

Chapitre 5

Synchronisation

5.1 Introduction

Avant de traiter des problèmes de synchronisation, il est sans doute utile de rappeler qu'une transmission synchrone n'a d'intérêt que lorsque le canal de transmission est suffisamment stable au cours du temps. Sinon, l'avantage d'une détection cohérente par rapport aux méthodes de détection incohérente est perdu du fait de l'importance excessive des erreurs de synchronisation.

Un signal passe-bande autour d'une fréquence porteuse ν_0 porte une suite de symboles binaires¹ a_n . La réponse impulsionnelle du filtre d'émission, typiquement un demi Nyquist, est notée $s(t)$ et la durée symbole T . Le signal en bande translatée émis s'écrit :

$$e(t) = \text{Re} \left(e^{i2\pi\nu_0 t} \sum_n a_n s(t - nT) \right)$$

Après passage dans un canal de transmission linéaire, le récepteur observe :

$$r(t) = \text{Re} \left(e^{i2\pi\nu_0 t} \left(\sum_n a_n q(t - nT) + \eta(t) \right) \right)$$

La réponse $q(t)$ est constituée de la mise en cascade du filtre d'émission et du canal, $\eta(t)$ est un bruit additif. Lorsque l'enveloppe complexe de $r(t)$ est parfaitement récupérée, il est possible de construire une statistique suffisante en filtrant $\sum_n a_n q(t - nT) + \eta(t)$ par $q^*(-t)$, filtre adapté à $q(t)$, puis en échantillonnant de manière synchrone au rythme de un point par durée symbole. En notant $p(t) = q(t) * q^*(-t)$ et $n(t) = \eta(t) * q^*(-t)$, le signal discret ainsi construit s'écrit :

$$y_k = \sum_n a_n p(t - nT) + n(t) \Big|_{t=kT}$$

Notons qu'il est difficile voire impossible de réaliser le filtre adapté $q^*(-t)$ dans le domaine analogique du fait de la présence du canal inconnu. De plus, une mise en œuvre

¹ Les algorithmes décrits dans ce chapitre s'adaptent très simplement aux modulations PSK ou QAM. Nous nous limitons essentiellement aux signaux BPSK pour simplifier la présentation.

numérique offre plus de souplesse. Pour qu'une mise en œuvre purement numérique du filtre adapté soit possible sans perte d'optimalité, il suffit d'échantillonner le signal en bande de base en respectant le théorème d'échantillonnage de Shannon. Cela suppose de

préfiltrer le signal dans la bande $\left[-\frac{1}{2T_e}, +\frac{1}{2T_e}\right]$ avant de l'échantillonner aux instants

kT_e . Par commodité, la période d'échantillonnage T_e sera un sous-multiple de T ($T = \lambda T_e$ avec λ entier, typiquement 2 ou 4) ; notons $z_k = z(kT_e)$ le signal discret ainsi obtenu et $h(t)$ la réponse impulsionnelle qui relie z aux données transmises.

L'objectif de ce chapitre est de donner un aperçu des techniques de base pour la récupération du rythme et de la porteuse.

Pour aborder correctement les problèmes de synchronisation, il faut tout d'abord modéliser le signal à temps continu sortant du démodulateur en présence d'une erreur de démodulation et d'un décalage temporel.

Ce signal est de la forme :

$$y(t) = e^{i\xi(t)} \sum_n a_n p(t - nT - \tau) + n(t)$$

où le retard $\tau = \varepsilon T$ est une fraction de la durée symbole et où la phase $\xi(t)$ modélise l'erreur liée au changement de fréquence. Très souvent, l'erreur de phase est bien approchée par une dérive linéaire : $\xi(t) = 2\pi\delta t + \xi_0$ où δ représente une erreur de fréquence et ξ_0 une erreur sur la phase.

Deux dispositifs sont utilisés pour lutter contre les asynchronismes :

1. Un dispositif de récupération de porteuse qui assure que le démodulateur complexe est parfaitement synchronisé avec le modulateur.
2. Un dispositif de récupération de rythme qui estime les instants d'échantillonnage optimaux.

Les systèmes de synchronisation estiment pour les compenser l'erreur de phase $\xi(t)$ ainsi que le décalage temporel τ . Lorsque ces deux paramètres sont parfaitement estimés, on retrouve le modèle synchronisé :

$$y_k = y(kT + \tau) = \sum_n a_n p[(k - n)T] + n(kT + \tau)$$

Dans une approche numérique, le signal préfiltré échantillonné s'écrit :

$$z_k = z(kT_e) = e^{i\xi(kT_e)} \sum_n a_n h(kT_e - nT - \tau) + n(kT_e)$$

Dans ce chapitre nous supposons que le système de transmission respecte le critère de Nyquist. Typiquement, cela signifie qu'un demi-Nyquist est utilisé du côté de l'émetteur ainsi qu'au récepteur et que la réponse fréquentielle du canal est constante dans la bande utile. Dans ce cas, après normalisation de l'amplitude, le modèle synchronisé se réduit à $y_k = y(kT + \tau) = a_k + n_k$ avec $n_k = n(kT + \tau)$ et la séquence y_k

peut directement servir d'entrée à un décodeur, si un code correcteur d'erreur a été utilisé à l'émission, ou à une simple décision si aucun codage n'a été mis en œuvre.

Dans le cas plus complexe d'un canal non idéal, il faudra également détecter les symboles à l'aide d'un détecteur de Viterbi ou de toute autre méthode. Dans ce cas, la synchronisation symbole et l'estimation du canal sont étroitement liées.

5.1.1 Synchronisation totalement numérique

Tête de réception optimale

Lorsque le signal sortant du démodulateur complexe est affecté d'une erreur de rythme et d'une erreur de phase, la tête de réception optimale pour un récepteur totalement numérique est celle de la figure suivante :

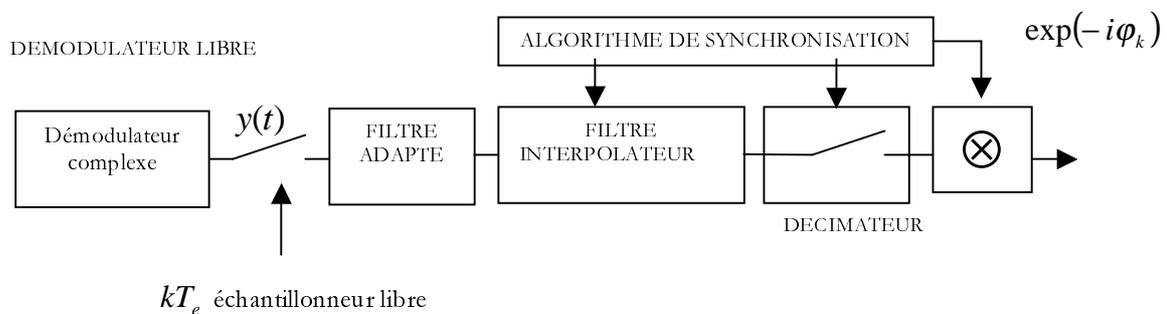


Figure 5.1 - Tête de réception optimale numérique

Dans cette structure, les oscillateurs locaux utilisés pour la démodulation comme pour l'échantillonnage sont libres, c'est-à-dire non asservis par une quelconque commande. Cette situation est à opposer à celle de récepteurs plus traditionnels pour lesquels les oscillateurs sont asservis soit par un système analogique (tête de réception analogique) soit par un algorithme numérique (systèmes hybrides analogique-numérique).

Dans un récepteur numérique, le signal reçu est démodulé autour de la fréquence porteuse attendue ; ce faisant, une erreur de phase subsiste ; un traitement en bande de base est à même de compenser cette fluctuation de phase.

Le signal issu du démodulateur est ensuite traité par le filtre adapté à l'ensemble de la chaîne de transmission, puis échantillonné. Une première possibilité consiste à aligner les instants d'échantillonnage sur les symboles transmis grâce à un asservissement de l'échantillonneur. L'alternative numérique consiste à employer un échantillonneur non asservi pour ensuite procéder à un changement numérique du rythme. Pour que le ré-échantillonnage numérique du signal soit possible, la fréquence d'échantillonnage doit respecter le critère de Shannon, c'est-à-dire être strictement supérieure à deux fois la fréquence maximale contenue dans le signal. L'excès de bande (roll-off) des signaux de communication est généralement compris entre 0 et 100% et la fréquence d'échantillonnage doit ainsi être supérieure à deux points par symbole. On utilise typiquement une valeur de deux ou de quatre fois la cadence symbole. Les oscillateurs utilisés par l'émetteur et par le récepteur ayant des fréquences incommensurables, le

dispositif numérique de changement de rythme doit utiliser un filtre interpolateur variant en temps capable d'interpoler dans l'intervalle $[-T_e/2, +T_e/2]$ suivi d'un système de décimation permettant d'effectuer un décalage d'un nombre entier de symboles.

Ce schéma de réception suppose que les paramètres de synchronisation, rythme et porteuse, sont estimés par ailleurs. L'objectif de ce chapitre est de fournir quelques schémas élémentaires pour l'estimation de ces deux paramètres.

5.1.2 Cas particulier : récupération de porteuse

Lorsque l'échantillonnage est parfait $y_k = \left(\sum_n a_n p(t - nT) + n(t) \right) e^{i\xi(t)} \Big|_{t=kT}$ se réduit²

à $y_k = a_k e^{i\xi_k} + b_k$ car le système respecte le critère de Nyquist. Il faut alors estimer la phase ξ_k à partir du signal discret y_k . Le bruit b_k est supposé blanc, gaussien, centré et circulaire de variance σ^2 .

5.1.3 Cas particulier : récupération de rythme

En l'absence d'erreur de phase, la sortie du démodulateur complexe est de la forme $y(t) = \sum_n a_n p(t - nT - \tau) + n(t)$ et il faut estimer les instants $t_k = kT$ à partir de $y(t)$ ou du signal échantillonné $\{z(kT_e)\}$.

5.1.4 Cas général : estimation conjointe

Il est également possible d'estimer conjointement plusieurs paramètres de synchronisation. Par exemple, l'erreur de démodulation peut être essentiellement une erreur sur la fréquence de démodulation et le décalage de rythme est généralement supposé constant sur l'intervalle d'observation. Dans ce cas, les deux paramètres peuvent être déduits d'une même procédure d'estimation.

5.2 Récupération de porteuse

Il convient de distinguer deux cas qualitativement différents selon l'ordre de grandeur de l'erreur sur la porteuse.

5.2.1 Décalages forts : estimation de fréquence

Typiquement, dans des cas tels qu'une communication par satellite, l'erreur sur la porteuse liée au décalage Doppler peut être grande vis à vis de la largeur de bande du signal, il est alors nécessaire d'estimer et de compenser ce décalage avant tout traitement en bande de base. Pour s'en convaincre, il suffit de considérer le cas extrême dans lequel le décalage est si important que le signal reçu sort de la bande passante du filtre de réception. Dans ce cas, il est clair qu'aucun traitement en bande de base ne résoudra le

² A un facteur constant près.

problème. Des systèmes capables d'estimer de gros écarts fréquentiels existent : on peut par exemple construire une mesure de l'erreur de fréquence en mesurant la différence d'énergie entre deux bandes de part et d'autre de la porteuse et utiliser cette mesure pour asservir la démodulation. C'est la technique des filtres duaux. Cette approche est équivalente aux systèmes connus sous le nom de quadricorrélateurs. De nombreuses variantes ont été proposées ; nous ne détaillons pas ce problème plus avant, le lecteur intéressé peut consulter [4,5] et ses références secondaires.

5.2.2 Décalages faibles : poursuite de phase

A partir de maintenant, nous considérons que le décalage fréquentiel résiduel est suffisamment faible pour qu'il soit possible d'envisager un traitement numérique en bande de base.

Fréquence pure

Approche du maximum de vraisemblance (MV)

Supposons que le signal n'est pas modulé ($a_k = 1$ pour tout k) et que la phase est constante ; les observations sont alors de la forme $y_k = e^{i\xi} + b_k$. La vraisemblance des $k+1$ observations $Y_k^T = (y_0, \dots, y_k)$ s'écrit :

$$P(Y_k, \xi) = \prod_{n=0}^k \frac{1}{\sigma^2 \pi} \exp\left(-\frac{|y_n - e^{i\xi}|^2}{\sigma^2}\right)$$

La solution du maximum de vraisemblance φ_k est la valeur du paramètre ξ qui rend la probabilité $P(Y_k, \xi)$ maximale, celle qui explique au mieux les observations. Elle vérifie l'équation implicite $\frac{\partial}{\partial \xi} \log P(Y_k, \xi) = 0$, c'est-à-dire $\sum_{n=0}^k \text{Im}(y_n e^{-i\varphi_k}) = 0$.

Dans ce cas très simple, l'estimateur s'écrit explicitement :

$$\varphi_k = \text{Arc tan} \left(\frac{\sum_{n=0}^k \text{Im}(y_n)}{\sum_{n=0}^k \text{Re}(y_n)} \right)$$

Boucle de phase d'ordre un

Pour des problèmes d'estimation non linéaires, il est le plus souvent impossible d'obtenir une forme explicite pour l'estimateur du maximum de vraisemblance et l'on doit se contenter d'approcher la solution par une procédure itérative du type :

$$\varphi_{k+1} = \varphi_k + \gamma \text{Im}(y_{k+1} e^{-i\varphi_k})$$

Cette forme approchée correspond au schéma de la figure suivante.

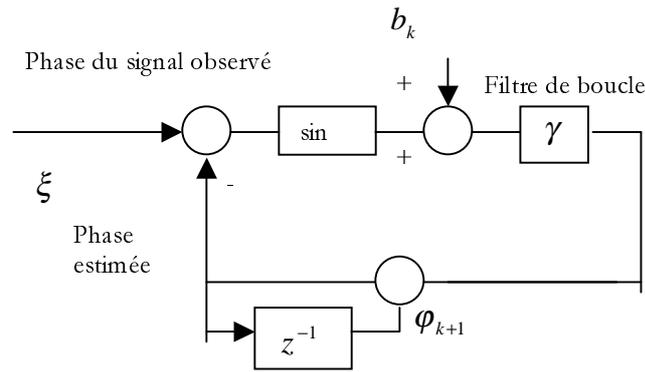


Figure 5.2 - Boucle à verrouillage de phase

Ce dispositif est appelé « boucle à verrouillage de phase ». Remarquons qu'il s'agit d'une approximation obtenue en supposant la phase constante et que le paramètre γ doit être choisi. Pratiquement, γ doit être choisi en fonction du rapport signal à bruit et de la vitesse de variation de la phase (voir plus loin).

Adaptation au cas d'une dérive de phase

Voyons tout d'abord comment la boucle de phase peut être adaptée pour tirer parti d'une connaissance *a priori* sur l'évolution de la phase. Commençons pour cela par spécifier un modèle statistique réaliste pour décrire les fluctuations de la phase.

En pratique, deux oscillateurs n'ont jamais exactement la même fréquence. Il est réaliste de supposer qu'une petite erreur de fréquence ε_k existe, cette fréquence pouvant elle-même être affectée d'une faible gigue aléatoire modélisée par une variable gaussienne centrée w_k^2 . Ajoutons que la phase ξ_k est également perturbée par une gigue gaussienne w_k^1 . Ces considérations conduisent au modèle d'évolution suivant :

$$\xi_{k+1} = \xi_k + \varepsilon_k + w_k^1$$

$$\varepsilon_{k+1} = \varepsilon_k + w_k^2$$

Cette information *a priori* permet de raffiner la boucle de phase et de la remplacer par l'algorithme suivant³ :

$$\varphi_{k+1} = \varphi_k + \delta_k + \gamma_1 E_k$$

$$\delta_{k+1} = \delta_k + \gamma_2 E_k$$

Pour la boucle de phase décrite précédemment $E_k = \text{Im}(y_{k+1} e^{-i\varphi_k})$. φ_k est un estimateur de la phase et δ_k un estimateur de sa dérive moyenne entre deux observations successives. Cet algorithme diffère du précédent par un filtrage de l'erreur

³ Cet algorithme s'obtient en calculant le filtre de Kalman étendu associé au modèle d'évolution et à l'équation d'observation $y_k = e^{i\xi_k} + b_k$

E_k à l'aide d'un filtre de réponse $H(z^{-1}) = \gamma_1 + \frac{\gamma_2}{1-z^{-1}}$. Pour mettre en évidence ce « filtre de boucle », on écrit volontiers cet algorithme sous la forme :

$$\varphi_{k+1} = \varphi_k + \left(\gamma_1 + \frac{\gamma_2}{1-z^{-1}} \right) E_k$$

On le nomme « boucle de phase du second ordre ». L'intégrateur supplémentaire qui apparaît pour $\gamma_2 \neq 0$ permet de suivre sans erreur statique une dérive linéaire de phase.

Cette modification de l'algorithme initial rend le système plus performant en poursuite lorsque le modèle concorde avec la réalité. Par contre, la robustesse est moindre : clairement, si l'information *a priori* est fautive, il vaut mieux en rester à une boucle d'ordre un. De plus l'accrochage est plus délicat pour la boucle du second ordre et il est généralement nécessaire de lui adjoindre une aide pour l'initialiser avec une valeur de la dérive proche de la réalité.

Signal modulé à deux états de phase

Nous allons maintenant adapter les méthodes précédentes au cas de signaux modulés par déplacement de phase à deux états ($a_k = \pm 1$).

Approche du maximum de vraisemblance

Les observations $y_k = a_k e^{i\xi} + b_k$ dépendent maintenant des symboles binaires $a_k = \pm 1$. La vraisemblance d'un bloc d'observations $Y_k^T = (y_0, \dots, y_k)$ conditionnée par les symboles émis s'écrit :

$$P(Y_k, \xi / a_1, \dots, a_k) = \prod_{n=0}^k \frac{1}{\sigma^2 \pi} \exp \left(- \frac{|y_n - a_n e^{i\xi}|^2}{\sigma^2} \right)$$

Ces symboles étant inconnus du récepteur, celui-ci doit s'en affranchir. En supposant les a_k indépendants, équiprobables et uniformément répartis sur $\{-1, +1\}$, il est possible d'intégrer sur les a_k inconnus :

$$P(Y_k, \xi / a_1, \dots, a_k) = \left(\frac{1}{\sigma^2 \pi} \right)^{k+1} \prod_{n=0}^k \left\{ \begin{array}{l} \frac{1}{2} \exp \left(- \frac{|y_n|^2 - 2 \operatorname{Re}(y_n e^{-i\xi}) + 1}{\sigma^2} \right) \\ + \frac{1}{2} \exp \left(- \frac{|y_n|^2 + 2 \operatorname{Re}(y_n e^{-i\xi}) + 1}{\sigma^2} \right) \end{array} \right\}$$

L'annulation de la log vraisemblance conduit à la solution implicite :

$$\sum_{n=0}^k \frac{2}{\sigma^2} \operatorname{Im}(y_n e^{-i\varphi_k}) \tanh \left(\frac{2}{\sigma^2} \operatorname{Re}(y_n e^{-i\varphi_k}) \right) = 0$$

et à l'algorithme approché :

$$\varphi_{k+1} = \varphi_k + \gamma \operatorname{Im}(y_{k+1} e^{-i\varphi_k}) \tanh \left(\frac{2}{\sigma^2} \operatorname{Re}(y_{k+1} e^{-i\varphi_k}) \right)$$

Selon le rapport signal sur bruit, le terme $\tanh\left(\frac{2}{\sigma^2}\operatorname{Re}(y_{k+1}e^{-i\varphi_k})\right)$ peut être approché essentiellement de deux façons.

1. **Rapport signal sur bruit fort.** Lorsque σ^2 est petit, l'argument de la tangente hyperbolique est grand et celle-ci est bien approchée par une fonction signe. Le terme $\tanh\left(\frac{2}{\sigma^2}\operatorname{Re}(y_{k+1}e^{-i\varphi_k})\right)$ est donc une « décision dure » \tilde{a}_{k+1} quant au symbole émis a_{k+1} et l'algorithme se réduit à $\varphi_{k+1} = \varphi_k + \gamma \operatorname{Im}(\tilde{a}_{k+1}y_{k+1}e^{-i\varphi_k})$. Les symboles inconnus sont éliminés en remodulant, c'est à dire en multipliant par les décisions \tilde{a}_{k+1} . D'où le nom de boucle à remodulation donné à cet algorithme. Cet algorithme se généralise à des constellations QAM ou PSK à plus grand nombre d'états de la manière suivante :

$$\varphi_{k+1} = \varphi_k + \gamma \operatorname{Im}(\tilde{a}_{k+1}^* y_{k+1} e^{-i\varphi_k})$$

2. **Rapport signal sur bruit faible.** Lorsque son argument est faible, la tangente hyperbolique est approximativement linéaire et l'algorithme s'approche de $\varphi_{k+1} = \varphi_k + \gamma \operatorname{Im}(y_{k+1}^2 e^{-i2\varphi_k})$. La mise au carrée est une autre manière d'éliminer la modulation. Cet algorithme est appelé « boucle de Costas ». Il se généralise à des modulations de phase à N états comme suit :

$$\varphi_{k+1} = \varphi_k + \gamma \operatorname{Im}(y_{k+1}^N e^{-iN\varphi_k})$$

Ces différents algorithmes peuvent être adaptés à des dérives de phase de forme connue *a priori* en ajoutant un filtre de boucle exactement de la même manière que dans le cas de l'estimation d'une fréquence pure. On présentera en section 5.5.2 une implantation en Matlab et quelques résultats de simulation des boucles de Costas d'ordre 1 et 2, pour la récupération de porteuse sur une modulation $\pi/4$ QPSK.

Réglage des pas

La qualité du fonctionnement des algorithmes précédents dépend du réglage du γ (pour les boucles d'ordre un) ou des pas γ_1, γ_2 (pour les boucles d'ordre deux). En toute rigueur, il est nécessaire de distinguer deux modes de fonctionnement :

- **L'accrochage ou convergence initiale.** C'est la première étape. La boucle est initialisée à l'aide de valeurs de la phase (φ_0) et de la dérive (δ_0) qui peuvent être très différentes des vraies valeurs. Le comportement est alors non linéaire et difficile à étudier théoriquement [4,5]. D'un point de vue pratique le comportement des boucles de phase est assez médiocre durant cette première étape et il est préférable d'aider la boucle en lui fournissant une estimée grossière de la dérive (estimateur de fréquence).
- **La poursuite.** La boucle est verrouillée sur la phase et suit ses variations. Les boucles à verrouillage de phase sont très efficaces dans cette phase de leur fonctionnement. Un réglage précis des pas les rend très proches de l'optimum théorique.

La distinction entre ces deux modes de fonctionnement extrêmes cache une particularité importante du comportement d'une boucle de phase : les glissements de cycle. Il peut arriver que la phase estimée varie brutalement, conduisant la boucle à

changer de détermination. Dans ce cas, une nouvelle acquisition se produit avant que la boucle ne se stabilise à nouveau au voisinage de sa nouvelle trajectoire. En pratique, les glissements de cycle doivent être inexistantes (extrêmement rares) pour que le fonctionnement de la boucle puisse être qualifié de correct.

Lorsque le modèle d'évolution de la phase est connu, les pas optimaux du point de vue de la poursuite des non-stationnarités ainsi que les performances obtenues pour ce réglage optimal peuvent être calculées. Il est même possible de construire des algorithmes qui estiment conjointement la phase et les valeurs optimales des pas (cf. [1], [2], [3]).

5.3 Récupération du rythme

Conceptuellement, la synchronisation est un cas particulier de l'estimation du canal de transmission dans lequel le canal est connu à un paramètre inconnu près. Remarquons également que dans les systèmes qui traitent les données reçues par bloc et pour lesquelles le canal de transmission ne peut pas être supposé constant sur la bande utile, le problème de la synchronisation symbole est directement intégré à la procédure d'estimation du canal. La prise en considération explicite de la synchronisation symbole est particulièrement importante lorsque le décalage de rythme τ ne peut pas être supposé constant sur la durée d'un bloc, dans les transmissions en flot continu ou lorsque le canal est proche de l'idéal. Nous nous plaçons à nouveau dans ce dernier cas.

Pour des modulations de type PSK ou QAM, le signal reçu ne contient aucune périodicité liée à la période symbole, il n'est donc pas possible de récupérer le rythme symbole par une simple boucle à verrouillage de phase. Nous verrons qu'une sous-catégorie importante des systèmes de synchronisation est celle des méthodes dites de « raie spectrale » dans lesquelles la cyclostationnarité du signal est transformée en une véritable raie spectrale, laquelle peut être suivie par un système bouclé, tel qu'une boucle de phase, ou par un simple filtrage passe-bande centré autour de la fréquence symbole attendue.

En supposant que la phase est parfaitement compensée, le signal en sortie du démodulateur complexe s'écrit :

$$y(t) = \sum_n a_n p(t - nT - \tau) + n(t)$$

L'objectif est d'échantillonner $y(t)$ aux instants $t_k = kT + \tau$. Pour cela, la fréquence $1/T$ d'émission des symboles doit être estimée et le peigne d'échantillonnage de fréquence $1/T$ doit être aligné sur celui qu'utilise l'émetteur.

5.3.1 Approche du maximum de vraisemblance

Comme nous l'avons fait pour l'estimation de la phase, nous utilisons ici l'approche du maximum de vraisemblance pour déduire une solution approchée au problème de synchronisation sur un bloc d'observations. Nous approchons également les solutions obtenues par des systèmes bouclés simples. A titre d'illustration, nous montrons comment certaines de ces boucles de synchronisation peuvent être obtenues à l'aide d'autres approches (algorithmes adaptatifs par exemple).

Comme nous l'avons déjà mentionné, il est souvent plus intéressant d'échantillonner le signal reçu après un simple préfiltrage pour réaliser numériquement le filtre adapté, cette approche permet également d'étudier la synchronisation dans le domaine numérique.

Les échantillons prélevés en sortie du préfiltrage s'écrivent :

$$z_k = z(kT_e) = e^{i\xi(kT_e)} \sum_n a_n h(kT_e - nT - \tau) + n(kT_e)$$

La variation de phase $\xi(t)$ est typiquement de la forme $\xi(t) = 2\pi\delta t + \xi_0$. Nous nous intéressons à l'estimation du rythme sous différentes hypothèses concernant l'erreur de fréquence δ . Le problème consiste à estimer les erreurs de synchronisation à partir des z_k . Contrairement au problème de récupération de porteuse dans lequel le modèle d'observation est instantané, nous devons ici travailler avec un modèle convolutif dans lequel chaque échantillon observé dépend de plusieurs symboles transmis. En supposant la réponse $h(t)$ approximativement finie, une suite $\underline{z}^T = (z_{-k}, \dots, z_{+k})$ de $2k+1$ observations dépend de $2L+1$ symboles selon une relation linéaire de la forme $\underline{z} = \underline{H}\underline{x} + \underline{n}$ avec $\underline{n}^T = (n_{-k}, \dots, n_{+k})$, $\underline{H} = [h_{-L}, \dots, h_{+L}]$, $\underline{x}^T = [a_{-L}, \dots, a_{+L}]e^{i\xi_0}$ et

$$\underline{h}_\lambda^T = [h(-kT_e - \lambda T - \tau)e^{-i2\pi\delta kT_e}, \dots, h(+kT_e - \lambda T - \tau)e^{+i2\pi\delta kT_e}]$$

D'où, pour un bruit blanc et gaussien, la vraisemblance de l'observation :

$$\Lambda(\underline{z}|\tau, \delta, \underline{x}) \propto \exp\left\{-\frac{1}{\sigma^2} \|\underline{z} - \underline{H}\underline{x}\|^2\right\}$$

En développant la vraisemblance, il vient :

$$\Lambda(\underline{z}|\tau, \delta, \underline{x}) \propto \exp\left\{-\frac{1}{\sigma^2} \underline{z}^H \underline{z}\right\} \exp\left\{-\frac{1}{\sigma^2} \underline{x}^H \underline{H}^H \underline{H} \underline{x}\right\} \exp\left\{\frac{2}{\sigma^2} \text{Re}(\underline{z}^H \underline{H} \underline{x})\right\}$$

Il est clair que le premier terme ne dépend pas des paramètres de synchronisation. Un simple développement montre qu'il en est de même pour le deuxième terme (lorsque k est grand). Ainsi, l'optimisation peut être faite en considérant seulement :

$$\Lambda(\underline{z}|\tau, \delta, \underline{x}) \rightarrow \exp\left\{\frac{2}{\sigma^2} \text{Re}(\underline{z}^H \underline{H} \underline{x})\right\}$$

De cette expression, il est possible de tirer des systèmes de synchronisation utilisant les données (dits DA, Data Aided), des systèmes pilotés par des décisions quant aux données transmises (dits DD, Decision Directed) ou des systèmes « aveugles » (dits NDA, Non Data Aided). Les deux premiers cas étant assez immédiats, nous nous concentrons surtout sur les systèmes NDA. Nous donnerons tout de même un exemple de système DA ou DD dans le cas des signaux à spectre étalé.

De même que pour la récupération de phase, la dépendance par rapport aux symboles porteurs d'information peut être éliminée en intégrant sur la constellation. Malheureusement, cette approche conduit ici à des calculs inextricables et des simplifications doivent être faites.

La première simplification classique consiste à **supposer le rapport signal à bruit faible** pour développer l'exponentielle en série, ainsi :

$$\Lambda(\underline{z}|\tau, \delta, \underline{x}) \rightarrow A + B \operatorname{Re}(\underline{x}^H \underline{H}^H \underline{z}) + C \operatorname{Re}^2(\underline{x}^H \underline{H}^H \underline{z})$$

A , B et C sont des constantes indépendantes des paramètres de synchronisation. Des systèmes NDA peuvent être obtenus en intégrant sur la constellation, c'est la méthode qui a été utilisée pour obtenir des boucles de phase adaptées aux signaux BPSK (Costas et remodulation). Cette méthode permet de définir des algorithmes adaptés à des constellations particulières. Il est également possible d'obtenir des systèmes plus généraux en faisant une hypothèse approximativement valable pour un grand nombre de constellations. Cela est par exemple possible en supposant la phase équirépartie sur l'intervalle $[0, 2\pi[$.

Sous cette hypothèse, une intégration donne :

$$\Lambda(\underline{z}|\tau, \delta) \rightarrow \underline{z}^H \underline{H} \underline{H}^H \underline{z} = \|\underline{H}^H \underline{z}\|^2$$

Selon que l'erreur fréquentielle est connue ou non, différents dispositifs pratiques pour la synchronisation peuvent être déduits de cette expression.

Erreur sur la fréquence connue

Pour de grands échantillons, la fonction de vraisemblance $\Lambda(\underline{z}|\tau, \delta) \rightarrow \|\underline{H}^H \underline{z}\|^2$ représente l'énergie en sortie du filtre adapté échantillonné à un point par symbole (aux instants $kT + \tau$) après compensation de l'erreur de fréquence. Ce résultat peut donner lieu à une implantation itérative du type « boucle de synchronisation ». Le principe d'une boucle de synchronisation consiste à reconstruire localement, au niveau du récepteur, un signal de référence périodique puis à amener cette référence en synchronisme avec le signal reçu. Une mesure de l'erreur entre la référence locale et le signal reçu permet d'asservir le signal de référence donc l'échantillonnage. En dérivant la vraisemblance, nous choisissons le signal d'erreur :

$$E(\tau) = \frac{\partial}{\partial \tau} (y_k(\tau) y_k^*(\tau)) = 2 \operatorname{Re} \left(y_k^*(\tau) \frac{\partial}{\partial \tau} y_k(\tau) \right)$$

où $y_k(\tau) = y(kT + \tau)$ représente la sortie du filtre adapté pour le retard τ après compensation de l'erreur de fréquence. Cette erreur $E(\tau)$ conduit à la boucle de synchronisation suivante :

$$\tau_{k+1} = \tau_k - \gamma \operatorname{Re} \left(y_k^*(\tau_k) \frac{\partial}{\partial \tau} y_k(\tau_k) \right)$$

En pratique, la dérivée peut être approchée par une simple différence finie (cf. les simulations ; cf. aussi l'exemple de la boucle « early-late » pour les signaux à spectre étalé).

De même que pour les boucles de phase, le pas γ peut être remplacé par un filtre ; le réglage de ce filtre peut être effectué en tenant compte des informations *a priori*

disponibles quant à l'évolution du rythme. En 5.5.3, nous présentons l'implantation Matlab ainsi que quelques résultats de simulation pour un tel dispositif de récupération de rythme.

Bien que la sortie du filtre adapté soit exempte d'interférences entre symboles lorsqu'on l'échantillonne de manière synchrone au rythme de un point par symbole, il n'en est pas de même pour la dérivée de ce filtre. Ainsi, le système de synchronisation symbole obtenu grâce à l'approximation rapport signal à bruit faible souffre d'un **bruit propre** présent même en l'absence de bruit sur l'observation. Ceci constitue la limitation majeure de ces systèmes simples.

Autre approche : algorithmes adaptatifs.

Dans le paragraphe précédent, nous avons abordé le problème de la synchronisation symbole sous forme « bloc » tout en précisant comment il est possible de déduire des boucles de synchronisation à partir d'une solution en bloc. Lorsque l'objectif est d'obtenir une simple boucle de synchronisation, cette approche n'est pas la seule possible. A titre d'illustration, voyons comment la boucle de synchronisation précédente peut être obtenue par d'autres voies. Pour plus de simplicité, supposons l'erreur de fréquence connue et déjà compensée.

L'objectif est d'amener $y(kT + \tau) = y_k(\tau)$, sortie du filtre adapté avancé de τ , en synchronisme avec les données transmises. En supposant dans un premier temps les données connues du récepteur (apprentissage), considérons la fonction de coût naturelle :

$$J(\tau|a_k) = E|y_k(\tau) - a_k|^2$$

Un algorithme du gradient stochastique pour la minimisation de J conduit à la boucle de synchronisation suivante :

$$\tau_{k+1} = \tau_k - \gamma \operatorname{Re} \left((y_k^*(\tau_k) - a_k^*) \frac{\partial}{\partial \tau} y_k(\tau_k) \right)$$

Ce système est du type DA. Il peut également être utilisé en mode DD en remplaçant a_k par une décision prise en sortie du filtre adapté. Pour obtenir une version NDA, il suffit d'intégrer sur la constellation des a_k , ceci permet de retrouver l'algorithme obtenu par approximation du maximum de vraisemblance.

Bien sûr, de nombreuses autres boucles de synchronisation peuvent être trouvées dans la littérature spécialisée. Parmi les procédures les plus efficaces, Mueller & Müller [4, 5] ont proposé une boucle de synchronisation pilotée par les décisions \hat{a}_k quant aux symboles transmis qui utilise l'erreur $\tilde{a}_{k-1}y(kT + \hat{\tau}) - \tilde{a}_k y(kT - T + \hat{\tau})$.

Erreur sur la fréquence inconnue

Lorsque qu'aucune information *a priori* n'est disponible quant à l'erreur sur la fréquence de démodulation, cette erreur peut être supposée uniformément répartie sur la bande $\left[-\frac{1}{2T_e}, +\frac{1}{2T_e}\right]$. En intégrant par rapport à la loi de δ , il vient :

$$E_{\delta} \{ \Lambda(\underline{z}|\tau, \delta) \} \rightarrow \underline{z}^H E_{\delta} \left[\underline{H} \underline{H}^H \right] \underline{z}$$

Il est facile de voir que la matrice $E_{\delta} \left[\underline{H} \underline{H}^H \right]$ est diagonale avec comme élément diagonal $q_{\tau}(l) = \sum_{n=-L}^{+L} |h(lT_e + nT - \tau)|^2$. Dans ces conditions, la vraisemblance s'écrit :

$$\Lambda(\underline{z}|\tau, \delta) \rightarrow \text{Re} \left(\sum_{l=-k}^{+k} q_{\tau}(l) |z_l|^2 \right)$$

$q_{\tau}(l)$ est une référence locale qui tend à être T/T_e -périodique lorsque $l \rightarrow \infty$. En approchant cette référence par des formes plus simples (premier harmonique par exemple), il est possible d'obtenir des algorithmes plus faciles à implanter comme l'algorithme de Oerder [6].

5.3.2 Méthode de la raie spectrale

L'approche du maximum de vraisemblance, quoique théoriquement satisfaisante, ne conduit à des systèmes pratiquement utilisables qu'au prix de simplifications drastiques. Les synchroniseurs obtenus perdent ainsi beaucoup des bonnes propriétés d'un véritable estimateur au sens du MV. Cette assertion est particulièrement claire lorsque l'on s'interroge sur l'origine du bruit propre des synchroniseurs : ce défaut majeur est une conséquence directe de l'approximation rapport signal à bruit faible.

Dans ces conditions, il est raisonnable de construire des systèmes de synchronisation en tirant profit de diverses propriétés statistiques des signaux de communications. Une propriété fondamentale pour l'estimation du rythme d'émission des symboles est la cyclostationnarité du signal $y(t)$, c'est-à-dire le fait que ses propriétés statistiques varient de manière périodique au cours du temps. Les plus simples des méthodes qui utilisent la cyclostationnarité, ou méthodes cycliques, sont dites « méthodes de raie spectrale ».

Leur principe consiste à faire apparaître une fréquence pure liée à la fréquence symbole puis à suivre cette raie spectrale. La variance de $y(t)$ s'écrit :

$$E|y(t)|^2 = E(a_k^2) \sum_n |p(t - nT)|^2 + \sigma^2$$

C'est une fonction T-périodique dont le $n^{\text{ème}}$ coefficient de Fourier vaut :

$$c_n = \frac{E(a_k^2)}{T} \int_{-\infty}^{+\infty} P(v) P^* \left(v - \frac{n}{T} \right) dv + \sigma^2 \delta_n$$

où $P(v)$ désigne la transformée de Fourier de $p(t)$.

Pour un excès de bande⁴ inférieur à 100%, $P(v)$ est contenu dans l'intervalle $[-1/T, +1/T]$ et seuls les termes c_0 , c_{-1} et c_{+1} peuvent être non nuls. Pour que c_{-1} et c_{+1} soient non nuls, il faut que le signal présente un excès de bande (roll-off non nul). Dans ce cas, la variance instantanée fait apparaître une raie spectrale de période T .

Ainsi, une mise au carré du signal fait apparaître une véritable périodicité qu'il est possible de suivre simplement.

5.3.3 Interpolation numérique

Les méthodes décrites jusqu'ici utilisent le signal à temps continu $y(t)$ ou une version correctement suréchantillonnée $z_k = z(kT_e)$. En pratique, il faudra échantillonner le signal à des instants précis soit physiquement, en pilotant l'échantillonneur à l'aide du retard estimé $\hat{\tau}$, soit numériquement en interpolant le signal numérique $z(kT_e)$. Cette dernière solution tend à se généraliser du fait des progrès de la microélectronique. Nous donnons maintenant les bases concernant les techniques d'interpolation numérique.

Principe

Après échantillonnage à la cadence T_e , les échantillons observés $z(kT_e)$ correspondent à l'échantillonnage du signal retardé de τ . Pour reconstituer les échantillons associés au signal non retardé, il suffit de translater numériquement d'un nombre non entier d'échantillons. Une simple application de la formule de Shannon donne une solution à ce problème lorsque la cadence d'échantillonnage T_e respecte le critère de Shannon :

$$z(kT + \tau) = \sum_{n=-\infty}^{+\infty} z(nT_e) \text{Sinc} \left(\frac{\pi}{T_e} (kT + \tau - nT_e) \right)$$

Pratiquement, deux problèmes majeurs se posent :

1. Pour implanter un tel filtre, le sinus cardinal doit être tronqué et il devient nécessaire de centrer correctement la réponse de ce filtre sur l'échantillon à estimer. Pour cela, on limite l'interpolation à l'intervalle $[-T_e/2, +T_e/2]$ et l'on compense la partie entière du décalage indépendamment.
2. Les périodes d'échantillonnage T_e et T sont incommensurables et le filtre interpolateur devient variant en temps. Il est alors impératif de séparer la partie fractionnaire et la partie entière du retard d'interpolation et d'exprimer les instants utiles en fonction de la période T_e , seule horloge connue du récepteur.

Il faut déterminer les valeurs du signal aux instants $nT + \tau = T_e \left(n \frac{T}{T_e} + \varepsilon \frac{T}{T_e} \right)$, en décomposant $n \frac{T}{T_e} + \varepsilon \frac{T}{T_e}$ en sa partie entière n_n et sa partie fractionnaire v_n , il est clair

⁴ roll-off

que cela nécessite d'interpoler au décalage v_n puis de sélectionner l'échantillon n_n en sortie de cet interpolateur.

Complexité de l'interpolateur, filtres polyphasés

Pratiquement, il est hors de question d'appliquer directement la formule d'interpolation de Shannon, fut-elle tronquée. Une manière classique de mettre en œuvre un filtre interpolateur consiste à utiliser un filtrage polyphasé. L'idée consiste à stocker en mémoire plusieurs échantillonnages décalés d'une fraction de T_e du sinus cardinal puis à sélectionner pour interpoler, le filtre à réponse impulsionnelle finie fournie par la table la plus proche du décalage souhaité. Cette procédure peut être raffinée en interpolant entre les tables. Le plus simple consiste à interpoler linéairement ou par des cubiques. Le lecteur intéressé par ces aspects pourra consulter [8].

5.4 Cas des signaux à spectre étalé

Les techniques présentées jusqu'ici sont basiques et peuvent être aisément adaptées à divers cas particuliers. A titre d'exemple, nous donnons maintenant un exemple de synchroniseur pour les systèmes à étalement de spectre par séquence directe.

Les systèmes de communication pour la téléphonie mobile ont pour la plupart opté pour le partage de la ressource à répartition par codage (CDMA) afin de gérer l'aspect multi-utilisateurs. Parmi les méthodes d'étalement de spectre, l'étalement de spectre par séquence directe (Direct Sequence CDMA, DS-SS) sera utilisé par le système européen UMTS⁵.

Pour un signal DS-SS, l'impulsion de mise en forme comprend un code $c(t)$ qui doit posséder de bonnes propriétés d'autocorrélation (« proche d'un Dirac »). Le code est généralement une séquence binaire composée de M chips ; la vitesse des transitions entre chips étant M fois plus élevée que la vitesse des transitions entre les symboles, le spectre est étalé d'un facteur M . Notons $T_c = T/M$ la durée d'un chip.

Donnons quelques notions de base sur la synchronisation d'un signal DS-SS.

Synchronisation grossière, acquisition.

L'autocorrélation du code étant proche d'une impulsion de Dirac, une simple intercorrélacion entre le signal reçu et le code permet de synchroniser approximativement. Lorsque le canal présente des trajets multiples, il reste possible "d'accrocher" le trajet dominant. En pratique, différentes stratégies, comme la vérification sur plusieurs symboles consécutifs de la position du pic, peuvent être adoptées pour s'assurer du bon accrochage initial.

Synchronisation fine, poursuite.

Les boucles de synchronisation du rythme s'adaptent très simplement aux signaux à spectre étalé. Le signal observé en sortie du filtre adapté à l'ensemble de la chaîne de transmission est toujours de la forme :

⁵ Universal Mobile Telecommunications System.

$$y(t) = \sum_n a_n p(t - nT - \tau) + n(t)$$

et la vraisemblance des observations est proportionnelle à :

$$\exp\left(\sum_k \frac{1}{\sigma^2} \operatorname{Re}(a_k y^*(kT + \tau))\right)$$

Pratiquement, en remplaçant les données inconnues par une décision sur la sortie du filtre adapté (fonction signe en BPSK), la maximisation de la vraisemblance peut être atteinte à l'aide d'une procédure itérative :

$$\tau_{k+1} = \tau_k - \gamma \operatorname{sign}(y(t)) \operatorname{Re}(y(t + \Delta t) - y(t - \Delta t)) \Big|_{kY + \tau_k}$$

Cet algorithme⁶ est appelé une boucle à verrouillage de délai, ou boucle « early-late », du fait qu'elle utilise la corrélation du signal reçu par le code pour trois valeurs du retard $(-\Delta t, 0, +\Delta t)$. La valeur Δt est le plus souvent réglée à la moitié de la durée du chip.

Même si les algorithmes s'écrivent formellement de la même façon pour des signaux avec ou sans étalement, il existe tout de même quelques différences de comportement liées aux propriétés des codes utilisés. Pour certains codes, *e.g.* les séquences binaires de longueur maximale, la fonction de corrélation est constante, et proche de zéro, pour toutes les valeurs du retard, sauf celles comprises entre $-T_c$ et $+T_c$ (voisinage du pic de corrélation). Ainsi, pour qu'une boucle « early-late » fonctionne, il est nécessaire qu'elle soit initialisée dans cette plage, d'où la nécessité de décomposer la synchronisation en une phase grossière consistant à se caler à un chip près (acquisition) et une phase fine durant laquelle un algorithme affine l'estimation du retard puis suit ses éventuelles variations (poursuite). Si, au cours de son fonctionnement, la boucle sort de cette plage d'accrochage, le verrouillage est perdu : d'où la nécessité pratique de surveiller son fonctionnement pour décider, le cas échéant, d'une réacquisition, c'est-à-dire du déclenchement d'une nouvelle procédure d'acquisition. Notons qu'en pratique, la plage de recherche pourra tout de même être plus réduite que pour la première acquisition, mais que, fondamentalement, la démarche reste la même. Le pendant de cette perte d'accrochage pour les boucles de phase classiques est le phénomène de glissement de cycle (changement de détermination de la phase estimée). Les glissements de cycles sont gênants mais ne conduisent pas à une perte de synchronisation de longue durée. *À contrario*, la perte d'accrochage pour une boucle « early-late » la place dans un état dans laquelle elle peut rester très longtemps...

D'autres remèdes à ce problème de perte d'accrochage peuvent être envisagés :

- **Corrélateurs étendus.** La construction d'une mesure de l'erreur non nulle sur une plage plus large est possible en combinant plus de deux sorties du filtre adapté au code, c'est l'idée des corrélateurs étendus. Naturellement, cette extension de la plage d'accrochage se traduit aussi par une augmentation du bruit dans la boucle.

⁶ Notez que cet algorithme est de la forme de l'algorithme de Gardner.

- **Une estimation optimale...** Il est important de noter que l'apparition de la distinction entre acquisition et poursuite ne surgit que lorsque les méthodes de résolution utilisées sont sous optimales. Dans une approche du type filtrage non linéaire, il n'y a pas de raison de distinguer acquisition et poursuite [1].

5.5 Exemples de mise en œuvre en MATLAB

Dans cette section, nous illustrons les idées développées dans ce chapitre à l'aide d'une simulation sous Matlab. Nous considérons l'exemple d'une modulation de phase en quadrature à 4 états (QPSK), et nous nous intéresserons uniquement à la transmission ramenée en bande de base. Dans un premier temps, il s'agira donc de mettre en place les signaux QPSK qui serviront par la suite. Dans un second temps, on s'intéressera à la synchronisation de porteuse, en considérant que le signal obtenu après démodulation est entaché d'un résidu de porteuse. Dans un troisième temps, on considérera le problème de la synchronisation de rythme.

5.5.1 Mise en place des signaux

Il s'agit ici de mettre en place les différents signaux. On s'appuie ici sur des scripts de la *Communication Toolbox* de Matlab.

Puisque l'on simule ici en partie des signaux analogiques, on introduit un facteur de suréchantillonnage *Surech* par rapport à la fréquence symbole. Si on note *F_s* la fréquence d'échantillonnage effective (s pour *sampling*) et *F_d* la fréquence symbole, (d pour *digital*) on aura *Surech*=*F_s*/*F_d*. On aura alors *Surech* points par période symbole.

Les symboles sont mis en forme par un filtre en cosinus ajusté surélevé, pour lequel on choisit un facteur de *roll-off* différent de 0, de sorte à avoir un peu d'excès de bande. La réponse impulsionnelle du cosinus surélevé n'est pas causale, et on doit alors introduire un retard *Delay* afin de la rendre causale. Ce filtre est un filtre de Nyquist, qui résulte de la concaténation de deux filtres en « racine de cosinus surélevé », le premier placé à l'émission et le second à la réception.

Plutôt que de travailler avec deux signaux en phase et quadrature, il est plus simple sous Matlab de travailler directement avec l'enveloppe complexe. La fonction `Symb=randint(S,1,[0,M-1])` permet d'obtenir une séquence de longueur *S*, sur *M* niveaux entre 0 et *M*-1. D'autre part, la modulation en bande de base, sur *M* niveaux, sans suréchantillonnage -- c'est-à-dire simplement le *mapping* des symboles sur des niveaux complexes, est effectuée par `dmodce(Symb,Fd,Fd,'qask',M)`. Enfin, la réponse impulsionnelle du cosinus surélevé est fournie par `rcos=(rcosine(Fd,Fs,'default',RollOff,Delay))`.

LISTING 5.1 - CRÉATION DES OBSERVATIONS

```
% Fichier CreaObs.m
% Date : novembre 1999 - mise a jour avril 2001
% Auteur : jfb

% Variables
%
*****
Surech=8;          % Facteur de surechantillonnage = Fs/Fd
```

```

S=128*8;           % Nombre de symboles
M=4;              % Nombre de niveaux pour M-PSK
Fd=1;             % Frequence symbole (frequence baud)
RollOff=0.4;     % Pour le filtre de mise en forme
Delay=10;        % Retard = longueur de la partie non causale
                  % = etendue de la RI : de -Delay a +Delay en

symbole
Delta=0;         % Retard d'echantillonnage (eventuel)

% Creation des signaux
%
*****
% Symb :          sequence de symboles
% ymodbb :        signal module en bande de base
% ymodbb_s        signal module en bande de base, surechantillonne
% ymodbb_sf       signal module en bande de base, surechant. et
filtre
% ymodbb_sf       signal module en bande de base, surechantillonne,
                  filtre,
%
                  et complete par des zeros

Fs=Surech*Fd;    % Frequence d'echantillonnage
                  % Creation de la sequence de symboles
Symb=randint(S,1,[0,M-1]); % sur M niveaux entre 0 et M-1
                  % Modulation QPSK, en bande de base,
ymodbb=dmodce(Symb,Fd,Fd,'qask',M); % sans surechantillonnage

% ---> surechantillonnage :
ymodbb_s=zeros(1,Surech*S);
ymodbb_s(1:Surech:Surech*S)=ymodbb(1:S);

                  % Creation du cosinus sureleve
rcos=(rcosine(Fd,Fs,'default',RollOff,Delay));
ymodbb_sf=conv(rcos,ymodbb_s);

ymodbb_sc=[zeros(1,Delay*Surech) (ymodbb_s) zeros(1,Delay*Surech)];

                  % signal en bb surechant
                  % et complete de zeros pour recalage a cause
                  % de la convolution
Lsf=length(ymodbb_sf); % longueur de la seq surech et filtree

```

La *figure 5.3* représente la partie réelle du signal modulé, en bande de base, avant et après filtrage par le filtre en cosinus surélevé. En début de séquence, on note ainsi un « transitoire », lié au retard introduit par le filtre. On note également que le filtre est bien un filtre de Nyquist, puisque l'on retrouve la valeur des symboles en échantillonnant la sortie du filtre de Nyquist.

figure 5.3 - Parties réelles du signal bande de base avant et après filtrage de Nyquist.

5.5.2 Récupération de porteuse

Le signal en bande de base est contaminé par un résidu de porteuse caractérisée à la fois par un déphasage ϕ_0 et une phase linéaire. On choisit ici pour pente de la déviation linéaire un *Deltaf* de 10^{-3} (en normalisé par rapport à la fréquence d'échantillonnage F_d). Rappelons que la dérive linéaire correspondrait au cas où l'on a un écart de fréquence entre les oscillateurs de l'émetteur et du récepteur. Le terme ϕ_0 ne correspond qu'à un décalage des deux oscillateurs ou à un déphasage introduit par les

différents filtres. Pour inclure ce résidu de porteuse dans le signal recueilli au niveau du récepteur, il suffit de multiplier celui-ci par une exponentielle complexe :

```
smod=ymodbb_sf.*exp(j*2*pi*Deltaf*[1:Lsf] + j*Phi0) ;
```

Il s'agit maintenant d'éliminer ce résidu du signal, ce qui revient à utiliser un algorithme qui estime les paramètres du déphasage. On peut alors mettre en place une boucle de Costas. La boucle de Costas d'ordre 1 permet de corriger un déphasage constant et de « suivre » une dérive linéaire, tandis qu'à l'ordre 2, la boucle de Costas est capable de corriger la dérive linéaire.

Rappelons que le principe fondamental de ces deux boucles est d'utiliser un algorithme de gradient stochastique estimant le résidu de porteuse ou de phase au fur et à mesure que le récepteur reçoit des échantillons. Pour cela on se définit une erreur d'estimation donnant l'écart le symbole reçu et le symbole théorique, tout en annulant l'effet de la modulation.

Ici, les symboles sont émis en $\pi/4$ QPSK, c'est-à-dire que les symboles a_k sont de la forme $\exp(jm\pi/4)$, avec $m=1..4$. La constellation est ainsi déphasée de $\pi/4$. On a alors $(a_k)^4=-1$, pour tout k , et l'effet de la modulation est ainsi gommé. Ainsi, si $y(k)$ est le signal reçu entaché d'une phase ϕ_k , l'erreur :

$$e(k) = \left(y(k) e^{-i\pi/4} e^{-i\phi_k} \right)^4 - 1$$

s'annule lorsque $\phi_k = \phi_k^*$. En minimisant alors l'erreur quadratique moyenne, à l'aide d'un algorithme de gradient stochastique, on obtient une récurrence de la forme⁷ :

$$\phi_k = \phi_{k-1} - \mu \operatorname{Im} \left(y(k)^4 e^{-i4\phi_{k-1}} \right)$$

Cette boucle est très simple de mise en œuvre, comme l'indique le listing 5.2 :

LISTING 5.2 - BOUCLE DE COSTAS D'ORDRE 1

```
% Fichier Costas.m
% Date : novembre 1999 - mise a jour avril 2001
% Auteur : jfb

% Creation du signal avec residu de porteuse et dephasage
% *****

Lsf=length(ymodbb_sf);          % longueur de la seq surech et filtree
Peigne = zeros(1,Lsf);         % creation d'un peigne de Dirac
Peigne(1+Delta:Surech:Lsf)=ones(1,length([1+Delta:Surech:Lsf]));

Phi0=0.1;                       % Initialisation du residu
Deltaf=0.001;

                                % Signal avec residu de phase
smod=ymodbb_sf.*exp(j*2*pi*Deltaf*[1:Lsf] + j*Phi0);
smod_e=smod.*Peigne;           % echantillonne a la periode symbole
smod_es=smod(1+Delta:Surech:Lsf);

phiopt=((2*pi*Deltaf*Surech)*[1:S] + Phi0 );
```

⁷ Le changement de signe du pas d'adaptation par rapport à l'expression fournie en 5.2.2 est liée au décalage de $\pi/4$ de la constellation. Si on n'en tenait pas compte a priori, la phase identifiée serait biaisée de $\pi/4$.

```

% Mise en oeuvre de la boucle de Costas d'ordre 1
%
*****
%
% phik  : phase courante
% phik1 : phase precedente
%
%
Phi=[]; phik=0; phik1=0.2;
for k=1:S
    yk=smod_es(k);
    Erreur(k)=abs(-yk^M*exp(-j*M*phik1)-4)^2;
    phik=phik1 - gamma*imag(yk^M*exp(-j*M*phik1));
    Phi=[Phi phik];
    phik1=phik;
end;

```

En ce qui concerne le choix du pas d'adaptation, voir la *figure 5.4*, on retrouve que plus il est important, plus la convergence vers la valeur du résidu est rapide mais oscille autour de la valeur optimale. Pour diminuer ces sur-oscillations, on peut diminuer la valeur du pas. On aura alors un algorithme plus lent mais moins sensible aux rapides variations de phase (cas de sauts de phases importants ou présence de bruit). Si le pas est trop faible, l'algorithme n'arrive plus à estimer le résidu car le terme correctif n'est pas assez influent pour suivre la déviation temporelle de la phase.

Enfin, la *figure 5.5* confirme l'existence d'un biais d'estimation : la phase estimée est toujours en retard par rapport à la phase linéaire.

Figure 5.4 - Évolution de la phase identifiée par une boucle de Costas d'ordre 1, pour différentes valeurs de μ .

Figure 5.5 - Différence entre la vraie phase et la phase identifiée par une boucle de Costas d'ordre 1, pour différentes valeurs de μ .

Pour compenser ce biais, il faudrait « prédire » l'évolution de la phase, comme nous l'avons présenté en 5.2.2 -- *Adaptation à une dérive de phase*. C'est précisément ce que réalise la boucle de Costas d'ordre 2, qui comporte un filtrage de l'erreur, à la manière d'un correcteur PI, bien connu des automaticiens. L'estimation de la phase répond alors à la relation symbolique :

$$\phi_k = \phi_{k-1} - \left(\gamma_1 + \frac{\gamma_2}{1-z^{-1}} \right) \text{Im} \left(y(k)^4 e^{-i4\phi_{k-1}} \right)$$

L'implantation de cette boucle, donnée sur le *listing 5.3* est également fort simple :

LISTING 5.3 - BOUCLE DE COSTAS D'ORDRE 2

```

% Boucle de Costas
Phi=[]; phik=0; phik1=0;      % Initialisations des différentes
Epsilon=[]; epsilonk1=0;     % variables utilisees

deb=1;
for k=deb:S
    yk=smod_es(k);
    Erreur(k)=abs(-yk^M*exp(-j*M*(phik1+epsilonk1))-sqrt(2)^M)^2;
    gradient= imag(yk^M*exp(-j*M*(phik1+epsilonk1)));
    phik=phik1 + epsilonk1 - gamma1*gradient;
    epsilonk = epsilonk1 - gamma2*gradient;
    phik1=phik;
    epsilonk1 = epsilonk;
    Phi(k)=phik;              % Sauvegarde des resultats
    Epsilon(k)=epsilonk;
end;

```

Sur la *figure 5.6*, nous donnons l'erreur entre la phase théorique et la phase identifiée par cette boucle, pour quelques valeurs des paramètres. On remarque alors la disparition du biais d'estimation, mais aussi des oscillations caractéristiques d'un système du second ordre lors de la convergence.

Figure 5.6 - Écart entre la phase théorique et la phase estimée par la boucle de Costas d'ordre 2, pour différentes valeurs des paramètres.

5.5.3 Récupération de rythme

Il s'agit maintenant de prendre en compte une méconnaissance du rythme. Cette méconnaissance peut être liée à un retard entre les horloges de l'émetteur et du récepteur ou au fait que la période symbole utilisée au niveau du récepteur est différente de celle de l'émetteur. Dans les deux cas, ceci se traduit par un mauvais échantillonnage des signaux reçus, qui ne permet plus de retrouver les symboles de départ. On introduit alors un retard τ_k à la réception, de manière à compenser la méconnaissance du rythme. Si T_s' est la période symbole utilisée en réception, et Δt le décalage entre les horloges, alors il faut que $y(kT_s' - \tau_k) = a_k$ pour que l'échantillonnage soit correct, soit $\tau_k = k(T_s - T_s') - \Delta t$. En cas d'erreur sur la période symbole, le retard τ_k devra ainsi être une pente linéaire.

On utilise ici l'approche adaptative présentée en 5.3.1 : on cherche à minimiser l'erreur quadratique moyenne $J(\tau_k) = E[|y(kT_s' - \tau_k) - a_k|^2]$. Le gradient instantané de cette fonction de coût s'écrit :

$$\frac{\partial J(kT_s' - \tau_k)}{\partial \tau_k} = \text{Re} \left\{ \left(y(kT_s' - \tau_k) - a_k \right)^* \frac{\partial y(t)}{\partial t} \Bigg|_{t=kT_s' - \tau_k} \right\}$$

Ce gradient est utilisé pour estimer de manière itérative la valeur de τ_k . Dans la mesure où l'on peut s'attendre à une variation linéaire, on choisit de construire une boucle du second ordre à la manière de la boucle de Costas d'ordre 2. Le problème est ici compliqué par le fait que le signal n'est connu qu'aux instants d'échantillonnage, et non en tout instant ($kT_s' - \tau_k$), et qu'il faut également accéder à la dérivée du signal. On utilise ici une fonction d'interpolation quadratique, `interp3.m`, fournie en annexe à ce chapitre, qui rend la valeur interpolée et le gradient correspondant (on pourrait également utiliser une interpolation et un calcul de dérivée par différences finies). D'autre part, la séquence de symboles a_k n'est pas connue nécessairement, et l'on remplacera alors les a_k par la

décision sur les symboles interpolés. Muni de ces remarques, on peut implanter un algorithme du second ordre répondant à la relation symbolique :

$$\tau_k = \tau_{k-1} - \left(\gamma_1 + \frac{\gamma_2}{1 - z^{-1}} \right) \operatorname{Re} \left\{ \left[y(kT_s' - \tau_k) - \operatorname{dec}(y(kT_s' - \tau_k)) \right] \frac{\partial y(t)}{\partial t} \Big|_{t=kT_s' - \tau_k} \right\}$$

Le *listing 5.4* fournit l'implantation Matlab de cette boucle de récupération de rythme.

LISTING 5.4 - RÉCUPÉRATION DE RYTHME

```
% Fichier RecupRythme.m
% Date : novembre 1999 - mise a jour avril 2001
% Auteur : jfb

% Creation du signal
% *****
Deltaf=0.0;
Phi0=0;
smod=ymodbb_sf.*exp(j*2*pi*Deltaf*[1:Lsf] + j*Phi0);
y=smod;

%***** Recuperation de rythme *****
% Avec decision dans la boucle et derive lineaire
% (ie on n'a pas passe le bon temps symbole)
% On travaille avec Surech echantillons par periode symbole
% *****

S=300;           % Nombre de symboles
Ts=7.2;         % Longueur du temps symbole
Deltat=0;       % Erreur sur la synchro

           % initialisations des variables
tauk1=Deltat/Surech; tauk=0;
Tau=[]; Erreur=[]; Y=[];
epsilonk1=0;

% Boucle d'asservissement sur le rythme

for k=21:S      % Boucle sur les symboles
    % Instant courant = (k-1)*Ts + tauk1 + 1
    % qu'on divise en une partie entiere et un reste

    tps_courant=fix((k-1)*Ts + tauk1 + 1); % partie entiere
    Dt=(k-1)*Ts + tauk1 + 1 - tps_courant % reste

    % On interpole ensuite la valeur de y((k-1)*Ts + tauk1 + 1)
    % et de son gradient

    [y_inter,y_delta,y_grad]=interp3(y(tps_courant-
1:tps_courant+1),Dt);

    % Calcul des decisions...
    deck=sign(real(y_delta)) + j*sign(imag(y_delta));

    % et de l'erreur et de son gradient
    erreur=abs(y_delta-deck)^2;
    grad=real(conj(y_delta-deck)*y_grad);
```

```

% Mise a jour de tau(k)
tauk=tauk1 + epsilonk1 - gamma1*grad;
epsilonk = epsilonk1 - gamma2*grad;

epsilonk1 = epsilonk;
tauk1=tauk;

% Sauvegarde des resultats
Erreur=[Erreur erreur];
Y=[Y y_delta];
Tau=[Tau tauk];
end;

```

Les *figures 5.7 et 5.8* illustrent le comportement de l'algorithme dans le cas d'une erreur sur la période symbole : alors que la période symbole à l'émetteur était ici de 8 périodes d'échantillonnage (8 échantillons par période symbole), on a supposé que la période était de 7.2 périodes d'échantillonnage. Sans correction, il s'en suit une désynchronisation complète du récepteur. La boucle permet ici de corriger l'erreur en moins de 50 symboles. Sur la *figure 5.9*, on a donné les résultats obtenus pour un retard entre les deux horloges : on a utilisé ici un décalage de 3 périodes d'échantillonnage, soit pratiquement la moitié d'un symbole.

Figure 5.7 - Récupération de rythme pour une erreur sur la période symbole : différence entre l'instant d'échantillonnage optimal et l'instant calculé, au cours des itérations, pour différentes valeurs des paramètres.

Figure 5.8 - Récupération de rythme pour une erreur sur la période symbole : erreur quadratique entre le signal interpolé et la décision, au cours des itérations, pour différentes valeurs des paramètres.

Figure 5.9 - Récupération de rythme pour un retard d'échantillonnage : différence entre l'instant d'échantillonnage optimal et l'instant calculé, pour différentes valeurs des paramètres.

On a présenté dans cette section quelques exemples d'implantation et des résultats de simulation d'algorithmes de synchronisation. Il s'agit d'une présentation forcément parcellaire et incomplète. En effet, les algorithmes ont été considérés comme des blocs indépendants du récepteur. En réalité, dans la conception d'un récepteur, on devra faire face *à la fois* à des problèmes de synchronisation de rythme, de phase, mais également aux problèmes d'égalisation. Il faudra donc *coupler* les différents algorithmes d'estimation et de récupération.

LISTING 5.5 - FONCTION D'INTERPOLATION QUADRATIQUE

```

function [y_inter,y_delta,y_grad]=interp3(y,delta);
% SYNTAX: fonction [y_inter,y_delta,y_grad]=interp3(y,delta);
%
% DESCRIPTION: Interpolates between three points using a quadratic
model.
% Return the interpolated value y_delta, and also the gradient
% y_grad at point delta with delta in [-1,1].

```

```

% Nota: the "distance" between two points is considered as 1,
% and the points are labelled {-1,0,1}.
%
% Author: jfb - Nov. 1999

L=100;
y=y(:);
if length(y)~=3
    error('input y must be of length 3');
end;
yr=real(y);
yi=imag(y);

%y=ax^2 + bx +c
x=[1:4*L]/L - 2; % x=[1:2*L]/L - 1;

A=[1 -1 1; 0 0 1; 1 1 1]; % for x={-1 0 1}
% y=A*coeffs;
coeffsr=inv(A)*yr; % coeffs=[a b c]'
coeffsi=inv(A)*yi; % coeffs=[a b c]'

yr_inter=coeffsr(1)*x.^2 + coeffsr(2)*x + coeffsr(3);
yi_inter=coeffsi(1)*x.^2 + coeffsi(2)*x + coeffsi(3);
y_inter=yr_inter+j*yi_inter;

yr_delta=yr_inter(2*L+round(L*delta));
yi_delta=yi_inter(2*L+round(L*delta));

y_delta=yr_delta+j*yi_delta;

% Gradient:

yr_grad = 2*coeffsr(1)*delta + coeffsr(2);
yi_grad = 2*coeffsi(1)*delta + coeffsi(2);
y_grad = yr_grad + j*yi_grad;

```

5.6 Conclusion

Ce chapitre a présenté quelques notions de base concernant la synchronisation⁷. Les études d'algorithmes efficaces, cruciaux pour le bon fonctionnement d'un système de transmission, restent largement ouvertes, en particulier pour des schémas de modulation récents tels que les multiporteuses, les mélanges OFDM/CDMA ou les modulations non linéaires de type CPM.

Références bibliographiques

[1] P.O. Amblard, J.M. Brossier. *Phase Tracking : what do we gain from optimality ? Particle filtering vs PLL*. Soumis à Signal Processing.

⁷ Remerciements à Pierre-Olivier Amblard et Thierry Lenez pour leurs contributions.

[2] A. Benveniste, M. Métivier, P. Priouret. *Adaptive algorithms and stochastic approximations*, Springer Verlag, Applications of Mathematics, vol. 22, Berlin, Heidelberg, New York, 1990.

[3] H.J. Kushner, J. Yang. *Analysis of Adaptive Step-Size SA Algorithms for Parameter Tracking* IEEE transaction on Automatic Control, vol. 40, N°8, August 1995.

[4] Heinrich Meyr, G. Ascheid. *Synchronization in Digital communication*, Wiley 1989.

[5] Heinrich Meyr, Marc Moeneclaey, Stefan A. Fechtel. *Digital communication receivers. Synchronization, channel estimation and signal processing*, Wiley 1998.

[6] M. Oerder, H. Meyr. *Digital Filter and Square Timing Recovery*, IEEE transaction on Communications, vol. 36, n°5, May 1988.

[7] Jaune Riba, Gregori Vazquez. *Non-Data-Aided Frequency Offset and Symbol Timing Estimation for Binary CPM : Performance Bounds*, ICASSP 2000.

[8] P.P. Vaidyanathan. "Multirate Systems and Filterbanks" Prentice Hall 1993.