

# Transmission d'un message numérique caché dans un signal audio

Joël Liénard

LIS/ENSIEG/INPG

Domaine Universitaire BP 46 - 38400 St Martin d'Hères

Joel.lienard@inpg.fr

**Résumé** – On décrit la réalisation d'un dispositif de transmission d'une information numérique à faible débit superposée à un signal audio. On expose les choix algorithmiques, basés sur une technique d'étalement de spectre mis en œuvre avec un procédé numérique original de correction des dérives d'horloge, particulièrement économique en calcul. On présente le processus de développement à l'aide du logiciel Mustig utilisé pour les expérimentations en simulation et en fonctionnement temps réel, puis en générateur de code C pour le portage rapide sur un processeur de signal.

**Abstract** – *One describes the realization of a numerical information transmission device at low bit rate superimposed on an audio signal. One exposes the algorithmic choices, based on a technique of spread spectrum implemented with an original numerical process of correction of the clock drifts, particularly economic in calculation. One presents the process of development using the Mustig software for the experiments in simulation and real time operation, and then the generation of C code for a fast implementation on a DSP.*

## 1. Introduction

L'objectif est de transmettre un message numérique en le superposant à un signal audio (parole ou musique). Cette transmission doit être faite le plus discrètement possible, c'est à dire que le message ne doit pratiquement pas être discerné par l'oreille. En d'autres termes, il ne doit pas gêner l'écoute du message sonore.

En raison de cette discrétion et de la faible bande passante disponible, il est évident que le débit binaire de la transmission numérique ne peut être que très faible. On ne peut guère envisager plus de 100 bits/seconde.

Le signal numérique pourra être de type non permanent, dans des applications de type télécommande, ou permanent, dans des applications de transmission des paramètres d'une animation vidéo associée au son. Notons que le problème posé ressemble à celui des applications de type tatouage ou "watermarking", mais qu'il ajoute une exigence de débit et de rapidité de traitement.

Notons également que la transmission du signal audio encodé est supposée réalisée en technique analogique. Il faut donc prendre en compte une dérive relative des horloges entre le codeur et le décodeur.

Une difficulté importante de l'étude de ce système est la définition d'un critère de gêne acceptable pour le signal ajouté. Cette gêne est subjective. Nous n'avons pas trouvé d'autre solution que des essais sur de nombreuses séquences et par plusieurs utilisateurs.

## 2. Choix de la méthode de codage

Compte tenu de la ressemblance du problème avec le tatouage, nous avons tout d'abord envisagé les techniques

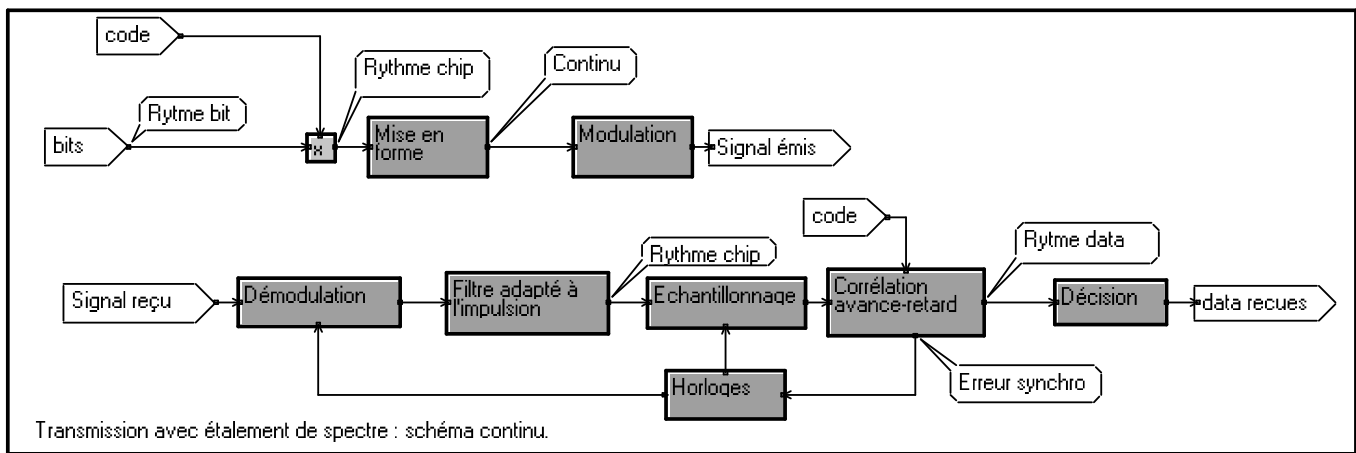
propres à ce domaine [1]. On peut les séparer en deux classes: les méthodes additives et les méthodes convolutives. Ces dernières nous ont paru intéressantes, car a priori très discrètes : une convolution revient à un filtrage, qui ne fait que modifier légèrement le timbre ou la réverbération. Pourtant, après quelques essais, il nous a semblé impossible d'obtenir un débit acceptable avec cette technique.

Les méthodes additives sont basées sur l'ajout d'un signal codé qui est reconnu dans le décodeur par une corrélation. Cela correspond tout à fait à ce qui est fait, dans la domaine des communications numériques, pour les modulations de type étalement de spectre par codage et ces techniques sont connues également pour leur qualité de discrétion.

Notre problème est donc semblable à celui d'une communication numérique [2]. Une différence importante est que, bien que le signal audio puisse être considéré comme un bruit perturbateur pour la transmission d'information numérique, ce bruit est connu du codeur. Il est donc possible d'en tenir compte à l'émission pour, par exemple, asservir la puissance du signal numérique à celle du signal audio. Nous n'avons cependant pas envisagé de méthodes très sophistiquées car le cadre de notre étude imposait une simplification maximum du décodeur. Nous avons en particulier évité délibérément tout calcul de transformée de Fourier et donc tout passage en fréquence.

## 3. Choix de la bande utile

La bande passante utilisable s'étend de 300 à 6000 Hz. Nous avons fait nos premiers essais avec un signal codé dans la bande de 300 à 3000 Hz. Elle avait l'avantage d'utiliser une fréquence d'échantillonnage faible.



Transmission avec étalement de spectre : schéma continu.

Il est vite apparu préférable d'utiliser la bande de 3000 à 6000 Hz. En effet, cette bande correspond au début de la chute de sensibilité de l'oreille : le signal codé est donc moins audible. En revanche, cela est plus coûteux en puissance de calcul.

## 4. Technique de démodulation

### 4.1 Principe

Compte tenu de la faible fréquence porteuse, il est possible de réaliser le traitement entièrement en numérique. Cela permet, tant pour le codeur que pour le décodeur, l'intégration de tout le traitement dans un seul DSP de faible puissance.

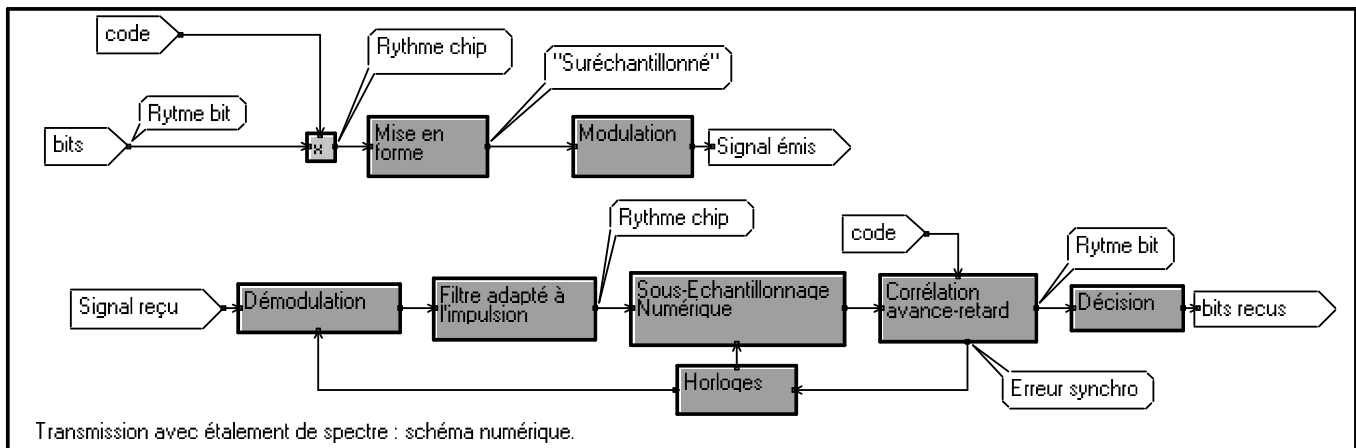
La figure résume une chaîne typique de transmission par étalement de spectre. Chaque bit du message numérique module une séquence binaire dite séquence d'étalement. On appelle "chip" les bits de cette séquence. Soit  $N$  le nombre de bits de cette séquence. Ces chips sont transmis avec un débit  $N$  fois plus grand que les données par une méthode classique de transmission numérique : une impulsion de mise en forme transforme la suite binaire en un signal continu d'étendue spectrale limitée qui est ensuite translaté en fréquence par modulation. En réception, une démodulation ramenant le signal en bande de base est suivie d'un filtrage adapté à l'impulsion de mise en forme et de l'échantillonnage au rythme chip. Les données sont retrouvées par corrélation avec la séquence d'étalement et seuillage. Il est nécessaire d'ajouter en réception un asservissement de fréquence porteuse et de

rythme symbole. La technique la plus classique est la corrélation avance retard, qui consiste à calculer en permanence trois points de corrélation : un à la position supposée du pic pour obtenir les données et un sur chaque flanc de ce pic. La différence des amplitudes des deux derniers détermine le sens de l'erreur de phase.

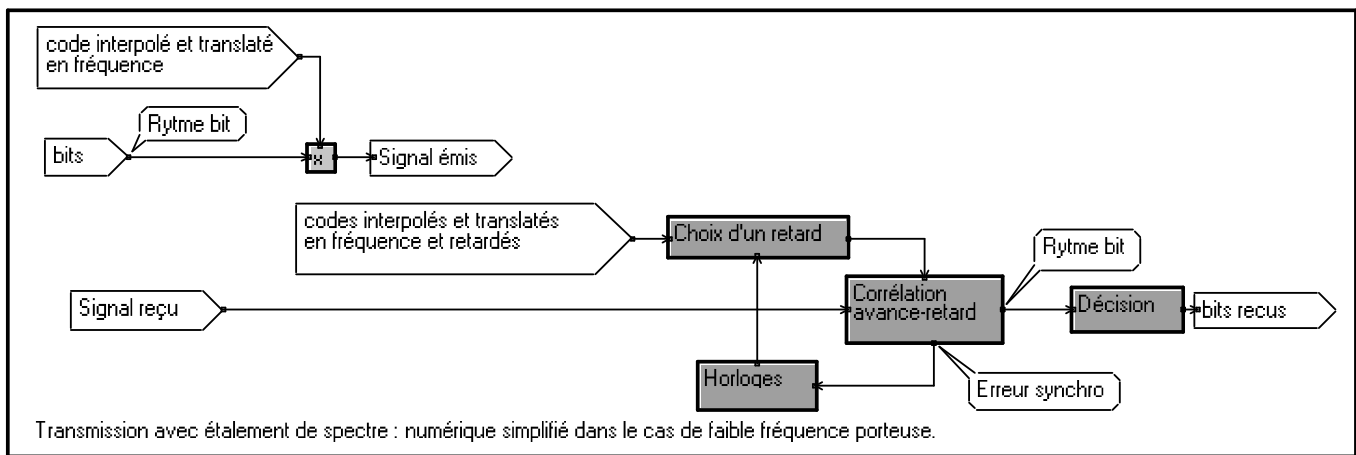
En version "tout numérique", on remplace le signal continu par un signal numérique échantillonné à une fréquence multiple, dans un rapport  $K$ , de la fréquence des chips. La difficulté se trouve dans la fonction numérique équivalente à celle de l'échantillonneur. Si le rapport  $K$  est très grand, un simple sous échantillonnage est suffisant. Sinon, il est nécessaire d'y inclure un interpolateur pour calculer les valeurs à des instants où l'on ne dispose pas d'échantillon. Pour de faibles valeurs de  $K$ , l'interpolation pourra nécessiter plus d'une dizaine de coefficients. On peut modéliser cet interpolateur comme un filtre multi-cadence à coefficients variables, appelé aussi filtre polyphase[3].

La chaîne de réception comporte donc normalement trois filtres en cascade : le filtre passe-bas de démodulation, le filtre adapté à l'impulsion de mise en forme et le filtre interpolateur. Ces filtres sont suivis de la fonction de corrélation au code. On peut également formaliser cette opération par un filtre multi-cadence : le filtre adapté au code suivi d'un sous-échantillonnage d'un facteur égal à la longueur du code

L'originalité de l'algorithme proposé est le remplacement de tous ces filtres par un seul filtre multi-cadence.



Transmission avec étalement de spectre : schéma numérique.



Cela revient à effectuer la corrélation directement sur le signal reçu, sans changement de fréquence, par le code translaté en fréquence. La fonction de l'interpolateur est obtenue par une table de versions décalées en temps de ce code translaté : on remplace l'interpolation du signal par l'interpolation du code.

Ce regroupement des fonctions se résume en deux modifications : d'une part, travail sans démodulation et d'autre part, interpolation reportée sur le code.

La suppression de la démodulation supprime les calculs nécessités par le filtrage associé, mais elle augmente le nombre de points sur lesquels est calculée la corrélation. Dans notre application, compte tenu de la très faible fréquence porteuse utilisée relativement à la largeur de bande, le bilan est positif.

Le remplacement de l'interpolation du signal par celle du code procure une économie en puissance de calcul importante. Elle entraîne par contre une augmentation des besoins en mémoire pour la mémorisation de codes décalés. Cette augmentation est partiellement compensée par la suppression de la mémorisation des coefficients du filtre interpolateur et des tables de sinus du modulateur. De plus, la structure obtenue devient très simple.

Le codeur est également simplifié. Le code est mémorisé sous sa forme modulée et interpolée. Le codage ne nécessite alors plus que la multiplication de celui-ci par les bits d'information.

## 4.2 Choix du signal de codage.

Dans les systèmes de modulation à étalement de spectre type "CDMA", il est courant d'utiliser des séquences d'étalement binaire. Elles ont l'avantage de simplifier de façon importante l'opération de codage et surtout celle du corrélateur du décodage puisque les multiplications par des +1 ou -1 sont très simples à réaliser, en analogique comme en numérique. Dans un système tout numérique, l'intérêt de cette simplicité est moins évident, car le filtre interpolateur du sous-échantillonneur numérique a nécessairement des coefficients quelconques et devient ainsi plus coûteux que le corrélateur. Dans notre réalisation compacte, même si le code est initialement binaire, les codes translatés en fréquence, et interpolés, ne le sont plus. Il n'est donc pas nécessaire de

prendre des codes binaire, ce qui augmente les choix possibles.

## 4.3 Asservissement à la puissance

La récupération des bits de donnée est faite, au récepteur, par le signe de la corrélation entre le code modulé et le signal reçu. Si le code ajouté est trop faible, la composante audio de celui-ci peut entraîner une erreur. Or, comme on l'a signalé dans l'introduction, le codeur connaît ce signal audio. Il calcule donc, sur chaque bloc de signal correspondant à l'émission d'un bit de donnée, l'amplitude minimum du code ajouté pour être certain que, compte tenu du signal audio de ce bloc, la réception soit faite sans erreur. Cette stratégie doit être affinée pour assurer également que le corrélateur avance retard du système de synchronisation n'est pas lui non plus perturbé par ce signal audio.

Il est bien entendu possible de faire un asservissement plus élaboré, fréquence par fréquence, mais nous y avons renoncé pour conserver un décodeur très simple.

## 4.4 Synchronisation

La transmission du signal est supposée faite en analogique. La fréquence d'échantillonnage n'est donc pas synchronisée entre le codeur et le décodeur. De plus, nous nous intéressons plus particulièrement à un cas de transmission non permanente, pour une application de type télécommande. Le récepteur doit donc, avant d'effectuer le décodage de données, détecter la réception d'un message puis se synchroniser finement sur celui-ci. Pour donner au décodeur le temps d'effectuer ces deux phases, le codeur fait précéder le message d'un préambule composé de bits identiques, ce qui correspond à l'ajout du code avec un signe constant.

La détection de l'arrivée du message est délicate car, si l'asservissement de puissance fait par le codeur garantit que le signe de la corrélation est toujours bon, il ne peut garantir que le pic de corrélation soit fortement contrasté. On améliore le contraste en moyennant plusieurs blocs mais le nombre de blocs moyennés est limité par la précision relative des horloges entre le codeur et le décodeur : les horloges ne doivent pas se déphaser de façon notable pendant toute la durée du moyennage. De plus, ce moyennage n'est pas efficace pour tous les signaux audio. En particulier, des

fréquences pures multiples du rythme des bits perturberont le système.

Ce moyennage nécessite bien entendu un préambule plus long mais cet allongement autorise le décodeur à effectuer un calcul séquentiel de la corrélation pour chercher la position du maximum et permet donc une réalisation très économique de ce décodeur.

Le solution employée est de simuler dans le codeur le fonctionnement du décodeur afin de constater les fausses détections et de prolonger le préambule en conséquence.

## 5. Méthode de développement

Le logiciel MUSTIG a été utilisé dans tout le processus de développement. Grâce à sa programmation graphique très intuitive et sa facilité de manipulation, la mise au point des algorithmes a pu être réalisée de façon particulièrement rapide. Comme il a été souligné dans l'introduction, il était nécessaire de faire dès le début de l'étude des tests sur des signaux réels pour juger de la gêne apportée par le codage. Cela a été réalisé sur un simple PC en utilisant l'interfaçage de MUSTIG avec la carte son. Il est ainsi possible d'écouter le son encodé et simultanément d'effectuer un décodage du message numérique afin de vérifier la qualité de transmission par comptage des erreurs. L'utilisation de deux PC, un en codeur et l'autre en décodeur, permet de vérifier, grâce à leur asynchronisme, le fonctionnement des fonctions de synchronisation.

L'implantation finale du codeur et du décodeur sur des processeur de signal (DSP) a été réalisée de façon presque automatique. Les graphes MUSTIG ont été disposés de façon à isoler, sous forme des deux "macros", les fonctions de codage d'une part, de décodage d'autre part, de l'environnement de test et de l'interface avec la carte audio. Ces deux macros ont été automatiquement transcodées en "C" par MUSTIG. Sur les DSP du codeur et du décodeur, seul un programme relativement simple a été développé manuellement, pour la gestion du DSP, de ses interruptions et des convertisseurs analogiques. Ce programme est essentiellement une boucle d'appel au programme généré par MUSTIG.

L'avantage de l'automatisation de cette procédure est qu'il est très rapide de faire évoluer le traitement : il suffit d'effectuer une modification dans le programme MUSTIG, de la vérifier en fonctionnement sur le PC, puis de refaire la génération du code C.

## 6. Résultats et prospective

Nous avons obtenu un fonctionnement satisfaisant, avec une gêne généralement acceptable, dans un cas de messages courts (100 bits) pour une application de type télécommande.

La difficulté la plus importante dans ce cas est la phase d'acquisition et synchronisation : il est difficile de détecter de façon certaine l'émission du message. Cela conduit à l'utilisation d'un préambule de longueur comparable à celle du message. Malgré tout, il peut arriver que le signal audio

ressemble trop au signal codé, ce qui provoque des fausses détections. Il est possible de prédire ces situations dans le codeur puisque le signal audio est connu, et de prolonger le préambule en conséquence. Cela conduit à un allongement aléatoire du temps de transmission. Ces problèmes devraient être moins délicats dans le mode transmission continue. Nous expérimentons actuellement un tel système dans le but de transmettre, en superposition à un signal de parole, une information image très comprimée : seulement les paramètres des lèvres du locuteur.

## Références

- [1] F.Petitcola, R.Anderson, M.Kuhn *Information Hiding-A Survey* Proceeding of the IEE vol. 87 No. 7. July 1999
- [2] J.Proakis. *Digital Communications*. Mc. Graw Hill. 1995.
- [3] R.Crochiere, L.Rabiner *Interpolation and decimation of digital signals - A tutorial revue*. IEEE Proc, vol. 69, n.3, march 1981.
- [4] GRESILOG.S.A. *Logiciel MUSTIG* [www.gresilog.com](http://www.gresilog.com)