

# Chapitre 4

## Égalisation

### 4.1 Introduction

En principe, si le canal est parfaitement connu, il est possible de rendre l'interférence entre symboles arbitrairement faible, ou même de l'éliminer complètement, en utilisant une paire de filtres d'émission et de réception tels que

$$\sum_n G(f + n/T_s)C(f + n/T_s)H(f + n/T_s) = T_s,$$

c'est-à-dire que la chaîne complète vérifie le critère de Nyquist (le filtre global  $G(f)C(f)H(f)$  est un filtre de Nyquist). En pratique, cependant, on ne connaît que très rarement les caractéristiques exactes du canal, et tout au plus des valeurs moyennes. Par ailleurs, il subsiste des erreurs dans la correction de l'interférence entre symboles, en raison des imperfections sur l'implantation des filtres  $g$  et  $h$ . Enfin, le canal peut ne pas être stationnaire, c'est-à-dire que ses caractéristiques varient au cours du temps. L'effet de ces différents facteurs est une interférence entre symboles (éventuellement résiduelle), qu'il faut compenser, à l'aide d'un dispositif appelé *égaliseur*, ce dispositif pouvant également être variable dans le temps.

En bande de base, si les filtres d'émission et de réception sont fixés (et forment ensemble un filtre de Nyquist), le rôle de l'égaliseur est simplement de compenser la réponse du canal. En notant ainsi  $e(t)$  la réponse impulsionnelle de l'égaliseur et  $E(f)$  sa fonction de transfert, il faut obtenir

$$E(f) = \frac{1}{C(f)}, \text{ sur } [-B, B].$$

Comme la paire des filtres d'émission et de réception forment un filtre de Nyquist, on peut parfaitement reconstituer la séquence d'entrée  $\{a_n\}$ . Du point de vue spectral, on a « égalisé » la réponse du canal.

Afin d'illustrer les performances et la mise en œuvre de certains des algorithmes que nous présenterons ci-après, nous utiliserons les trois canaux test de Proakis [1]. Les fonctions de transfert sont représentées sur la *figure 4.1*. Le canal (a) est assez facile, le canal (b) modérément difficile et le canal (c) très difficile à égaliser. Les réponses impulsionnelles sont fournies sur la table 4.1. Notons ici que les performances que nous obtiendrons ne comprennent que l'aspect égalisation, et qu'il est possible d'améliorer notablement ces taux d'erreurs en utilisant un codage canal.

Figure 4.1 – Fonctions de transfert des trois canaux test de Proakis.

Canal (a)	0.04	-0.05	0.07	-0.21	-0.5	0.72	0.36	0	0.21	0.03	0.07
Canal (b)	0.407	0.815	0.407								
Canal (c)	0.227	0.46	0.688	0.460	0.227						

TABLEAU 4.1 – RÉPONSES IMPULSIONNELLES DES TROIS CANAUX DE PROAKIS.

### 4.2 Les structures classiques de l'égalisation

Du point de vue des hypothèses, on supposera dans ce qui suit, que :

- la séquence des symboles  $a_n$  est stationnaire, centrée et blanche,
- le bruit d'observation  $b(n)$  est centré, décorréolé (ou indépendant) des symboles  $a_n$ , et blanche,
- la réponse du canal  $c(t)$  est causale.

## 4.2.1 L'égaliseur transverse

Les égaliseurs transverses sont les plus simples à mettre en œuvre. En effet, il s'agit simplement d'utiliser un filtre numérique à réponse impulsionnelle finie, pour lesquels les méthodes de calcul et d'implantation sont bien connues. La structure du filtre est donnée sur la *figure 4.2* et correspond à la relation entrée-sortie

$$z(n) = \sum_{i=0}^{M-1} e(i)y(n-i),$$

où  $e(n)$  est la réponse impulsionnelle de l'égaliseur, de longueur  $M$ ,  $y(n)$  est la séquence d'observations, et  $z(n)$  la sortie de l'égaliseur.

Figure 4.2 – Structure d'un égaliseur transverse

### L'égaliseur par « zero forcing »

Le *zero forcing* est un filtre qui tente d'inverser exactement la fonction de transfert du canal, ce qui est *a priori* précisément le but recherché, idéalement, par l'égalisation. Ce faisant, l'interférence entre symboles est exactement compensée (pourvu que l'ensemble des filtres d'émission et de réception soit Nyquist), et l'on dit que l'interférence entre symboles est forcée à zéro. On a ainsi

$$E(z) = \frac{1}{C(z)}.$$

Dès maintenant, on peut s'apercevoir que cette démarche souffre de deux défauts : d'abord,  $C(z)$  peut posséder des zéros de module supérieur à 1, ce qui induit des pôles instables pour  $E(z)$ , si celui-ci doit être causal ; d'autre part, si  $c(n)$  est une réponse impulsionnelle finie, alors  $e(n)$  est à réponse impulsionnelle infinie. On peut tourner partiellement la première difficulté en introduisant un retard  $R$  lors de la résolution ce qui permet de prendre en compte une éventuelle partie non causale (mais retardée), et tient également compte du nécessaire retard lié à la mise en œuvre des filtres. Le choix de ce retard est à la fois important et difficile. Globalement, on peut dire que le retard engendré par les deux filtres est égal à la moitié de la longueur du filtre équivalent, longueur qui vaut ici  $N + M - 1$ . D'autre part, on choisit de prendre un ordre  $M$  assez grand pour que l'éventuelle erreur de troncature de la réponse impulsionnelle soit peu sensible. Dans ces conditions, on peut écrire l'équation de convolution correspondant à l'inversion du canal

$$\sum_{i=0}^{M-1} e(i)c(n-i) = \delta(n-R),$$

où  $\delta$  est une impulsion de Dirac à temps discret,  $\delta(n-R) = 1$  pour  $n = R$  et  $\delta(n-R) = 0$  sinon. Cette relation de convolution étant valable pour tout  $n$ , on peut se donner  $M$  équations, par exemple pour  $n = 0..M-1$ , et résoudre exactement le système linéaire correspondant pour obtenir les coefficients de l'égaliseur  $e(i)$ ,  $i = 0..M-1$ . À la sortie de l'égaliseur, on obtient alors

$$z(n) = a_{n-R} + b'(n),$$

où  $b'(n)$  est le bruit d'observation filtré par l'égaliseur. L'annulation des interférences entre symboles se fait généralement au prix d'une augmentation sensible du niveau de bruit. En effet, la fonction de transfert du canal est en général de type passe-bas, et son inverse est de type passe haut. Lorsque le bruit est large bande, il s'en suit une forte augmentation du bruit en haute fréquence et une dégradation du rapport signal-à-bruit. Ainsi, en dehors du cas où l'on est assuré d'un faible niveau de bruit d'observation, cette solution n'est pas à retenir. On notera en outre que le canal est supposé parfaitement

connu ; dans cette méthode *supervisée*, on devra alors passer par une estimation préalable de la réponse impulsionnelle du canal. Les erreurs d'estimation de la réponse impulsionnelle se répercuteront alors sur les coefficients de l'égaliseur et entraîneront une dégradation des performances.

La *figure 4.3* fournit les performances de l'égaliseur *zero forcing* pour les trois canaux de Proakis, en fonction du rapport signal-à-bruit (RSB). Ce dernier est défini ici, comme dans toute la suite de ce chapitre, comme

$$RSB = \frac{P_a \sum c(n)^2}{P_b},$$

où  $P_a$  et  $P_b$  sont respectivement la puissance de la séquence de symboles et la puissance du bruit additif<sup>1</sup>. On constate que si les performances pour le canal (a) sont bonnes, les canaux (b) et (c) montrent un taux d'erreur supérieur à 10% jusqu'à un RSB très élevé. Ceci est lié à la présence des évanouissements en fréquence (voir la *figure 4.1*), dont l'inverse produit une amplification du bruit.

Figure 4.3 – Performances de l'égaliseur à zero forcing pour les trois canaux test.

Les performances ont été obtenues à l'aide du script MATLAB du LISTING 4-1, qui prépare les données, calcule l'égaliseur *zero forcing* par appel à une fonction externe `ffe_zf`, voir le LISTING 4-4, section 4.4, puis évalue les performances en fonction du rapport signal-à-bruit. Ce premier script n'est donné ici que comme exemple d'emploi d'une fonction spécialisée, et nous ne décrirons plus ensuite que les fonctions MATLAB de calcul des égaliseurs.

LISTING 4-1 : SCRIPT MATLAB POUR L'ÉVALUATION DES PERFORMANCES DU *zero forcing*

```
N=length(c);           % N : longueur du canal
R=(N+M)/2-1;          % R : choix du retard
L=length(y);          % L : longueur de la séquence d'observation
RSB=[5 7 9 12 15 ...  % Niveaux de RSB pour l'étude (en dB)
     20 25 30 35 40];

% Calcul de l'égaliseur à zero forcing
e_zf=ffe_zf(y,sym,M,c,nom_canal);

% Evaluation des performances
for i=1:length(RSB)
    yb=y+10^(-RSB(i)/20)*randn(1,L); % Signal bruité avec le RSB=RSB(i)
    z=filter(e_zf,1,yb);             % Calcul de la sortie de l'égaliseur
    dec=sign(z(1+R:L));              % Puis des décisions,
    err=sym(1:L-R)-dec;              % de l'erreur,
    BER(i)=length(find(err))/L;      % et du taux d'erreur
end
```

## Égaliseur à erreur quadratique minimale

Alors que l'égaliseur à *zero forcing* résoud le problème en faisant abstraction du bruit d'observation, l'idée de l'égaliseur à erreur quadratique minimale (égaliseur EQM) est de minimiser l'erreur quadratique entre la séquence d'entrée (symboles) et la sortie de l'égaliseur. Le bruit est ainsi pris en compte dans le critère. On cherche ainsi à minimiser l'erreur quadratique moyenne

$$\Xi(\mathbf{e}) = E \left[ |z(n) - a_{n-R}|^2 \right].$$

<sup>1</sup>À toutes fins utiles, on notera que dans [1], le bruit est pris de variance  $N_0/2$ , mais que seul un facteur  $N_0$  intervient dans sa définition du RSB, ce qui entraîne un écart de 3dB sur les performances.

Avec  $z(n) = \sum_{i=0}^{M-1} e(i)y(n-i) = \mathbf{e}^T \mathbf{y}(n)$ , où  $\mathbf{e}^T = [e(0) \dots e(M-1)]$  et  $\mathbf{y}(n)^T = [y(n) \dots y(n-M+1)]$ , on a

$$\Xi(\mathbf{e}) = \text{E} \left[ |\mathbf{e}^T \mathbf{y}(n) - a_{n-R}|^2 \right].$$

Ainsi, la minimisation de l'erreur quadratique moyenne est obtenue pour

$$\frac{\partial \Xi(\mathbf{e})}{\partial \mathbf{e}} = 2\text{E} \left[ \mathbf{y}(n) \left( \mathbf{e}^T \mathbf{y}(n) - a_{n-R} \right) \right] = \mathbf{0},$$

soit

$$\text{E} \left[ \mathbf{y}(n) \mathbf{y}(n)^T \right] \mathbf{e} = \text{E} \left[ \mathbf{y}(n) a_{n-R} \right]$$

On tire de cela

$$\mathbf{R}_{yy} \mathbf{e} = \mathbf{R}_{ya}(R),$$

et

$$\mathbf{e} = \mathbf{R}_{yy}^{-1} \mathbf{R}_{ya}(R), \quad (1)$$

où  $\mathbf{R}_{yy}$  est la matrice de corrélation de  $\mathbf{y}(n)$  et  $\mathbf{R}_{ya}(R)$  est le vecteur d'intercorrélation entre  $\mathbf{y}(n)$  et  $a(n-R)$ . Le nombre de coefficients du filtre est nécessairement limité. Si l'égalisation obtenue est clairement de meilleure qualité que celle fournie par un *zero forcing*, en raison de la prise en compte effective du bruit, elle reste souvent de qualité médiocre, en particulier en présence d'évanouissements sélectifs (non stationnarités). Ceci est également lié à la structure transverse (pas de pôles) qui limite la capacité de représentation d'une réponse quelconque. Par ailleurs, pour la mise en œuvre pratique, il est nécessaire de connaître  $a(n-R)$ . Pour ce faire, on utilise une séquence connue du récepteur, une *séquence d'apprentissage*, pour calculer les coefficients du filtre, voir *figure 4.4*. La nécessité d'inclure dans l'émission une séquence d'apprentissage, éventuellement répétée périodiquement si le système est non stationnaire, limite en outre le débit en données utiles.

Figure 4.4 – Structure d'égalisation avec période d'apprentissage

La *figure 4.5* fournit les performances de l'égaliseur minimisant l'erreur quadratique moyenne, pour les trois canaux. Par rapport à l'égaliseur *zero forcing*, on constate que la prise en compte du bruit améliore sensiblement les performances, et permet d'égaliser les canaux présentant des évanouissements en fréquence. Le programme MATLAB correspondant figure au LISTING 4-5.

Figure 4.5 – Performances de l'égaliseur minimisant l'erreur quadratique moyenne.

## Solution d'égalisation à maximum de vraisemblance : algorithme de Viterbi

La présence d'interférences entre symboles, liée à un canal imparfait, est caractéristique d'une *mémoire* dans le signal. Comme dans nous l'avons vu dans le chapitre sur le récepteur optimal, on peut alors chercher à reconstituer *la séquence de symboles* au sens du maximum de vraisemblance, en exploitant l'interdépendance des données reçues, et en maximisant la vraisemblance à l'aide de l'algorithme de Viterbi. Rappelons que celui-ci permet de sélectionner dans un treillis le chemin de métrique la plus faible, où la métrique est une mesure cumulée, mise à jour de noeud en noeud. Ceci signifie en fait que la densité de probabilité conjointe des observations pour une séquence particulière (la vraisemblance), peut s'écrire comme le produit des vraisemblances élémentaires à chaque instant. Il est important de noter alors que ceci n'est vrai que si *le bruit d'observation, après échantillonnage, est blanc*.

Dès lors, si le bruit d'observation est blanc, on pourra utiliser l'algorithme de Viterbi sur le signal reçu échantillonné au rythme symbole. Cependant, le bruit d'observation n'est pas nécessairement

blanc d'une part, et même si tel était le cas, il perdrait cette propriété en sortie du filtre de réception. Notons  $S_{bb}(f)$  la densité spectrale de puissance du bruit ; on peut toujours factoriser  $S_{bb}(f)$  comme  $S_{bb}(f) = S_b(f)S_b^*(f)$ , où  $S_b(f)$  est une fonction de transfert d'inverse causal et stable (à phase minimale), et l'on peut *blanchir* le bruit d'observation en plaçant en tête de récepteur un filtre blanchisseur de fonction de transfert  $1/S_b(f)$ . La partie utile du signal étant également filtrée par ce filtre blanchisseur, on choisira alors comme filtre de réception un *filtre adapté* (qui est optimal, rappelons le, si le bruit en entrée est blanc) à la cascade filtre d'émission, canal, filtre blanchisseur, c'est-à-dire avec une fonction de transfert  $(G(f)C(f)/S_b(f))^*$ , à un déphasage près. En sortie du filtre de réception, le bruit est à nouveau corrélé, avec une densité spectrale de puissance de la forme  $\sigma_b^2 |G(f)C(f)|^2 / S_{bb}(f)$ . Après échantillonnage au rythme symbole, en utilisant le fait que  $G(f)H(f)$  est un filtre de Nyquist, on obtient le modèle discret équivalent

$$y(n) = \sum_i r(i)a_{n-i} + b'(n),$$

où  $r$  est la réponse impulsionnelle du canal équivalent à temps discret, et qui vaut ici  $r(n) = \int_{-\infty}^{\infty} c'(t)c'^*(t - nT)dt$  (avec  $c'$  la réponse impulsionnelle associée à  $C(f)/S_b(f)$ ), et où  $b'(n)$  a pour densité spectrale de puissance  $S_{b'b'}(z) = \sigma_b^2 C(z)C^*(1/z^*)/S_{bb}(z)$ . À nouveau, il est possible de blanchir le bruit en factorisant  $S_{b'b'}(z)$  sous la forme  $S_{b'b'}(z) = S_{b'}(z)S_{b'}^*(1/z^*)$ , et en filtrant  $y(n)$  par  $1/S_{b'}(z)$ , pour obtenir finalement le modèle discret à bruit blanc additif

$$y(n) = \sum_{i=0}^{L-1} v(i)a_{n-i} + w(n).$$

Notons que lorsque le bruit additif sur le canal est blanc, on trouve pour  $V(z)$  le facteur d'inverse causal et stable d'une factorisation de  $R(z)$ . Même si les différentes opérations énoncées ci-dessus ne sont pas de mise en œuvre aisée, il nous a semblé opportun de nous attarder un peu sur ces différents traitements, souvent éludés, dont le but est de se ramener au modèle à bruit blanc additif, ce modèle garantissant l'optimalité des procédés de réception, et en particulier de la mise en œuvre de l'algorithme de Viterbi qui nous occupe ici.

La réponse impulsionnelle est ici de longueur  $L$ . La mémoire du signal vaut alors  $L$ . Ainsi,  $y(n)$  dépend des symboles  $a_{n-L+1}, a_{n-L+2}, \dots, a_n$  et  $y(n+1)$  dépendra des symboles  $a_{n-L+2}, a_{n-L+3}, \dots, a_{n+1}$ . Ces deux séquences de symboles contiennent  $L - 1$  symboles communs, et il n'y a que  $M$  possibilités pour passer de la première séquence à la seconde, si  $a_{n+1}$  prend ses valeurs dans un alphabet de cardinal  $M$ . On peut donc bâtir un treillis associé aux  $M^L$  états de  $L$  symboles, et chaque noeud du treillis comporte  $M$  chemins entrants et  $M$  chemins sortants.

Si on considère maintenant une séquence de symboles de longueur  $K$ , et que l'on note  $\mathbf{s}^{(n)}$  la  $n^e$  parmi les  $M^K$  séquences possibles, alors, *sous l'hypothèse d'un bruit additif blanc gaussien*, la densité de probabilité conjointe de la séquence d'observation  $y_1, y_2, \dots, y_K$  si la séquence  $\mathbf{s}^{(n)}$  a été émise, est

$$\begin{aligned} p(y_1, y_2, \dots, y_K | \mathbf{s}^{(n)}) &= \prod_{k=1}^K \frac{1}{\sqrt{2\pi\sigma_b^2}} \exp \left\{ -\frac{(y_k - s_k^{(n)})^2}{2\sigma_b^2} \right\} \\ &= \left( \frac{1}{\sqrt{2\pi\sigma_b^2}} \right)^K \exp \left\{ -\frac{1}{2\sigma_b^2} \sum_{k=1}^K (y_k - s_k^{(n)})^2 \right\}, \end{aligned}$$

où  $s_k^{(n)} = \sum_{i=0}^{L-1} v(i)\mathbf{s}^{(n)}(k-i)$  est la sortie à l'instant  $k$  qui correspondrait à l'entrée  $\mathbf{s}^{(n)}$ . on peut alors rechercher la séquence  $\mathbf{s}^{(n)}$  qui maximise la vraisemblance, ou qui minimise la métrique cumulée

$$D(K) = \sum_{k=1}^K (y_k - s_k^{(n)})^2.$$

À tout instant  $i$ , on observe que  $D(i) = D(i-1) + (y_i - s_i^{(n)})^2$ , ce qui signifie que l'on peut mettre en œuvre l'algorithme itérativement. Ceci étant posé, l'algorithme de Viterbi procède exactement comme

nous l'avons indiqué dans le chapitre sur le récepteur optimal : parmi tous les chemins entrants sur un noeud, on ne conserve que celui de plus faible métrique cumulée, et seul ce chemin est prolongé. La complexité calculatoire passe alors de  $M^K$  à  $KM^L$  chemins à évaluer. On répète ces opérations au cours du temps, et on peut prendre les décisions avec un retard de  $5L$ .

Cet égaliseur est l'égaliseur « extrême ». En effet, il affiche les meilleures performances, voir la figure 4.6, et la plus grande complexité en terme d'implantation et charge calculatoire : il n'est guère utilisable que pour des séquences binaires ( $M = 2$ ) et des canaux courts ( $L < 10$ ). C'est la raison pour laquelle nous n'avons pu calculer les performances pour le canal (a), dont la réponse impulsionnelle à 11 coefficients, nous mène à un treillis à 2048 états. . . En outre, on ne peut utiliser cet algorithme qu'à condition d'avoir accès à la réponse impulsionnelle du canal, ce qui nécessite donc d'identifier le canal préalablement à la transmission.

Figure 4.6 – Performances de l'égaliseur à maximum de vraisemblance.

## Égaliseur transverse adaptatif

Les égaliseurs précédents souffrent de deux limitations communes : d'une part une charge de calcul importante, et d'autre part un caractère « statique ». En effet, le canal est le plus souvent non seulement inconnu, mais variable dans le temps. On peut alors utiliser des périodes de « synchronisation » où on émet des séquences d'apprentissage afin de recalculer l'égaliseur. Ceci n'empêche cependant pas les performances de se dégrader entre deux étapes de synchronisation.

Les méthodes adaptatives sont des méthodes simples qui permettent de résoudre simultanément les problèmes liés à la méconnaissance du canal et à son caractère évolutif et déterminer le filtre égaliseur  $\mathbf{e}$ .

Il s'agit toujours de résoudre l'équation 1. En minimisant l'erreur moyenne quadratique, on a obtenu

$$\begin{aligned} \frac{\partial \Xi(\mathbf{e})}{\partial \mathbf{e}} &= 2\mathbb{E} \left[ \mathbf{y}(n) \left( \mathbf{e}^T \mathbf{y}(n) - a_{n-R} \right) \right] \\ &= -2\mathbb{E} \left[ \mathbf{y}(n) \epsilon(n) \right] = -2\mathbf{R}_{\mathbf{y}\epsilon}, \end{aligned}$$

où l'on a posé  $\epsilon(n) = a_{n-R} - \mathbf{e}^T \mathbf{y}(n)$  le terme d'erreur. L'approche adaptative consiste à rendre les quantités apparaissant dans 1 ou dans la relation précédente dépendantes du temps. L'approche la plus simple consiste à résoudre 1 en utilisant un algorithme du gradient :

$$\mathbf{e}(n) = \mathbf{e}(n-1) - \frac{\mu}{2} \frac{\partial \Xi(\mathbf{e})}{\partial \mathbf{e}},$$

où  $\mu$  est une constante positive. Dans notre cas, ceci fournit

$$\mathbf{e}(n) = \mathbf{e}(n-1) + \mu \mathbf{R}_{\mathbf{y}\epsilon}.$$

Plutôt que de manipuler l'intercorrélation  $\mathbf{R}_{\mathbf{y}\epsilon}$ , on remplace celle-ci par son estimée instantanée<sup>2</sup>

$$\mathbf{R}_{\mathbf{y}\epsilon}^n = \mathbf{y}(n) \epsilon(n).$$

L'algorithme devient alors simplement

$$\begin{cases} \mathbf{e}(n) = \mathbf{e}(n-1) + \mu \mathbf{y}(n) \epsilon(n), \\ \epsilon(n) = a_{n-R} - \mathbf{e}(n-1)^T \mathbf{y}(n). \end{cases}$$

Cet algorithme, qui résoud par un algorithme de gradient approché un critère quadratique, est le célèbre algorithme du « gradient stochastique » ou de « *Least Mean Square* » (LMS).

<sup>2</sup>On notera que cette estimée, même si elle peut paraître un peu « brutale », n'est pas biaisée, puisque  $\mathbb{E} \left[ \mathbf{R}_{\mathbf{y}\epsilon}^n \right] = \mathbb{E} \left[ \mathbf{y}(n) \epsilon(n) \right] = \mathbf{R}_{\mathbf{y}\epsilon}$ .

On notera que  $e(n-1)^T y(n)$  est simplement la sortie du filtre adaptatif à l'instant  $n$ ,  $z(n)$ . Cet algorithme permet donc, à chaque instant, de remettre à jour les coefficients du filtre, proportionnellement à l'erreur d'estimation  $e(n)$ . En cas de variations des caractéristiques du canal, l'égaliseur sera donc capable de s'adapter à celles-ci, et ce d'autant plus rapidement que  $\mu$  est grand. En contrepartie, plus  $\mu$  est grand, plus les variations liées au bruit d'observation induiront une variabilité sur les estimées  $e(n)$ .

La mise en œuvre de l'algorithme se fait suivant deux modes opératoires, voir *figure 4.7* :

- un mode supervisé, où la séquence  $a(n)$  est connue (apprentissage). Le calcul de  $z(n)$  ne sert alors qu'à adapter le filtre, jusqu'à convergence. Au bout de  $K$  itérations, on considère que  $e(K)$  a convergé vers la solution, et
- on commute en mode opérationnel. La sortie de l'égaliseur  $z(n)$  sert alors à estimer  $a_{n-R}$  :  $\hat{a}_{n-R} = \text{dec}(z(n))$ , où  $\text{dec}$  indique que l'on prend la décision sur  $z(n)$ . L'erreur est alors maintenant calculée à partir des décisions :  $e(n) = \text{dec}(z(n)) - z(n)$ .

Figure 4.7 – Égaliseur piloté par les décisions

La théorie montre que sous des hypothèses d'ergodisme et de moments bornés,  $e(n)$  atteint  $e$  asymptotiquement, c'est-à-dire lorsque  $n \rightarrow +\infty$ , avec une précision inversement proportionnelle à  $\mu$ , et une vitesse de convergence proportionnelle à  $\mu$ . Le pas optimal de l'algorithme est de l'ordre de  $\mu = 2/(\alpha M P_y)$ , où  $\alpha$  est de l'ordre de 2 ou 3 et  $P_y$  est une estimée de la puissance d'entrée  $1/N \sum_{i=1}^N y(i)^2$ .

Cet algorithme est très simple d'implantation, parfaitement stable numériquement et très peu coûteux ( $M$  multiplications). En phase opérationnelle, l'algorithme est dit « piloté par les décisions » (*decision directed*). Un exemple d'implantation en MATLAB, qui illustre cette simplicité d'implantation, est fourni sur le LISTING 4-2 ;

LISTING 4-2 : IMPLANTATION DU GRADIENT STOCHASTIQUE (APPRENTISSAGE)

```
% y : séquence d'observations           % z : séquence de sortie de l'égaliseur
% R : valeur du retard                 % e : erreur
% Napp : longueur d'apprentissage     % e_lms : réponse impulsion-
nelle de l'égaliseur
% sym : séquence de symboles          % mu : valeur du pas d'adaptation
%% Mode supervisé
% ----> Le gradient est calculé avec apprentissage

for n=R+1:Napp
    y_v=y(n:-1:n-M+1); %y_v=[y(n) ... y(n-M+1)];
    z(n)=y_v'*e_lms;
    e(n-R)=sym(n-R)-z(n);
    e_lms=e_lms+mu*e(n-R)*y_v;
end
```

En mode opérationnel, il suffit de poursuivre la boucle `for` par

```
for n=Napp+1 :N
et remplacer le calcul de l'erreur par
    e(n-R)=sign(z(n))-z(n);
```

Lorsque le canal est peu sévère, et pourvu que l'initialisation  $e(0)$  soit raisonnable, on peut se dispenser de période d'apprentissage. Plusieurs variantes de cet algorithme existent. Parmi celles-ci on peut citer l'algorithme LMS normalisé, où le pas d'adaptation  $\mu$  est normalisé par la puissance de l'entrée (moins sensible à des variations de puissance de l'entrée), ou « l'algorithme du signe », dans lequel

on remplace le terme d'erreur dans l'expression de  $\mathbf{e}(n)$  par son signe. Ceci est précieux pour l'implantation sur processeurs spécialisés, dans la mesure où la dynamique du produit  $\mathbf{y}(n)\text{Signe}\{\epsilon(n)\}$  est identique à celle de  $\mathbf{y}(n)$ .

Lorsque la rapidité de convergence est importante, on peut utiliser un algorithme du second ordre, comme l'algorithme des moindres carrés récursifs (MCR), ou *Recursive Least Squares*,

$$\mathbf{e}(n) = \mathbf{e}(n-1) + \mu \mathbf{G}(n) \mathbf{y}(n) \epsilon(n),$$

où  $\mathbf{G}(n)$  est une estimation de  $\mathbf{R}_{yy}^{-1}$  obtenue récursivement

$$\mathbf{G}(n) = \frac{1}{1-\mu} \left( \mathbf{G}(n-1) - \frac{\mu \mathbf{G}(n-1) \mathbf{y}(n) \mathbf{y}(n)^T \mathbf{G}(n-1)}{1-\mu + \mu \mathbf{y}(n)^T \mathbf{G}(n-1) \mathbf{y}(n)} \right).$$

Cet algorithme converge nettement plus vite que l'algorithme du gradient stochastique, mais il est également beaucoup plus coûteux.

## 4.2.2 Égaliseur récursif à retour de décision

Il s'agit ici de prolonger l'idée d'avoir un égaliseur piloté par les décisions, ce qui permet d'éviter une répétition de séquences d'apprentissage, tout en utilisant une structure récursive. Cette structure récursive permet d'obtenir des filtres de réponse impulsionnelle longue à l'aide d'un petit nombre de coefficients. Un tel égaliseur est représenté sur la *figure 4.8*.

Figure 4.8 – Égaliseur récursif à retour de décision

Dans le cas optimal, la décision sur  $u(n)$  est égale à la séquence d'entrée, à un retard près :  $\hat{a}_{n-R} = z(n) = \text{dec}(u(n))$ , c'est-à-dire que le signal est parfaitement égalisé. On a alors  $z(n) = u(n) = \sum_{i=0}^q b_i y(n-i) + \sum_{j=1}^p a_j z(n-j)$ , où les  $b_i$  et les  $a_j$  sont les coefficients du filtre. Afin de calculer ces coefficients, on peut utiliser un critère d'erreur quadratique moyenne. On peut en outre utiliser une approche de type gradient stochastique, ou LMS, de sorte à profiter de capacités adaptatives et alléger la charge en calcul. Posons ainsi

$$\begin{cases} \mathbf{e}(n) = [b_0(n) \dots b_q(n) \ a_1(n) \dots a_p(n)]^T = [\mathbf{b}(n)^T \ \mathbf{a}(n)^T]^T, \\ \mathbf{v}(n) = [y(n) \dots y(n-q) \ z(n-1) \dots z(n-p)]^T = [\mathbf{y}(n)^T \ \mathbf{z}(n)^T]^T. \end{cases}$$

À l'aide de ces notations, l'erreur  $\epsilon(n)$  s'écrit  $\epsilon(n) = a_{n-R} - u(n)$ , en mode apprentissage, et  $\epsilon(n) = z(n) - u(n)$  en mode piloté par les décisions, et l'algorithme LMS s'écrit alors

$$\mathbf{e}(n) = \mathbf{e}(n-1) + \mu \mathbf{v}(n) \epsilon(n),$$

ou encore

$$\begin{cases} \mathbf{b}(n) = \mathbf{b}(n-1) + \mu \mathbf{y}(n) \epsilon(n), \\ \mathbf{a}(n) = \mathbf{a}(n-1) + \mu \mathbf{z}(n) \epsilon(n), \\ \text{avec } u(n) = \mathbf{b}(n)^T \mathbf{y}(n) + \mathbf{a}(n)^T \mathbf{z}(n). \end{cases}$$

Lors de la mise en œuvre, on utilisera une période d'apprentissage, lors de laquelle on prendra les symboles connus pour  $z(n)$ ,  $z(n) = a_{n-R}$ , puis on basculera en mode opérationnel, en remplaçant la séquence d'apprentissage par les décisions,  $z(n) = \text{dec}(u(n))$ . Le LISTING 4-3 donne ainsi les quelques lignes MATLAB qui permettent d'implanter cet égaliseur.



### LISTING 4-3 : IMPLANTATION DE L'ÉGALISEUR RÉCURSIF À RETOUR DE DÉCISIONS

```

% Égaliseur récursif à retour de décisions
% M : taille de B(z)
% L : taille de A(z)
% Lc : longueur de la RI du canal
% Napp : longueur de la séquence d'apprentissage
% y : séquence d'observations      % z : séquence de sortie de l'égaliseur
% R : valeur du retard            % e : erreur
% sym : séquence de symboles      % mu : valeur du pas d'adaptation

% Initialisations
N=length(y);      % Longueur de la séquence d'observations
R=Lc+M-L;        % Valeur du retard
y_v=zeros(M,1);  % Initialisations des vecteurs y et z
z_v=zeros(L,1);

% Boucle de calcul de l'égaliseur
for n=R+M:Napp
    y_v=y(n:-1:n-M+1);      %y_v=[y(n) ... y(n-M+1)];
    z_v=sym(n-R:-1:n-R-M+1); % en mode apprentissage
    %z_v=z(n:-1:n-M+1);    % en mode opérationnel
    u=y_v'*b_lms-z_v'*a_lms; % Calcul de la sortie
    e=sym(n-R)-u;          % Erreur en mode apprentissage
    %e=z(n)-u;             % Erreur en mode opérationnel
    z(n+1)=sign(u);        % Décision
    b_lms=b_lms+mu*e*y_v;  % Mise à jour
    a_lms=a_lms-mu*e*z_v;  % des coefficients
end

```

Cet égaliseur est d'une charge calculatoire faible (peu de coefficients), mais il peut présenter des pôles instables, qui entraînent une divergence. Ce type d'égaliseur présente d'excellentes performances, y compris en environnement sévère. Ces performances sont rapportées sur la *figure 4.9*.

Figure 4.9 – Performances de l'égaliseur à retour de décisions.

## 4.3 Méthodes autodidactes

Les méthodes précédentes, qui nécessitent une ou plusieurs périodes d'apprentissage, pénalisent beaucoup de systèmes de communications. Lorsque le canal subit des variations brutales, les algorithmes adaptatifs peinent à poursuivre ces variations, et si le système est piloté par les décisions, qui deviennent fausses, il peut s'en suivre une totale désadaptation. On doit alors recourir à un apprentissage régulier, ce qui limite singulièrement le débit. De plus, dans un système de communications multipoints (comme la télédiffusion audio ou vidéo), dès qu'un récepteur n'arrive plus à égaliser le canal, il faut transmettre une séquence d'apprentissage, ce qui prive tous les autres récepteurs du flot d'information. C'est pour ces raisons que des méthodes autodidactes (on parle aussi de *méthodes aveugles*, c'est-à-dire qui ne requièrent pas de période d'apprentissage, ont été développées. Parmi le très grand nombre de méthodes proposées, nous ne présenterons ici que deux approches, l'une fondée sur une structure cascade de deux égaliseurs « spécialisés » (en suivant la présentation de O. Macchi, [3]), et l'autre qui utilise une famille de fonctions de coût non-quadratiques, en mettant en jeu une fonction non linéaire de la sortie  $z(n)$  de l'égaliseur.

Dans la mesure où ces méthodes travaillent sans référence, il subsiste certaines indéterminées sur la séquence obtenue. On obtient ainsi une indéterminée sur le gain, le signe et la phase (puisque en

multipliant la réponse impulsionnelle *inconnue* du canal par un facteur complexe  $A$ , la séquence égalisée est divisée par  $A$ ). De plus, si la réponse impulsionnelle du canal ou de l'égaliseur est décalée dans le temps, on obtient une sortie décalée. Comme l'entrée est inconnue, on ne dispose pas de référence temporelle, et la séquence de sortie sera donc reconstituée à un retard près. En ce qui concerne le gain, on contraint en général la puissance de sortie, par exemple avec  $E[|z(n)|^2] = 1$ , en introduisant une commande automatique de gain, avant l'étape de décision.

### 4.3.1 Égaliseur autodidacte cascade

Les zéros de la fonction de transfert du canal, c'est-à-dire de la transformée en  $z$  de sa réponse impulsionnelle, n'ont pas de raison particulière d'être de module inférieur ou supérieur à 1. Ainsi, la fonction de transfert inverse possède des pôles à l'intérieur et à l'extérieur du cercle unité, et la réponse impulsionnelle correspondante, stable, comporte une partie causale et une partie non causale. Dans le cas, où les zéros sont à l'intérieur du cercle unité, on parle de canal à phase minimale, dans le cas contraire de canal à phase maximale, et enfin, dans le cas général de canal à phase mixte. Pour aboutir à un égaliseur général, on peut procéder par la mise en série de deux égaliseurs spécialisés, l'un pour les canaux à minimum de phase et l'autre pour les canaux à maximum de phase.

#### Canal à phase minimale – prédicteur

Dans le cas d'un canal à phase minimale, les choses sont relativement simples. En supposant que les symboles d'entrée sont *décorrélés*, c'est-à-dire que l'on a éventuellement employé à l'émission un embrouilleur pour annuler les corrélations, l'égaliseur consiste simplement à *blanchir* (à l'ordre 2) la séquence d'observations  $y(n)$ , afin d'obtenir en sortie une séquence  $z(n)$  décorrelée. Or on sait, voir par exemple [2], que le seul blanchisseur *causal* (ce qui correspond à l'hypothèse de zéros de module inférieur à 1) est obtenu par la prédiction linéaire en moyenne quadratique. Ainsi, si l'on cherche à prédire la valeur  $y(n)$  à partir d'un passé de longueur  $p$ , sous la forme  $\hat{y}(n) = \sum_{i=1}^p \alpha_i y(n-i)$ , alors, l'erreur  $\epsilon(n) = \hat{y}(n) - y(n)$ , appelée *innovation*, est une séquence une séquence décorrelée, que l'on peut donc identifier à la séquence d'entrée  $a_n$ . L'égaliseur obtenu est ainsi un égaliseur de type transverse, représenté sur la *figure 4.10*.

Figure 4.10 – Structure du prédicteur transverse

En minimisant la puissance de l'innovation,  $E[|\epsilon(n)|^2]$ , on obtient facilement le gradient par rapport aux  $\alpha_i$ , et on en déduit les équations normales qui fournissent les coefficients  $\alpha_i$  :

$$\alpha = \mathbf{R}_{yy}^{-1} \mathbf{r}_y,$$

où  $\alpha = [\alpha_1, \dots, \alpha_p]^T$ , et où  $\mathbf{R}_{yy}$  et  $\mathbf{r}_y$  sont respectivement la matrice de corrélation et le vecteur d'inter-corrélation de  $y(n)$  :

$$\mathbf{R}_{yy} = \begin{bmatrix} R_{yy}(0) & \dots & R_{yy}(p-1) \\ \vdots & & \vdots \\ R_{yy}(1-p) & \dots & R_{yy}(0) \end{bmatrix} \text{ et } \mathbf{r}_y = \begin{bmatrix} R_{yy}(-1) \\ \vdots \\ R_{yy}(-p) \end{bmatrix}.$$

Bien entendu, il est possible d'adopter un gradient stochastique afin de minimiser  $E[|\epsilon(n)|^2]$ , et on obtient alors

$$\begin{cases} \alpha(n) = \alpha(n-1) + \mu \mathbf{y}(n-1) \epsilon(n), \\ \epsilon(n) = y(n) - \alpha(n-1)^T \mathbf{y}(n-1), \\ \text{avec } \mathbf{y}(n-1) = [y(n-1) \dots y(n-p)]^T. \end{cases}$$

Du fait de l'hypothèse de phase minimale, il est également possible de trouver un blanchisseur sous forme récursive. En effet, si l'on note  $k(i)$  les coefficients de la réponse impulsionnelle du canal équivalent (incluant le canal mais également les filtres d'émission et de réception), alors  $y(n) = \sum_{i=0}^M k(i)a_{n-i}$ , et réciproquement,

$$a_n = \sum_{i=1}^M \frac{k(i)}{k(1)} a_{n-i} - \frac{1}{k(1)} y(n).$$

La séquence  $a_n$  étant blanche (à l'ordre 2), il existe donc un blanchisseur récursif, et les coefficients optimaux  $e(i)$  de l'égaliseur récursif seraient tels que  $e(i) = k(i)/k(1)$ . La structure de cet égaliseur est donnée *figure 4.11*.

Figure 4.11 – Structure du prédicteur récursif

La difficulté est que le calcul exact du gradient de  $E[|\epsilon(n)|^2]$  est impossible, et l'on doit adopter une solution approchée, comme

$$\begin{cases} \mathbf{e}(n) = \mathbf{e}(n-1) + \mu \mathbf{e}(n-1) \epsilon(n), \\ \epsilon(n) = y(n) - \mathbf{e}(n-1)^T \boldsymbol{\epsilon}(n-1), \\ \text{avec } \boldsymbol{\epsilon}(n-1) = [\epsilon(n-1) \dots \epsilon(n-p)]^T. \end{cases}$$

L'avantage de la structure récursive est qu'elle reste avec un nombre de coefficients faible, même lorsque la réponse impulsionnelle de l'égaliseur doit être longue.

On notera que la commande automatique de gain peut également être réalisée par un algorithme adaptatif où l'on chercherait par exemple à minimiser  $E[|z(n)^2 - 1|^2]$ , ce qui fournit

$$g(n+1) = g(n) + \mu g(n)(z(n)^2 - 1).$$

## Canal à phase maximale – rétroinnovateur

Lorsque le canal n'est pas à phase minimale, le prédicteur précédent n'est pas adapté et les performances sont faibles. Pour un tel canal, les zéros de  $C(z)$  sont tous situés à l'extérieur du cercle unité, et la réponse impulsionnelle correspondante est typiquement croissante en module. Il s'ensuit que l'information dominante est portée par les derniers coefficients de la réponse impulsionnelle et que le retard de restitution, ou d'égalisation, doit être égal à la longueur de la réponse impulsionnelle du canal. La fonction de transfert de l'égaliseur,  $E(z)$ , doit tendre vers  $1/C(z)$ , à un retard près. Or ce filtre n'admet pas de représentation récursive stable, car les pôles sont de module inférieur à 1. Par contre, on peut y associer un filtre transverse en prenant le développement en série tronqué de  $1/C(z)$  :

$$\frac{1}{C(z)} = \frac{1}{\sum_i c_i z^{-i}} \simeq 1 - \sum_{j=1}^{M-1} \beta_{M-j} z^j = z^{M-1} \left( z^{1-M} - \sum_{j=1}^{M-1} \beta_j z^{-j+1} \right) = z^{M-1} R(z).$$

Ainsi,  $R(z)$  correspond à un filtre transverse que l'on appelle *rétro-innovateur*, dont la structure est fournie sur la *figure 4.12*. Celui-ci correspond à évaluer une erreur de prédiction *rétrograde*  $\epsilon(n) =$

Figure 4.12 – Structure d'un rétro-innovateur

$y(n-M+1) - \hat{y}(n-M+1)$ , avec  $\hat{y}(n-M+1) = \sum_{j=1}^{M-1} \beta_j z^{-j+1}$ . En minimisant la puissance de l'erreur  $E[|\epsilon(n)|^2]$ , à l'aide d'un algorithme de gradient stochastique, on obtient alors

$$\begin{cases} \beta(n) = \beta(n-1) + \mu \mathbf{y}(n-1) \epsilon(n), \\ \epsilon(n) = y(n-M+1) - \beta(n-1)^T \mathbf{y}(n-1), \\ \text{avec } \mathbf{y}(n-1) = [y(n-1) \dots y(n-p)]^T. \end{cases}$$

## Canal à phase mixte – cascade rétro-innovateur et innovateur

Dans les paragraphes précédents, on a traité le cas des canaux à minimum de phase, puis à maximum de phase. On peut regrouper les deux approches afin de traiter le cas des canaux à phase mixte. En effet, la fonction de transfert du canal peut s'écrire comme un produit de ses zéros, c'est-à-dire finalement comme le produit d'une fonction de transfert à phase minimale et d'une fonction de transfert à phase maximale, correspondant respectivement aux zéros de module supérieur et inférieur à 1 :

$$C(z) = \prod_{i=1}^N (1 - z_i z^{-1}) = \prod_{i=1}^M (1 - z_i z^{-1}) \prod_{j=M+1}^N (1 - z_j z^{-1}).$$

On peut donc utiliser un égaliseur cascade composé d'un bloc prédicteur et d'un bloc rétro-innovateur, chacun prenant en charge une portion de la fonction de transfert du canal. Le choix de l'ordre des blocs dans la cascade peut-être choisi afin que la structure puisse être aisément commutée en mode piloté par les décisions. Pour ce faire, on choisit d'utiliser un prédicteur récursif, placé en aval du rétro-innovateur, comme indiqué sur la *figure 4.13*. L'égalisation se déroule alors en deux phases. Durant la

Figure 4.13 – Structure d'un égaliseur autodidacte en cascade

première phase (commutateur en position 1), l'égaliseur fonctionne en aveugle et est dit *en phase de convergence*. On peut ensuite basculer en mode à *retour de décisions* (commutateur en position 2), ce qui a pour effet d'améliorer les performances.

La séquence  $\epsilon_2(n)$  doit être une séquence blanche, puisque les données  $a_n$  sont supposées être blanches. Par contre,  $\epsilon_1(n)$  n'a pas de raison d'être blanche, (en dehors du cas où le canal serait purement à phase maximale). Le critère à minimiser lors de la phase de convergence est donc  $E[|\epsilon_2(n)|^2]$ , par rapport aux coefficients  $\alpha$  et  $\beta$ . En mode retour de décisions, on utilise le critère  $E[|z(n) - \epsilon_2(n)|^2]$ , où  $z(n) = \text{dec}(\epsilon_2(n))$ . Afin d'utiliser simultanément ces deux fonctions de coût, on utilise un critère mixte

$$J(\alpha, \beta) = \rho E[|\epsilon_2(n)|^2] + (1 - \rho) E[|z(n) - \epsilon_2(n)|^2], \quad (2)$$

où la valeur de  $\rho$  varie au cours du temps :  $\rho = 1$  en début de convergence, pour atteindre  $\rho = 0$  en phase retour de décisions. Dans [3], la règle d'adaptation de  $\rho$  est choisie comme  $\rho(n) = \tanh|z(n) - \epsilon_2(n)|$ , où  $\tanh$  est la tangente hyperbolique. La minimisation du critère composite 2 conduit alors aux gradients stochastiques

$$\begin{cases} \alpha(n) = \alpha(n-1) - \mu_1 \mathbf{z}(n-1) \epsilon_\rho(n), \\ \beta(n) = \beta(n-1) - \mu_2 \mathbf{u}(n) \epsilon_\rho(n), \\ \text{avec } \epsilon_\rho(n) = (1 - \rho) z(n) - \epsilon_2(n), \\ \mathbf{z}(n-1) = [y(n-1) \dots y(n-M)]^T, \\ \mathbf{u}(n) = [u(n) \dots u(n-K+1)]^T, \\ \text{et } u(n) = y(n) - \alpha(n-1)^T \mathbf{u}'(n-1), \end{cases}$$

où  $\mathbf{u}'(n-1)$  est le vecteur  $\mathbf{u}(n-1)$  rapporté à la dimension  $M$ , et  $K$  est le nombre de coefficients du rétro-innovateur.

Du point de vue des performances, cette structure permet d'obtenir des performances équivalentes, sinon supérieures à celles des égaliseurs transverses supervisés (avec période d'apprentissage) ou de l'égaliseur récursif à retour de décisions. Une autre structure cascade, ainsi que des applications, sont présentées dans [3] et [4].

### 4.3.2 Autres égaliseurs autodidactes

La clé de l'égalisation autodidacte réside dans le choix d'une méthode d'adaptation des coefficients de l'égaliseur, qui permette de pallier à l'absence d'apprentissage. Pour cela, on utilise des

fonctions de coût non quadratiques, qui permettent de prendre en compte les statistiques d'ordre élevé du signal observé  $y(n)$ . L'égaliseur est donc défini par une certaine fonction de coût  $E[\Psi(y(n))]$ . Ensuite, un gradient stochastique, ou une méthode du second ordre, conduit à un égaliseur adaptatif.

Les algorithmes de Godard utilisent la fonction de coût

$$\Psi(z(n)) = \left( |z(n)|^q - \frac{E[|a_n|^{2q}]}{E[|a_n|^q]} \right)^p, \text{ pour } p, q = 1, 2, \dots$$

Sous l'hypothèse que les symboles  $a_n$  soit circulaires,  $E[a_n^2] = 0$ , et que le kurtosis soit négatif, Godard a montré que le minimum de la fonction de coût conduit à l'égaliseur optimal. Pour  $p, q$  égaux à deux, on a une erreur  $\epsilon(n) = |z(n)|^2 - E[|a_n|^4]/E[|a_n|^2]$ , et le carré de cette erreur conduit à la fonction de coût de Godard pour  $p, q = 2$ . L'algorithme du gradient stochastique correspondant s'écrit

$$\mathbf{e}(n) = \mathbf{e}(n-1) + \mu z(n) \mathbf{y}(n) \epsilon(n).$$

Cet algorithme tendant à annuler l'erreur  $\epsilon(n)$ , il est clair qu'il privilégie les solutions de module constant. Il a été développé dans cet esprit, ainsi que différentes variantes, sous le nom de *Constant Modulus Algorithm* (CMA). Cet algorithme s'implante très simplement. Il suffit en fait de remplacer le calcul de l'erreur dans le gradient stochastique du LISTING 4-2 par

$$\{ e(n-R) = (1 - \text{abs}(z(n)^2)) * z(n) \}.$$

En fait, les algorithmes de Godard appartiennent à une classe plus large d'algorithmes appelés algorithmes de Bussbang. Le plus simple des algorithmes de Bussbang est l'algorithme piloté par les décisions, dans lequel on n'emploie pas de séquence d'apprentissage. Dans ce cas, les performances sont assez pauvres, à moins que la condition initiale soit bien choisie. Le premier algorithme d'égalisation autodidacte proposé dès 1975 est l'algorithme de Sato, et appartient à la classe des algorithmes de Bussbang. La fonction de coût est définie par le biais de  $\Psi(z(n)) = z(n) - E[|z(n)|^2]/E[|z(n)|] \text{signe}(z(n))$ . Cet algorithme converge sous l'hypothèse de kurtosis négatif, et montre de meilleures performances que l'algorithme piloté par les décisions.

Bien d'autres approches de l'égalisation autodidacte sont rapportées dans une littérature abondante. Les références [3] et [5] constituent des points de départ pour une étude plus approfondie.

## 4.4 Listings des programmes spécialisés

LISTING 4-4 : FONCTION MATLAB DE CALCUL DE L'ÉGALISEUR À ZERO FORCING.

```

function e_zf=ffe_zf(y,sym,M,c,nom_canal)
% SYNTAXE: e_zf=ffe_zf(y,sym,M,c,nom_canal)
%
% DESCRIPTION : Calcul de la réponse impulsionnelle
% d'un égaliseur par zero forcing. Le principe est de
% résoudre le système linéaire présenté dans le paragraphe
% << égalisation par zero forcing >>, pour forcer la
% convolution entre la RI de l'égaliseur et la RI du
% canal à être une impulsion de Dirac décalée en R.
% En termes matriciels, ce système s'écrit ici  $Q \cdot e_{zf} = \delta$ .
%
% ENTRÉES :
% - y : signal observé
% - sym : séquence de symboles
% - M : longueur de l'égaliseur
% - c : réponse impulsionnelle du canal
% - nom_canal : nom du canal
%
% SORTIE :
% - e_zf : réponse impulsionnelle de l'égaliseur
%
% Auteurs : P. Jardin et J.-F Bercher
% Date : nov. 2000
% =====

% Initialisations :
N=length(c); % Longueur du canal
Q=zeros(N+M-1,M); % Initialisation de Q
delta=zeros(N+M-1,1); % Définition de Delta(n-R)
R=(N+M)/2;
delta(R)=1;

% Définition de la matrice Q décrivant le système linéaire
for i=1:N+M-1
    for j=max(1,i-N+1):min(M,i)
        Q(i,j)=c(i-j+1);
    end
end

% Inversion du système
e_zf=pinv(Q)*delta;

% Pour vérification, calcul de la sortie z de l'égaliseur
% et de la décision associée, et comparaison à la
% séquence de symboles

z=filter(e_zf,1,y);
N=50;
figure;
subplot(211); plot(sym(1:N)); title('Séquence de symboles')
subplot(212); plot(1:N,z(1+R:N+R),'- ',1:N,sign(z(1+R:N+R)),':')
title(['Égalisation ZF du canal ' nom_canal])

%\label{ListingZF}
%\caption{Fonction Matlab de calcul de l'égaliseur à zero forcing.}

```

LISTING 4-5 : FONCTION MATLAB DE CALCUL DE L'ÉGALISEUR MINIMISANT L'ERREUR QUADRATIQUE MOYENNE

```

function [e_eqm,Rya]=ffe_mse(y,sym,M,L,nom_canal)
% SYNTAXE: e_eqm=ffe_mse(y,sym,M,L,nom_canal)
%
% DESCRIPTION : Calcul de la réponse impulsionnelle
% d'un égaliseur minimisant l'erreur quadratique moyenne (EQM).
% Il s'agit résoudre le système linéaire présenté dans le paragraphe
% << égaliseur à erreur quadratique ation par zero forcing >>,
% équation (1) :  $e\_eqm = \text{inv}(Ryy)*Rya(R)$ , où Ryy est la matrice de
% corrélation et Rya(R) le vecteur d'intercorrélacion entre y(n)
% et a(n-R).
%
% ENTRÉES :
% - y : signal observé
% - sym : séquence de symboles
% - M : longueur de l'égaliseur
% - L : longueur de la RI du canal
% - nom_canal : nom du canal
%
% SORTIE :
% - e_eqm : réponse impulsionnelle de l'égaliseur
%
% Auteurs : P. Jardin et J.-F Bercher
% Date : nov. 2000
% =====

% Initialisations :

Ns=length(y);
R=(L+M)/2-1;

% Calcul de la fonction d'autocorrélacion
% [Estimateur biaisé]
ryy=xcorr(y,y,M+1); % rend une séquence pour les retards
% compris entre -(M) et (M)
% Matrice de corrélation
Ryy1=toeplitz(ryy(M+2:2*M+1)); % Calcul de la matrice a structure de
% Toeplitz, symétrique, engendrée par
% le vecteur [Ryy(0) ... Ryy(M-1)]

% Vecteur d'intercorrélacion
rya=xcorr(sym,y,R); % rend les intercorrélacions entre y
% et a pour les retards compris entre -R et R
Rya=rya(2*R+1:-1:2*R+2-M); % vecteur des intercorrélacions pour les
% retards R à R-M+1

% Calcul de l'égaliseur....
e_eqm=inv(Ryy1)*Rya';

% Pour vérification, calcul de la sortie z de l'égaliseur
% et de la décision associée, et comparaison à la
% séquence de symboles

Na=50;
z=filter(e_eqm,1,y);
figure;
subplot(211);plot(sym(1:Na)); title('Séquence de symboles')
subplot(212);plot(1:Na,z(1+R:Na+R),'- ',1:Na,sign(z(1+R:Na+R)),': ')
title(['Égalisation EQM de ' nom_canal])
%\label{ListingEQM}
%\caption{Fonction Matlab de calcul de l'égaliseur minimisant l'erreur qua-

```





# Bibliographie

- [1] Proakis J. G. – *Digital Communications*, 4<sup>e</sup> édition, New York, Mc Graw-Hill, 2000.
- [2] Picinbono B. – *Signaux aléatoires* (en trois tomes), *Bases du traitement statistique du signal* (tome 3), Paris, Dunod, 1995.
- [3] Macchi O. – *L'égalisation numérique en communications*, Annales des télécommunications, vol. 53, n° 1-2, pp. 39-58, 1998.
- [4] Labat J., Macchi O., Laot C. – *Adaptive decision feedback Equalizers : Can you skip the training period*, IEEE trans. on Communications, 1996.
- [5] Tugnait J. K., Tong L., Ding Z. – *Single User Channel Estimation and Equalization*, IEEE Signal Processing Magazine, pp. 17-28, may 2000.