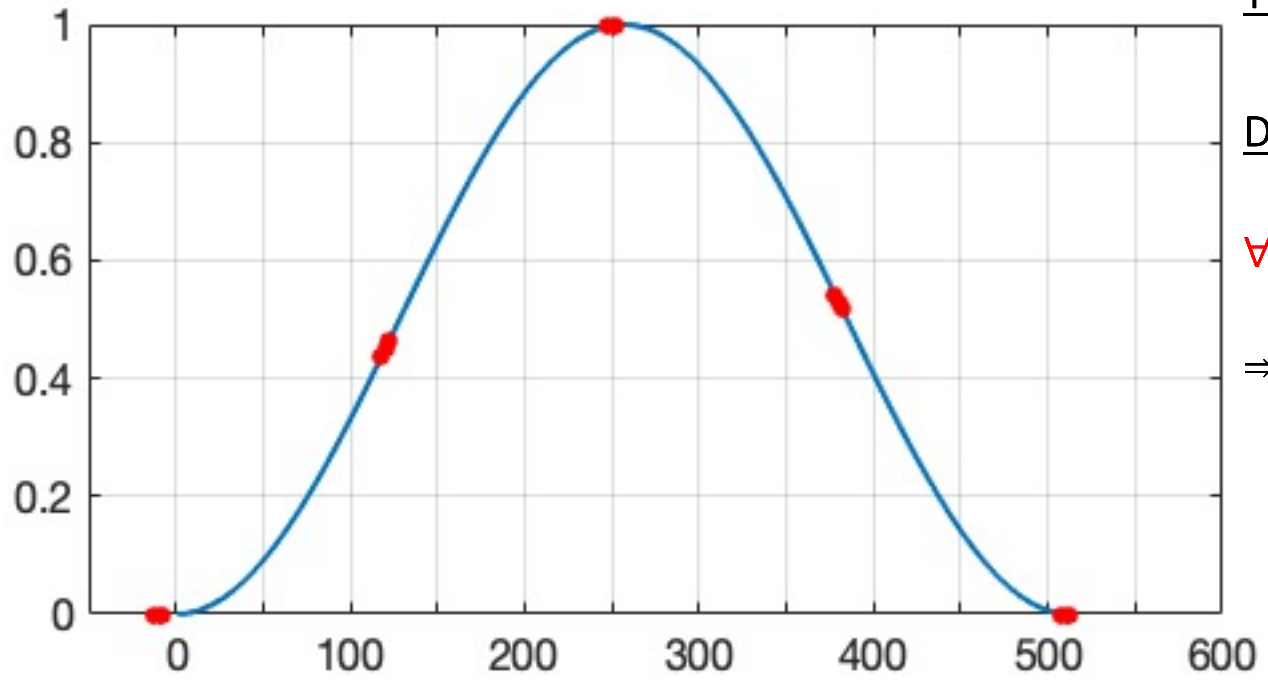
Démonstration :

$$\forall n, K = \sum_{k=0}^{L-1} w[n - kN_{hop}]$$

$$\Rightarrow K = w\left[N_{hop} - 1\right] + w\left[N_{hop} - 1 - kN_{hop}\right] + \cdots + w\left[N_{hop} - 1 - (L-1)N_{hop}\right]$$

Algorithme d'inversion de la TFCT : OLA

Etape 5: Normaliser le signal temporel



Théorème : si K indépendant de n alors $K = \frac{\sum_{k=0}^{N_{win}-1} w[n]}{N_{hop}}$

Démonstration :

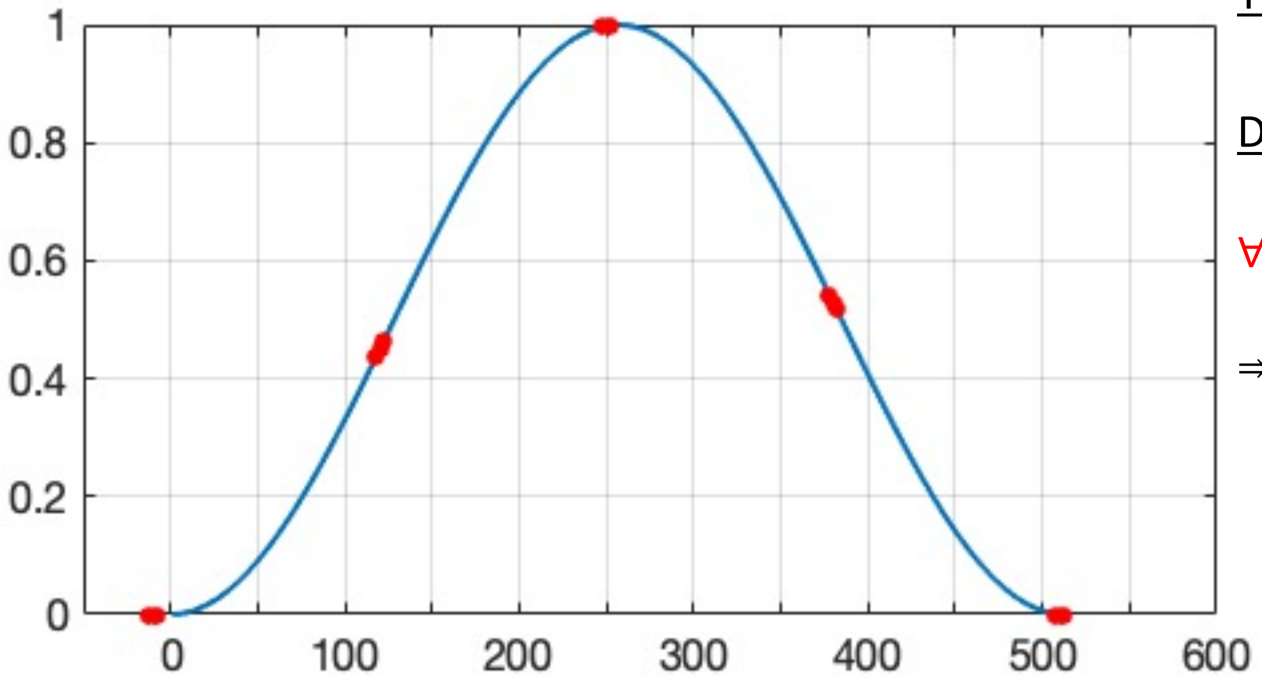
$$\forall n, K = \sum_{k=0}^{L-1} w[n - kN_{hop}]$$

$$\Rightarrow K = w[N_{win} - 1] + w[N_{win} - 1 - N_{hop}] + \cdots + w[N_{win} - 1 - (L-1)N_{hop}]$$

$$K = w[N_{win} - 2] + w[N_{win} - 2 - N_{hop}] + \cdots + w[N_{win} - 2 - (L-1)N_{hop}]$$

Algorithme d'inversion de la TFCT : OLA

Etape 5: Normaliser le signal temporel



Théorème : si K indépendant de n alors $K = \frac{\sum_{k=0}^{N_{win}-1} w[n]}{N_{hop}}$

Démonstration :

$$\forall n, K = \sum_{k=0}^{L-1} w[n - kN_{hop}]$$

$$\Rightarrow K = w[N_{win} - 1] + \boxed{w[N_{win} - 1 - N_{hop}]} + \cdots + w[N_{win} - 1 - (L-1)N_{hop}]$$

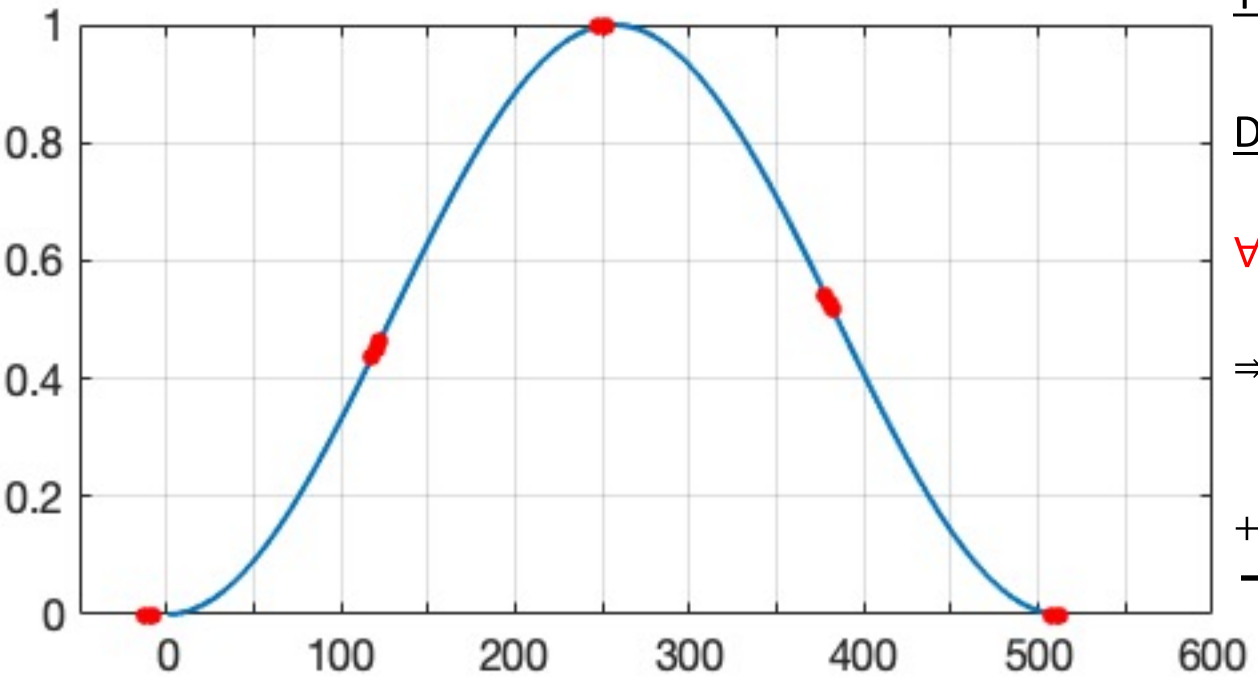
$$K = w[N_{win} - 2] + w[N_{win} - 2 - N_{hop}] + \cdots + w[N_{win} - 2 - (L-1)N_{hop}]$$

…

$$K = \boxed{w[N_{win} - N_{hop}]} + w[N_{win} - 2N_{hop}] + \cdots + w[N_{win} - LN_{hop}]$$

Algorithme d'inversion de la TFCT : OLA

Etape 5: Normaliser le signal temporel



Théorème : si K indépendant de n alors $K = \frac{\sum_{k=0}^{N_{win}-1} w[n]}{N_{hop}}$

Démonstration :

$$\forall n, K = \sum_{k=0}^{L-1} w[n - kN_{hop}]$$

$$\Rightarrow K = w[N_{win} - 1] + \boxed{w[N_{win} - 1 - N_{hop}]} + \cdots + w[N_{win} - 1 - (L-1)N_{hop}]$$

$$K = w[N_{win} - 2] + w[N_{win} - 2 - N_{hop}] + \cdots + w[N_{win} - 2 - (L-1)N_{hop}]$$

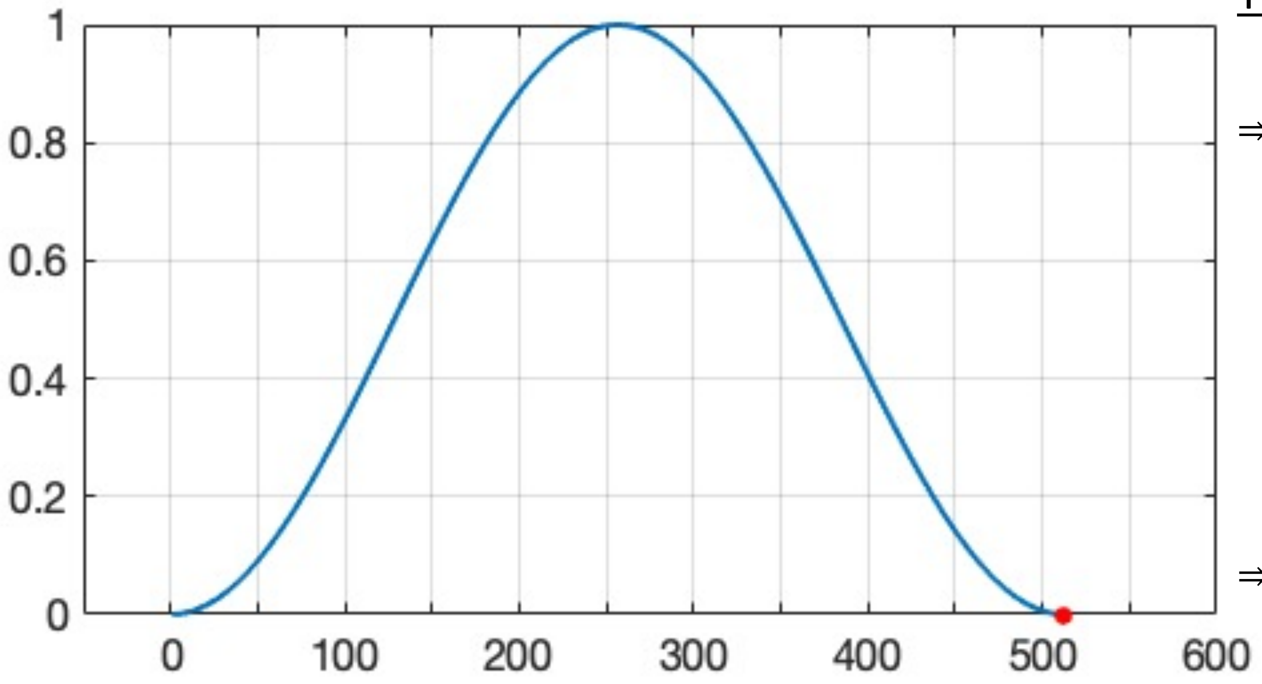
$$\ldots$$

$$+ K = \boxed{w[N_{win} - N_{hop}]} + w[N_{win} - 2N_{hop}] + \cdots + w[N_{win} - LN_{hop}]$$

$$N_{hop}\, K = \sum_{n=0}^{N_{win}-1} w[n]$$

Algorithme d'inversion de la TFCT : OLA

Cas particulier : $N_{hop} > \frac{N_{win}}{2}$



Théorème : si K indépendant de n alors $K = \frac{\sum_{k=0}^{N_{win}-1} w[n]}{N_{hop}}$

$\Rightarrow K = w[N_{win} - 1] + w[N_{win} - 1 - N_{hop}]$

$K = w[N_{win} - 2] + w[N_{win} - 2 - N_{hop}]$

...

$K = w[N_{hop} + 1] + w[1]$

$K = w[N_{hop}]$

$K = w[N_{hop} - 1]$

...

$K = w[0]$

$\Rightarrow$ Impossible