



RAPPORT DE PROJET INDUSTRIEL 2014

DELANNOY Anthony
BIACHE Quentin
COMBANIÈRE Jérôme
LI Xueyang

4 juin 2014

Table des matières

I	Introduction	3
II	L'organisation du projet	4
III	La conception de l'interface graphique	5
1	Cahier des charges	5
2	Présentation	5
IV	Création de signaux de télécommunication	7
V	Implémentation des méthodes de reconstruction	8
3	Interpolateur de Shannon	8
4	Formule de Lagrange modifiée	9
5	Reconstruction matricielle	12
6	Reconstruction par splines cubiques	13
7	Reconstruction itérative	13
VI	Conclusion	15

Première partie

Introduction

L'objectif de ce projet industriel a consisté à étudier la théorie de l'échantillonnage non-uniforme et à voir quelles sont les conséquences sur la reconstruction des signaux échantillonnés de cette façon.

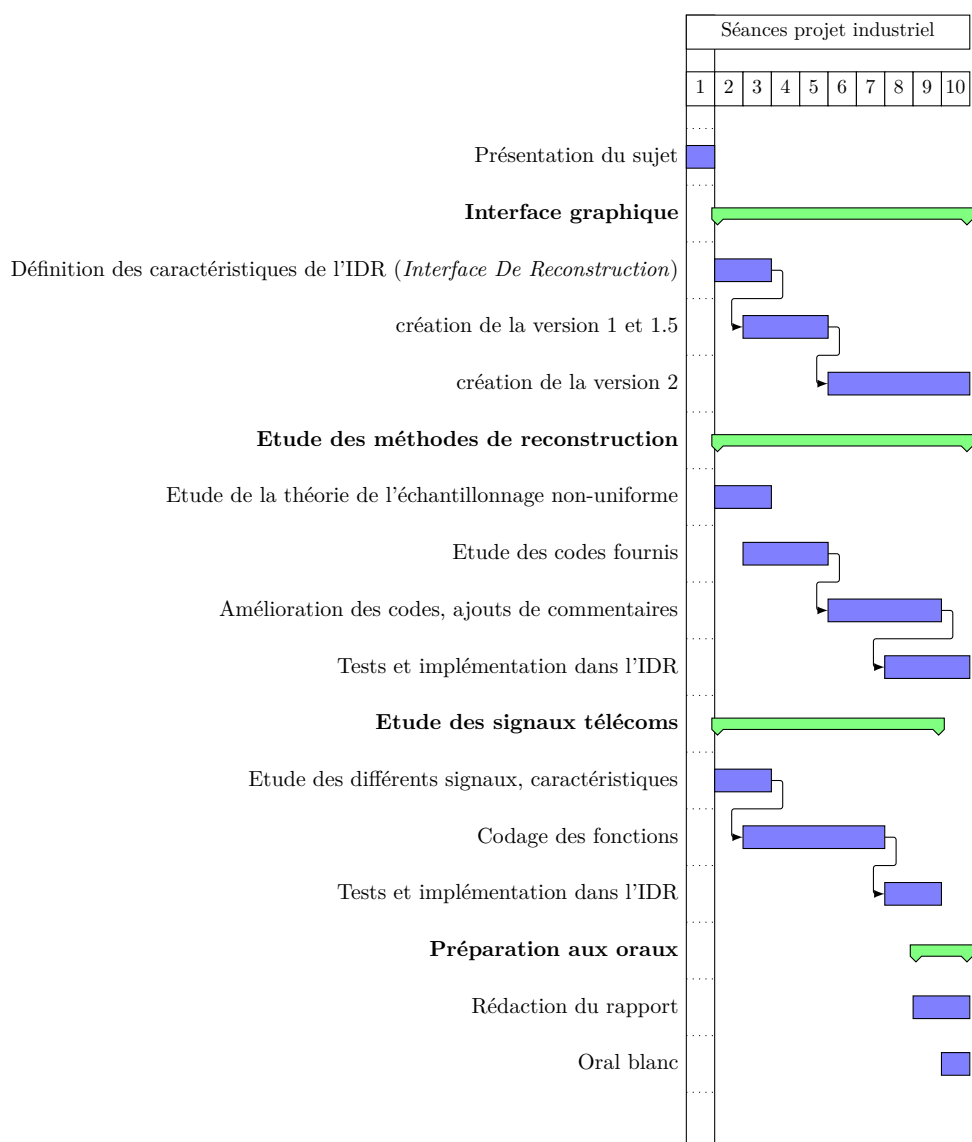
L'échantillonnage non-uniforme correspond à une réalité physique des systèmes d'acquisition. Il peut être délibéré (pour échantillonner là où se situe toute l'information, et éviter de stocker beaucoup de points sur des zones uniformes), mais peut être aussi subi dans le cas des défauts des échantillonneurs.

Les conséquences de cette non-uniformité peuvent être variables. C'est dans ce but que l'interface graphique présentée ici a été développée. L'objectif est de réunir dans une seule application une grande partie des méthodes de reconstruction, sans que l'utilisateur n'ait à se demander comment utiliser les fonctions, avec quels arguments etc...

Deuxième partie

L'organisation du projet

Ci-dessous, la répartition des tâches dans le temps pour ce projet industriel.



Troisième partie

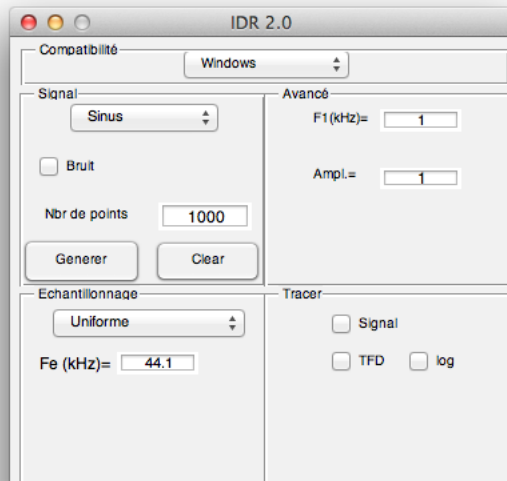
La conception de l'interface graphique

1 Cahier des charges

L'interface graphique doit être claire et intuitive tout en permettant beaucoup de fonctionnalités :

- Création de divers signaux télécoms.
- Tracé de la TFD.
- Choix d'échantillonner uniformément ou non.
- Multiples méthodes de reconstruction.
- Tracé d'erreur quadratique de reconstruction.
- Placement automatique des plots.
- Dynamisme de l'interface.

2 Présentation



La partie haute gauche permet le choix du signal, l'ajout de bruit avec SNR modifiable et le choix du nombre de points. En haut à droite il y a toutes les options relatives au signal lui-même (fréquence, amplitude, nombre de symboles, ...), ce bloc s'adapte dynamiquement selon le choix du signal. En bas à gauche il y a le bloc avec les options de l'échantillonnage : la fréquence, uniforme ou non, si l'on choisit non-uniforme on peut régler le pourcentage du jitter. En bas à droite il s'agit du bloc permettant tout les tracés, le placement des fenêtres est automatique (si l'option d'échantillonnage est *non-uniforme* alors le choix et le tracé de la reconstruction est disponible en plus de l'erreur quadratique de reconstruction).

Une fois tous les choix pour le signal effectués il faut appuyer sur le bouton *Generer* afin que l'interface prenne en compte les changements. De plus pour fermer toutes les fenêtres et repartir de 0 vous pouvez appuyer sur le bouton *Clear* qui fera cela pour vous.

Remarque :

- Le nombre de points et la fréquence d'échantillonnage étant fixés, c'est le nombre de période tracé qui varie.
- Nous avons rajouter la compatibilité du placement de fenêtre pour windows et Mac. Afin d'en profiter il suffit de choisir son système dans le panneau situé tout en haut.

Quatrième partie

Création de signaux de télécommunication

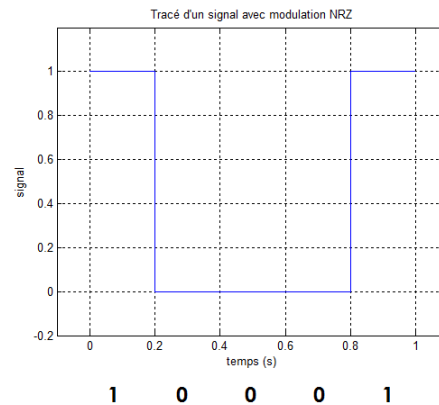
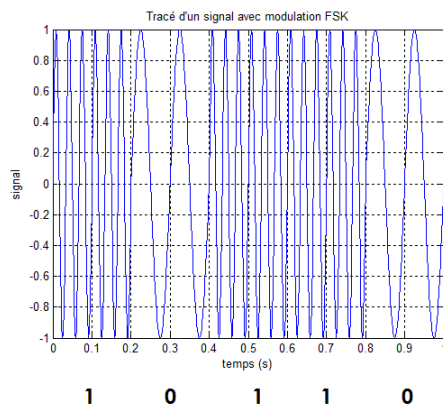
Afin d'étudier les capacités réelles des méthodes de reconstruction dans le cas non-uniforme, il est judicieux de ne pas les appliquer uniquement sur des signaux théoriques, mais aussi sur des signaux utilisés en réalité.

Le détail de la conception de chaque fonction figure dans les commentaires des codes Matlab. Chaque type de signal a été placé dans une fonction séparée. Les signaux ont été conçus en prenant soin à chaque fois d'adapter les arguments afin que les fonctions soient aisément exploitables par l'application globale.

On a veillé à utiliser au maximum des fonctions telles que le produit tensoriel et le produit matriciel afin d'optimiser les temps de calcul.

Au total, 8 signaux ont été créés sous Matlab, avec notamment les signaux modulés en ASK (*Amplitude Shift Keying*, codage par modulation d'amplitude), FSK (*Frequency Shift Keying*, codage par modulation de fréquence), 16QAM (*Quadrature Amplitude Modulation* modulation d'amplitude en quadrature) etc.

Exemples de signaux élaborés :



Cinquième partie

Implémentation des méthodes de reconstruction

3 Interpolateur de Shannon

La première méthode de reconstruction qui a été implémentée est la méthode de reconstruction de Shannon. Celle-ci consiste en un prolongement de l'expression générale de reconstruction d'un signal échantillonné de manière idéale et uniforme :

$$x_r(t) = \sum_{k=1}^{N_{ech}} x(t_k) \text{sinc}(\pi f_e(t - t_k))$$

Les $\{t_k | k \in \llbracket 1, N_{ech} \rrbracket\}$ correspondant aux instants d'échantillonnage non uniformes.

Il faut remarquer cependant que cette méthode reconstruit très mal le signal lorsque l'échantillonnage n'est plus uniforme.

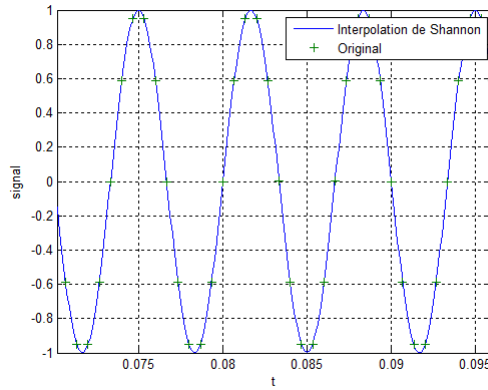


FIGURE 1 – Reconstruction avec échantillonnage uniforme

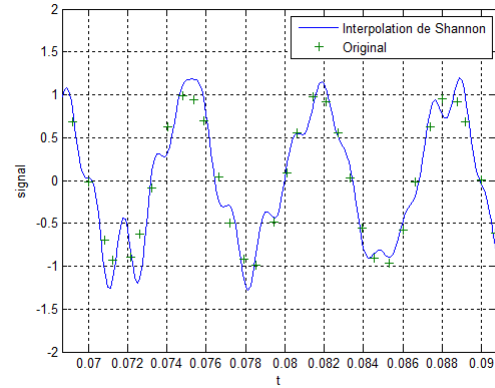


FIGURE 2 – Reconstruction avec échantillonnage non uniforme (jitter = 50% de $\frac{T_{ech}}{2}$)

La méthode perd nettement de son efficacité dans le cas d'un échantillonnage non uniforme. Cela se comprend du fait que le sinus cardinal

$$\text{sinc}(\pi f_e(t - t_k)) \quad (1)$$

pour k donné, n'a plus aucune raison de s'annuler aux points $t \in \{t_j | j \in \llbracket 1, N_{ech} \rrbracket, j \neq k\}$ lorsque les t_j sont quelconques, ce qui n'était pas le cas pour un échantillonnage idéal avec $t_k = kT_{ech}$

La reconstruction doit donc passer nécessairement par d'autres méthodes.

4 Formule de Lagrange modifiée

Cette méthode se base sur les polynômes interpolateurs de Lagrange. La théorie nous fournit alors les formules de reconstruction suivantes :

$$x(t) = \lim_{M \rightarrow \infty} H_M(t) \left[\sum_{0 \leq |k| \leq M} a_M(t, t_k) x(t_k) \right] \quad (2)$$

$$H_M(t) = \frac{F_M(t)}{G_M(t)} \sin(\pi t) \quad (3)$$

$$F_M(t) = \prod_{0 \leq |k| \leq M} \left(1 - \frac{t}{t_k} \right) \quad (4)$$

$$G_M(t) = \pi t \prod_{1 \leq |k| \leq M-1} \left(1 - \frac{t}{k} \right) \quad (5)$$

$$a_M(t, t_k) = \frac{G_M(t_k)}{(t - t_k) F'_M(t_k) \sin(\pi t_k)} \quad (6)$$

Cette forme semble relativement peu intuitive et difficile à comprendre au premier abord. On s'est donc proposé de simplifier cette expression avant de l'implémenter dans Matlab. On peut déjà remarquer que :

$$F'_M(t) = \sum_{0 \leq |k| \leq M} \left[-\frac{1}{t_k} \prod_{\substack{0 \leq |j| \leq M \\ j \neq k}} \left(1 - \frac{t}{t_j} \right) \right] \quad (7)$$

d'où :

$$F'_M(t_k) = -\frac{1}{t_k} \prod_{\substack{0 \leq |k| \leq M \\ j \neq k}} \left(1 - \frac{t_k}{t_j}\right) \quad (8)$$

En substituant dans l'expression des a_M il vient :

$$a_M(t, t_k) = \frac{\pi t_k}{\sin(\pi t_k)} \frac{\prod_{1 \leq |j| \leq M-1} \left(1 - \frac{t_k}{t_j}\right)}{(t - t_k) \left(-\frac{1}{t_k}\right) \prod_{\substack{0 \leq |j| \leq M \\ j \neq k}} \left(1 - \frac{t_k}{t_j}\right)} \quad (9)$$

$$= \frac{\pi t_k}{\sin(\pi t_k)} \frac{\prod_{1 \leq |j| \leq M-1} \left(1 - \frac{t_k}{t_j}\right)}{\left(1 - \frac{t}{t_k}\right) \prod_{\substack{0 \leq |j| \leq M \\ j \neq k}} \left(1 - \frac{t_k}{t_j}\right)} \quad (10)$$

Ainsi, en développant l'expression de $x(t)$:

$$x(t) = \lim_{M \rightarrow \infty} \sum_{0 \leq |k| \leq M} x(t_k) \frac{\text{sinc}(\pi t)}{\text{sinc}(\pi t_k)} \left[\frac{\prod_{0 \leq |j| \leq M} \left(1 - \frac{t}{t_j}\right)}{\prod_{1 \leq |j| \leq M-1} \left(1 - \frac{t}{t_j}\right)} \right] \left[\frac{\prod_{1 \leq |j| \leq M-1} \left(1 - \frac{t_k}{t_j}\right)}{\left(1 - \frac{t}{t_k}\right) \prod_{\substack{0 \leq |j| \leq M \\ j \neq k}} \left(1 - \frac{t_k}{t_j}\right)} \right] \quad (11)$$

$$= \lim_{M \rightarrow \infty} \sum_{0 \leq |k| \leq M} x(t_k) \frac{\text{sinc}(\pi t)}{\text{sinc}(\pi t_k)} \left[\frac{\prod_{\substack{0 \leq |j| \leq M \\ j \neq k}} \left(1 - \frac{t}{t_j}\right)}{\prod_{1 \leq |j| \leq M-1} \left(1 - \frac{t}{t_j}\right)} \right] \left[\frac{\prod_{1 \leq |j| \leq M-1} \left(1 - \frac{t_k}{t_j}\right)}{\prod_{\substack{0 \leq |j| \leq M \\ j \neq k}} \left(1 - \frac{t_k}{t_j}\right)} \right] \quad (12)$$

$$= \lim_{M \rightarrow \infty} \sum_{0 \leq |k| \leq M} x(t_k) \frac{f_k(t)}{f_k(t_k)} \quad (13)$$

Avec :

$$f_k(t) = \text{sinc}(\pi t) \frac{\prod_{\substack{0 \leq |j| \leq M \\ j \neq k}} \left(1 - \frac{t}{t_j}\right)}{\prod_{1 \leq |j| \leq M-1} \left(1 - \frac{t}{t_j}\right)} \quad (14)$$

Il devient alors évident que le terme $\frac{f_k(t)}{f_k(t_k)}$ vaut 1 sur l'instant t_k , et est nul sur tous les autres instants échantillonnés non-uniformément.

Il reste à présent à lever les indéterminations sur l'expression de f_k . La fonction se compose en effet d'une fraction rationnelle dont les pôles sont tous les entiers de l'intervalle $\llbracket -(M-1), M-1 \rrbracket \setminus \{0\}$.

Sachant que :

$$\frac{\text{sinc}(\pi t)}{1 - \frac{t}{j}} \xrightarrow{t \rightarrow j} (-1)^{j+1} \quad (15)$$

On a finalement :

$$f_k(t) = \begin{cases} (-1)^{l+1} \frac{\prod_{\substack{0 \leq |j| \leq M \\ j \neq k}} \left(1 - \frac{t}{t_j}\right)}{\prod_{\substack{1 \leq |j| \leq M-1 \\ j \neq l}} \left(1 - \frac{t}{t_j}\right)}, & \text{pour } t = l \in \llbracket -(M-1), M-1 \rrbracket \setminus \{0\} \\ \text{sinc}(\pi t) \frac{\prod_{\substack{0 \leq |j| \leq M \\ j \neq k}} \left(1 - \frac{t}{t_j}\right)}{\prod_{1 \leq |j| \leq M-1} \left(1 - \frac{t}{t_j}\right)}, & \text{sinon} \end{cases} \quad (16)$$

C'est cette expression qui a été implanté finalement dans Matlab. Le fait d'avoir développé l'expression a permis d'éliminer les discontinuités (lorsque la reconstruction est évaluée sur les instants non-uniformes) qui avaient été décelées dans le projet long.

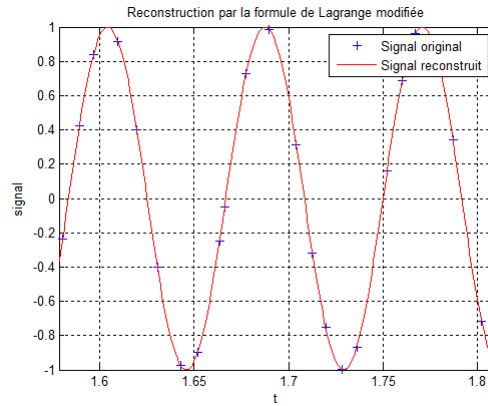


FIGURE 3 – Reconstruction du signal (jitter = 90% $\frac{T_{ech}}{2}$, fenêtre de 20 points)

5 Reconstruction matricielle

On a pu voir dans les parties précédentes que la reconstruction via l'interpolateur de Shannon ne pouvait plus s'appliquer lorsque l'échantillonnage n'est pas uniforme.

Le problème peut cependant être contourné si on évalue le signal reconstruit sur des instants uniformes. Si on note $\{t_k^* | k \in \llbracket 1, N_{eval} \rrbracket\}$ les instants uniformes sur lesquels on souhaite évaluer le signal, on peut écrire que :

$$x(t_j) = \sum_{k=1}^{N_{eval}} x_r(t_k^*) \text{sinc}(\pi f_e(t_j - t_k^*))$$

Tout se passe comme si les points échantillonnés non-uniformément étaient issus de la reconstruction du signal sur les instants uniformes ; Sauf qu'ici, ce qu'on cherche à obtenir ce sont les $x_r(t_k^*)$.

Le problème se ramène donc à un problème d'inversion matricielle. Le code Matlab élaboré lors du projet long a été repris mais sans modification notable ; Seule une fonction supplémentaire `window_find.m` a été ajoutée, afin d'améliorer la compréhension de la reconstruction sur une fenêtre.

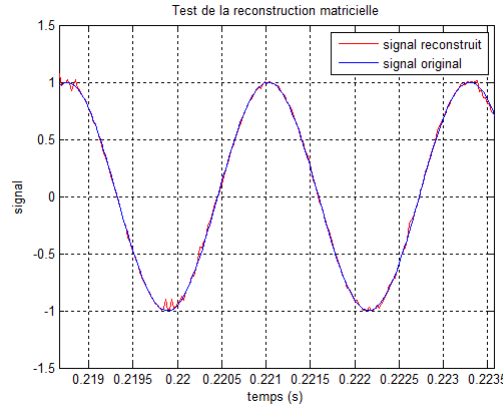


FIGURE 4 – Reconstruction du signal (jitter = $90\% \frac{T_{ech}}{2}$, fenêtre de 20 points)

On peut noter que globalement, l'allure du signal est retrouvée, même si quelques parasites viennent s'ajouter au signal.

6 Reconstruction par splines cubiques

Le principe de la reconstruction par splines cubiques se base sur les polynômes du même nom.

L'idée est de trouver un polynôme d'ordre 3 localement qui passe par deux échantillons consécutifs. On procède de même pour tous les autres points, en prenant soin d'assurer la continuité de la fonction ainsi obtenue, ainsi que la continuité de sa dérivée et de sa dérivée seconde.

La recherche de la solution passe par la résolution d'un système matriciel. Toutefois, les bibliothèques Matlab intégrant déjà cet outil, il n'a pas été nécessaire de recoder la fonction et on a pu se contenter d'utiliser la fonction `interp1`.

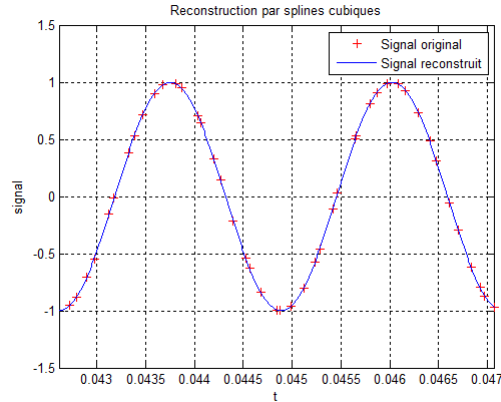
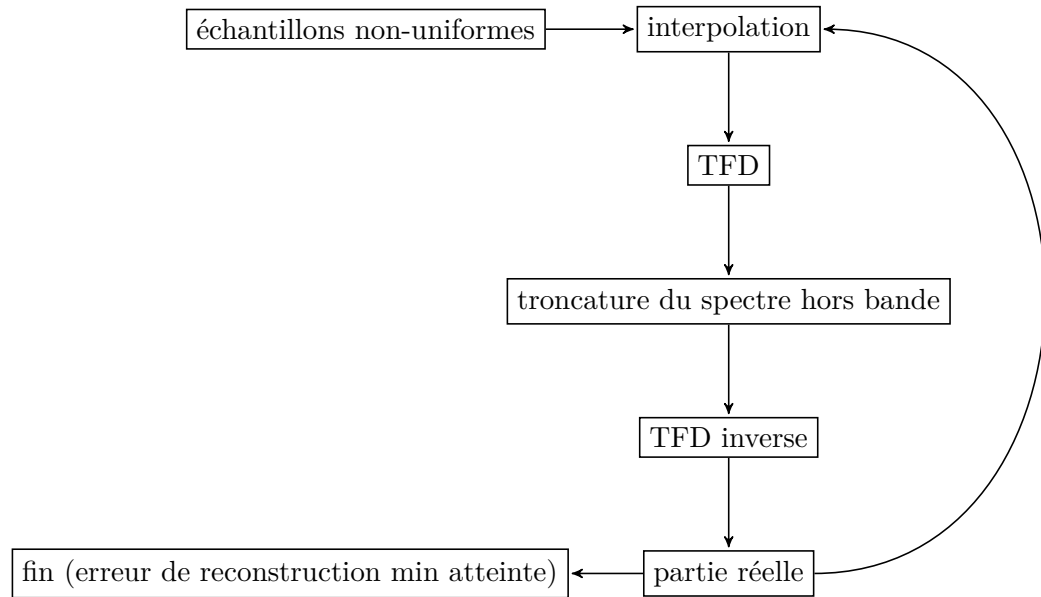


FIGURE 5 – Reconstruction du signal (jitter = $90\% \frac{T_{ech}}{2}$, fenêtre de 20 points)

7 Reconstruction itérative

Cette méthode diffère un peu des autres dans le sens où elle repose sur une méthode itérative. On exploite notamment que le signal a une bande spectrale finie et connue. La reconstruction peut alors se faire en itérant la

suite d'opérations suivante :



A chaque itération, on passe dans le domaine spectral pour éliminer tout le spectre hors-bande, d'où la nécessité de connaître les caractéristiques fréquentielles du signal à reconstruire.

Sixième partie

Conclusion

Ce projet industriel aura été l'occasion de s'intéresser aux problèmes liés à la réalité physique des systèmes d'acquisition, et aux solutions qui ont été trouvées jusque là.

L'interface graphique élaborée était conforme à l'objectif attendu, à savoir réunir en une seule application la génération de signaux de télécommunication, le tracé de la TFD, et enfin l'étude des méthodes de reconstruction dans le cas uniforme et non-uniforme.