

# Table des matières

Table des matières	ii
Liste des tableaux	iii
Table des figures	v
<b>INTRODUCTION</b>	<b>1</b>
<b>1 SYSTÈMES DE TRANSMISSION NUMÉRIQUE</b>	<b>3</b>
1.1 Introduction	3
1.2 Codage	4
1.2.1 Quelques notions de codage de canal	6
1.2.2 Les codes en blocs	9
1.2.3 Les codes convolutifs	12
1.2.4 Code reed-solomon	15
1.3 Modulation	18
1.3.1 Définition	18
1.3.2 Modulations par déplacement de phase (MDP, en anglais PSK, Phase Shift Keying)	20
1.3.3 Modulation d'Amplitude sur deux porteuses en quadrature (MAQ, en anglais QAM, Quadrature Amplitude Modulation)	21
1.3.4 Modulation par déplacement d'Amplitude et de phase (MDAP, en anglais APSK, Amplitude Phase Shift Keying)	23
1.4 Canal de transmission	25
1.4.1 Le canal stationnaire	25
1.4.2 Le canal non stationnaire	26
1.4.3 Le canal binaire	27
1.5 Conclusion	29
<b>2 STRUCTURE DE LA CHAÎNE DE TRANSMISSION NUMÉRIQUE</b>	<b>30</b>
2.1 Introduction	30
2.2 L'émetteur	31
2.2.1 La source	31

2.2.2	Codage canal . . . . .	32
2.2.3	Entrelacement . . . . .	38
2.2.4	Modulation . . . . .	39
2.3	Canal de transmission . . . . .	41
2.3.1	Canal à bruit blanc additif gaussien (Canal AWGN) . . . . .	41
2.3.2	Canal de Rayleigh . . . . .	42
2.4	Recepteur . . . . .	43
2.4.1	Démodulateur . . . . .	43
2.4.2	Désentrelacement . . . . .	44
2.4.3	Décodage de canal . . . . .	44
2.4.4	Décodeur source . . . . .	46
2.4.5	Bits reçus . . . . .	46
2.5	Conclusion . . . . .	46
<b>3</b>	<b>SIMULATIONS ET PERFORMANCES</b>	<b>48</b>
3.1	Introduction . . . . .	48
3.2	Implementation de La chaine de transmission . . . . .	49
3.3	Simulation de la chaîne de transmission sans codage . . . . .	49
3.3.1	Comparaison entre les méthodes de modulation sans codage . . . . .	52
3.3.2	Comparaison entre les méthodes de modulation avec canal rayleigh . . . . .	54
3.4	Insertion des blocs de codage et de décodage . . . . .	55
3.4.1	Insertion de codage convolutif . . . . .	55
3.4.2	Comparaison entre les méthodes de modulation avec codage convolutif dans un canal rayleigh . . . . .	57
3.4.3	poinçonnage . . . . .	58
3.4.4	Comparaison entre les méthodes de modulation avec codage . . . . .	59
3.4.5	Insertion de codage reed-solomon et l'entrelaceur . . . . .	60
3.5	Conclusion . . . . .	62
<b>4</b>	<b>Conclusion</b>	<b>63</b>
	<b>Bibliographie</b>	<b>65</b>
<b>A</b>	<b>LISTE DES SYMBOLES ET DES ABREVIATIONS</b>	<b>69</b>
<b>B</b>	<b>QUELQUES DÉFINITION</b>	<b>71</b>

# Liste des tableaux

1.1	Energie moyenne et distance euclidienne minimale des différentes modulations . . . . .	23
2.1	Table d'état du code convolutif NRNSC $R=1/2, K=3$ . . . . .	35
2.2	Vecteurs de poinçonnage . . . . .	37
2.3	Vecteurs de poinçonnage . . . . .	39
3.1	Vecteurs de poinçonnage . . . . .	59

# Table des figures

1.1	synoptique d'un systèmes de transmission numérique . . . . .	3
1.2	Représentation graphique du gain de codage . . . . .	7
1.3	Mots de code . . . . .	12
1.4	Schéma d'implémentation d'un code convolutif à l'aide de registre à décalage . . . . .	13
1.5	Exemple de régions de décision pour une modulation à 3 points alignés	19
1.6	Constellation de la modulation MDP-8 . . . . .	21
1.7	Constellation de la modulation MAQ-4/MDP-4 . . . . .	22
1.8	Constellation de la modulation MAQ-16 (constellation carrée) . . . . .	23
1.9	Constellation de la modulation MAQ32 (constellation en croix ) . . . . .	23
1.10	Constellation de la modulation MDAP-4-12 . . . . .	24
1.11	Constellation de la modulation MDAP-4-12-16 . . . . .	25
1.12	Modélisation du canal BBAG . . . . .	26
1.13	Description d'un canal binaire . . . . .	27
1.14	Canal binaire symétrique . . . . .	28
1.15	Modèle du canal de Gilbert-Elliot . . . . .	28
2.1	Le système de communications numériques . . . . .	30
2.2	Le bloc generateur Bernoulli . . . . .	31
2.3	Le bloc FEC . . . . .	33
2.4	Le bloc de codage reed-solomon . . . . .	33
2.5	Code convolutif NRNSC $R = 1/2$ , $K = 3$ . . . . .	34
2.6	Diagramme d'état d'un code convolutif . . . . .	35
2.7	Représentation en treillis d'un code convolutif . . . . .	36
2.8	Le bloc de codage convolutif . . . . .	37
2.9	Le bloc d'entrelacement . . . . .	39
2.10	LE bloc de modulation QPSK . . . . .	40
2.11	Le bloc de modulation QAM . . . . .	41
2.12	LE bloc de canal de transmission AWGN . . . . .	42
2.13	Le bloc de canal Rayleigh . . . . .	43
2.14	Le bloc de canal rayleigh . . . . .	43
2.15	Le bloc de demodulation QPSK . . . . .	43

2.16	Le bloc de désentrelacement . . . . .	44
2.17	Le bloc de décodeur viterbi . . . . .	45
2.18	Le bloc de décodeur reed-solomn . . . . .	46
3.1	Schéma global de la chaîne de transmission . . . . .	49
3.2	Modélisation de la chaîne de transmission sans codage . . . . .	50
3.3	Diagramme de constellation avant et après le canal de transmission BPSK	51
3.4	Diagramme de constellation avant et après le canal de transmission QPSK	51
3.5	Diagramme de constellation avant et après le canal de transmission 8PSK	52
3.6	comparaison entre la partie théorique et expérimentale d'une modulation BPSK . . . . .	53
3.7	Comparaison entre la partie théorique et expérimentale d'une modulation QPSK . . . . .	53
3.8	Performances du système pour différentes modulation sans codage . . .	54
3.9	Performances du TEB du système dans une modulation BPSK avec dans un canal Rayleigh . . . . .	54
3.10	performances du TEB du système dans une modulation QPSK dans un canal Rayleigh . . . . .	55
3.11	la chaîne de transmission avec codage convolutif . . . . .	56
3.12	Performances du T8B du système dans une modulation BPSK avec CC	56
3.13	Performances du TEB du système dans une modulation QPSK avec CC	57
3.14	Performances du TEB du système dans une modulation BPSK avec CC dans un canal Rayleigh . . . . .	58
3.15	Performances du TEB du système dans une modulation QPSK avec CC dans un canal Rayleigh . . . . .	58
3.16	Comparaison des performances du système pour la modulation BPSK et taux de codage CC . . . . .	59
3.17	Comparaison des performances du système pour la modulation QPSK et taux de codage CC . . . . .	60
3.18	Modélisation de la chaîne de transmission avec codage concaténé (CC+RS)	61
3.19	Le diagramme de constellation de codage concaténé . . . . .	61
3.20	Performances du système pour une modulation QPSK en utilisant un codage concaténé convolutif et reed-solomon avec un entrelaceur . . .	62

## INTRODUCTION

Au cours des dix dernières années, la demande pour des systèmes de transmission numériques fiables s'est considérablement accrue [1]. L'explosion de l'échange d'informations, et les nouvelles possibilités offertes par le traitement numérique du signal ont accentué cette tendance.

Les systèmes de communication modernes exigent des débits de plus en plus élevés afin de traiter des volumes d'informations en augmentation constante. Ils doivent être flexibles pour pouvoir gérer des environnements multinormes, et évolutifs pour s'adapter aux normes futures. La qualité d'une transmission numérique dépend principalement de la probabilité d'occurrence d'erreur dans les symboles transmis [2]. Cette probabilité étant fonction du rapport signal sur bruit, une amélioration de la qualité de transmission peut être envisagée en augmentant la puissance d'émission et en diminuant le facteur de bruit du récepteur. Malheureusement, cette solution implique des coûts énergétiques et technologiques importants, ce qui en limite sensiblement l'emploi. Le contrôle des erreurs par codage est ainsi indispensable. L'utilisation de techniques de traitement numérique du signal, et notamment le codage des informations à transmettre, permet la détection et/ou la correction d'éventuelles erreurs de transmission. Comme ces techniques permettent de contrôler les erreurs induites par le bruit du canal de transmission, elles sont nommées codages de canal. Parmi les principales techniques existantes, les codages en bloc et les codages convolutifs sont prédominants. Les codages en bloc sont utilisés notamment dans les réseaux Ethernet [3], dans les normes de transmission sans fils telles que bluetooth [4], et dans les normes de transmission HDTV [5] et DVB-C [6]. Le codage convolutif est très présent dans les systèmes de communication numérique sans fil [78]. Le contexte de ce projet se situe au niveau des systèmes de transmission fiable de données à haut débit. En effet, les systèmes actuels requièrent des débits de transmission très élevés, des techniques de modulation variées et des protocoles de communications complexes. Ces applications sont caractérisées par l'emploi, entre autres, de fonctions complexes et coûteuses en temps de traitement. Certaines de ces fonctions, telles que la modulation et la démodulation numériques, le filtrage, concernent l'adaptation du signal à la nature du support de transmission. D'autres fonctions, telles que le codage canal, ont pour but de renforcer la robustesse des informations transmises. La conception architecturale sûre de fonctionnement pour ces fonctions est un facteur déterminant, puisque notre objectif est non seulement de transmettre les données à haut débit mais aussi de les protéger contre des altérations non désirées. L'objectif principal de ce travail donc est de concevoir une architecture rapide et fiable pour un système de codage à faible coût et à haut débit permettant une protection optimale des données

Ce manuscrit est composé de trois chapitres.

Le premier chapitre présente quelques généralités sur les communications numériques. Après avoir décrit la chaîne de communication numérique, nous abordons plus en détail le codage de canal et les modulations numériques. Les principes des codes correcteurs d'erreurs et, plus particulièrement, ceux du codage convolutif et reed solomon sont présentés. De même, l'ensemble de concepts liés aux modulations numériques ainsi que les modulations utilisées dans notre étude sont décrits. Finalement nous nous intéressons aux deux modèles de canal de transmission considérés dans ce mémoire.

Le deuxième chapitre fait l'objet de l'étude du fonctionnement général du système émetteur-récepteur de la chaîne utilisé dans le cadre des communications numériques. L'émetteur, présenté est celui qui produira le signal devant être envoyé dans le canal avec exécution des opérations de randomisation, de codage de canal (FEC : Forward Error Correction), d'entrelacement, de modulation ; et la définition de chaque bloc utilisé dans la simulation puis les opérations inverse, la démodulation, le décodage canal et les différents canaux utilisés : canal awgn, canal Rayleigh.

Le troisième chapitre est consacré pour les simulations et les performances de notre chaîne en matlab afin de valider l'étude théorique. En fin nous terminons notre mémoire par une conclusion générale.

# Chapitre 1

## SYSTÈMES DE TRANSMISSION NUMÉRIQUE

### 1.1 Introduction

Les systèmes de transmission numérique véhiculent de l'information entre une source et un destinataire en utilisant un support physique comme le câble, la fibre optique ou encore, la propagation sur un canal radioélectrique. Les signaux transportés peuvent être soit directement d'origine numérique, comme dans les réseaux de données, soit d'origine analogique (parole, image...) mais convertis sous une forme numérique. La tâche du système de transmission est d'acheminer l'information de la source vers le destinataire avec le plus de fiabilité possible. Le schéma synoptique d'un système de transmission numérique est donné à la figure 1 où l'on se limite aux fonctions de base :

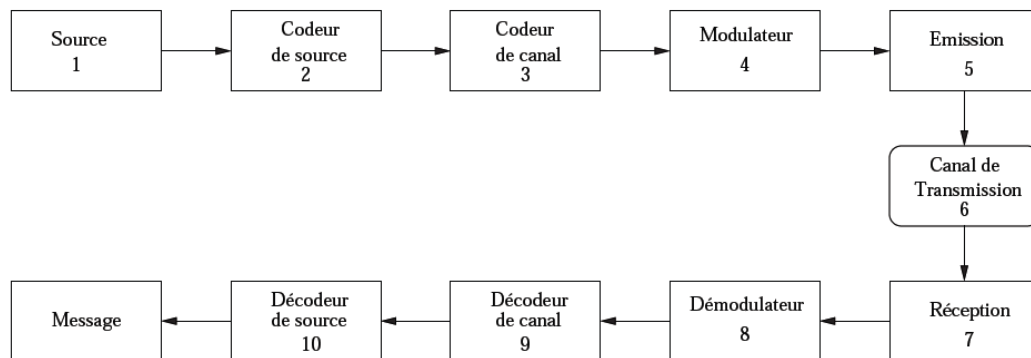


FIGURE 1.1 – synoptique d'un systèmes de transmission numérique



- La source émet un message numérique sous la forme d'une suite d'éléments binaires.
- Le codeur peut éventuellement supprimer des éléments binaires non significatifs (compression de données ou codage de source), ou au contraire introduire de la redondance dans l'information en vue de la protéger contre le bruit et les perturbations présentes sur le canal de transmission (codage de canal). Le codage de canal n'est possible que si le débit de source est inférieur à la capacité du canal de transmission (la probabilité d'erreur  $P_e$  tend dans ce cas vers 0 d'après les travaux de Hartley -Shannon).
- La modulation a pour rôle d'adapter le spectre du signal au canal (milieu physique) sur lequel il sera émis.
- Enfin, du côté récepteur, les fonctions de démodulation et de décodage sont les inverses respectifs des fonctions de modulation et de codage situées du côté émetteur.

Les trois caractéristiques principales permettant de comparer entre elles les différentes techniques de transmission sont les suivantes :

- La probabilité d'erreur  $P_e$  par bit transmis permet d'évaluer la qualité d'un système de transmission. Elle est fonction de la technique de transmission utilisée, mais aussi du canal sur lequel le signal est transmis. Il est à noter que  $P_e$  est une valeur théorique dont une estimation non biaisée au sens statistique est le Taux d'Erreur par Bit TEB.
- L'occupation spectrale du signal émis doit être connue pour utiliser efficacement la bande passante du canal de transmission. On est contraint d'utiliser de plus en plus des modulations à grande efficacité spectrale.
- La complexité du récepteur dont la fonction est de restituer le signal émis est le troisième aspect important d'un système de transmission.

## 1.2 Codage

Le codage de canal joue un rôle fondamental dans les systèmes de communication puisqu'il permet de protéger les données des perturbations introduites par le canal de transmission. L'une des questions fondamentales est de savoir la quantité d'information qu'il est possible de transmettre de façon fiable. En 1948, Th Shannon [9] démontre que pour une source d'information de débit  $D_s$  (bit/s) qui transmet sur un canal de capacité  $C_c$  (bit/s), il existe, si  $D_s < C_c$ , un code garantissant une transmission quasi parfaite. Ce théorème ne dit pas comment construire de tels codes, mais simplement qu'il est possible en optimisant le codage d'obtenir après le décodage une information avec très peu

d'erreurs. Des lors, de nombreux chercheurs se sont penchés sur la construction de ces codes.

Le bloc de codage de canal est en général équipé de un ou plusieurs code(s) correcteur(s) d'erreurs et d'un entrelaceur afin de lutter efficacement contre les erreurs de transmission. Le rôle du codeur est d'introduire à l'émission de la redondance dans le train binaire informatif; cette redondance permettant, du côté récepteur, de corriger et/ou de détecter d'éventuelles erreurs. Les bruits perturbateurs présents en pratique lors de transmissions radio-mobiles engendrent généralement des erreurs de type burst ou encore par paquets, c'est-à-dire que les perturbations affectant les bits d'information se produisent en rafales.

Dans cette situation, la présence d'un entrelaceur permet de limiter les effets de ce type d'erreurs en dispersant à l'émission les bits de chaque mot d'information à l'intérieur d'une trame de taille conséquente, très supérieure à celles des mots en question. Grâce à cela, les bits consécutifs concernés par la rafale d'erreurs n'appartiennent pas aux mêmes mots informatifs et peuvent être alors plus facilement et surtout efficacement corrigés après désentrelacement. La succession des deux traitements liés à l'opération d'entrelacement à l'émetteur et de désentrelacement au récepteur peut être vue pour simplifier comme une dispersion des erreurs tout au long de la trame, au lieu de quelles soient regroupées sur un ou plusieurs mots informatifs empêchant toute tentative de correction de celles-ci par le décodeur aux capacités de correction malgré tout limitées.

Dans la littérature, on distingue deux grandes familles de codes correcteurs d'erreurs, les codes en blocs et les codes convolutifs. Dans le cas des codes en blocs, l'information est découpée en blocs de  $k$  bits. Chaque bloc est transformé en un nouveau bloc de  $n$  bits ( $n > k$ ), qui forme un mot de code. Dans le cas d'un code convolutif, chaque bloc de  $n$  bits en sortie du codeur dépend non seulement des  $k$  bits présents en entrée du codeur mais également d'autres blocs de  $k$  bits ayant été introduits précédemment.

De même que les codeurs, il existe également deux grandes familles d'entrelaceurs, les entrelaceurs blocs et convolutifs. Un entrelaceur par bloc est un dispositif qui à partir d'un bloc de symboles reçu en entrée, va restituer en sortie un bloc contenant les mêmes symboles mais dans un ordre différent. Un entrelaceur convolutif (ou multiplexé) de période  $S$  est construit en multiplexant la séquence d'entrée de façon séquentielle en  $S$  sous-séquences, introduisant un retard pour chaque sous-séquence et en démultiplexant ces  $S$  sous-séquences avec prise en compte de ces retards.

De par leurs performances en terme de correction des erreurs et le fait qu'ils traitent l'information en flux, les codes convolutifs sont très prisés dans les communications radio-mobiles. En effet, que ce soit dans les standards de seconde génération (GSM [10]

et CDMA-2000 [11]), de troisième génération (UMTS [12] et WDMA) ou encore dans les futurs standards de quatrième génération (LTE [13]), on retrouve toujours l'utilisation de un ou plusieurs code(s) convolutif(s). De ce fait, nos travaux se sont portés sur la reconnaissance en aveugle des codes convolutif. L'identification aveugle des paramètres d'un code convolutif étant fortement basée sur les propriétés algébriques de ces codes, nous proposons dans ce premier chapitre une étude sur les propriétés des codes linéaires.

### 1.2.1 Quelques notions de codage de canal

Puisque ce projet aborde la problématique du codage de canal pour des communications à grande efficacité spectrale, l'opération de codage de canal dans la chaîne de transmission nécessite une description détaillée.

Dans cette section, nous introduisons quelques définitions relatives aux codes correcteurs d'erreurs, puis une description détaillée des codes directement liés au sujet de cette étude, à savoir les codes convolutifs.

#### Le codage correcteur d'erreurs

Le principe d'un code correcteur d'erreurs est d'ajouter une certaine quantité de bits de redondance à une séquence de bits d'information afin de pouvoir retrouver en réception la séquence d'information transmise. Nous nous limitons ici au codage linéaire de séquences binaires. Les codes binaires ne considèrent que les valeurs de l'alphabet binaire,  $B = \{0, 1\}$ . D'autre part, un code est dit linéaire si toute combinaison linéaire de mots de code est encore un mot de code.

La théorie de l'information énoncée par Th. Shannon en 1948 [9] montre qu'il existe des codes correcteurs d'erreurs qui rendent possible la transmission d'information avec une probabilité d'erreur arbitrairement petite. Shannon ne donne toutefois pas de solution pratique pour réaliser ces codes. Dans les années qui suivent, la théorie du codage de canal connaît des avancées importantes. Les codes en bloc sont introduits (par exemple, les codes de Hamming [14] et les codes produits [15]), ainsi que les codes convolutifs [16]. Nous allons introduire ci-après quelques définitions liées aux codes détecteurs-correcteurs d'erreurs avant d'aborder les codes convolutifs.

## Gain de codage

Les performances d'un code correcteur d'erreurs sont déterminées par une courbe de taux d'erreurs (TEB) en fonction du rapport signal à bruit de la transmission mesuré en dB. Ce rapport signal sur bruit s'exprime comme  $E_b/N_o$ , ou  $E_b$  est l'énergie transmise par élément binaire d'information utile et  $N_o$  la densité spectrale de puissance monolatérale de bruit gaussien .

Si l'on se place à une certaine valeur de taux d'erreurs, le gain de codage est déterminé par l'écart existant entre la courbe sans codage et la courbe avec codage. A titre d'illustration, sur la figure 1.2 nous avons représenté le gain de codage du code B à un  $TEB = 10^{-3}$  qui est d'environ 3,5 dB.

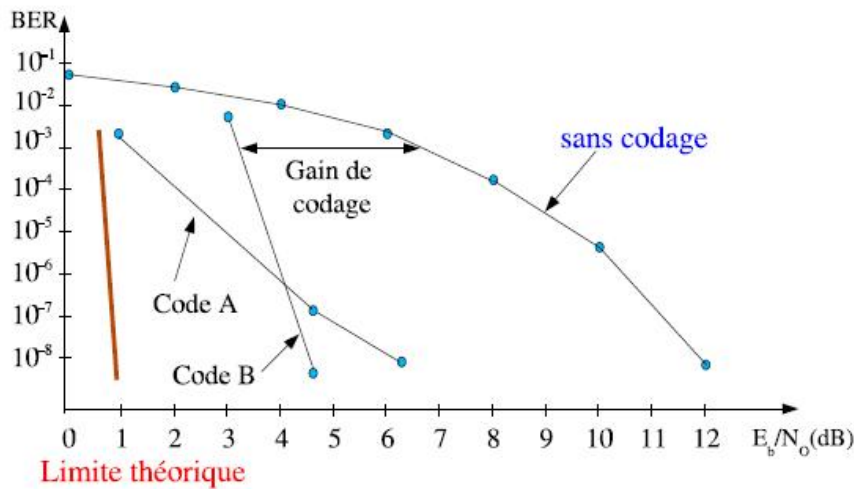


FIGURE 1.2 – Représentation graphique du gain de codage

## Distance minimale de hamming

Le poids de Hamming d'un mot de code est le nombre d'éléments binaires non nuls qu'il contient. Par exemple, si l'on considère un code binaire C dont la longueur des mots de code est 8 bits, le poids de Hamming de la séquence  $c_1 = (00101001)$  est égal à 3. Nous adopterons la notation  $w_H(c_1) = 3$ .

La distance de Hamming entre deux mots de code  $c_i$  et  $c_j$  est définie comme le nombre de bits différents entre  $c_i$  et  $c_j$ . Autrement dit, elle est égale au poids de Hamming du mot de code  $c_i \oplus c_j$ , ou  $\oplus$  représente la somme modulo 2, soit  $d_H(c_i, c_j) = w_H(c_i \oplus c_j)$ .

Prenant à nouveau le code  $C$  comme exemple, la distance de Hamming entre les séquences  $c_1 = (00101001)$  et  $c_2 = (10101000)$  est 2, et on le représente :  $d_H(c_1, c_2) = 2$ . La distance minimale de Hamming d'un code, notée  $d_{min}$ , est définie comme la distance de Hamming minimale entre l'ensemble de ses mots de code. Si le code est linéaire, elle peut également être calculée comme le poids minimal de ses mots non nuls :

$$d_{min} = \min_{c_i, c_j \in C, c_i \neq c_j} d_H(c_i, c_j) = \min_{c_i \in C, c_i \neq 0} w_H(c_i) \quad (1.1)$$

### Convergence et performance asymptotique

Deux paramètres sont à considérer lors de l'analyse des performances d'un code correcteur d'erreurs :

- **Le seuil de convergence** : il s'agit du rapport signal sur bruit à partir duquel le système codé devient plus performant que le système de transmission non-code. Lorsque ce seuil est faible, les performances du système à des niveaux de bruit fort et moyen sont proches de la limite théorique et on peut dire que le code présente une bonne convergence.
- **La performance asymptotique** : le comportement à très faible niveau de bruit du système de transmission codé est dicté par la distance minimale de Hamming du code et son rendement. Le gain asymptotique, noté  $G_a$ , est donné par l'expression suivante :

$$G_a \approx 10 \log (Rd_{min}) \quad (1.2)$$

où  $R$  est le rendement du code. Une faible valeur de  $d_{min}$  entraîne un fort changement de pente (on parle de flattening) dans la courbe de taux d'erreurs dû à un faible gain asymptotique.

Sur la figure 1.2 nous avons présenté les courbes de deux codes dont les comportements sont différents. Le code  $A$  présente un bon seuil de convergence mais un faible gain asymptotique. En revanche, le code  $B$  présente un bon comportement asymptotique mais un seuil de convergence plus élevé. La recherche d'un bon code dépend du taux d'erreur cible de l'application. En effet, pour les taux d'erreurs moyens ou élevés, il est

préférable de privilégier le seuil de convergence au détriment de la distance minimale du code. Pour les faibles taux d'erreurs, mieux vaut privilégier la distance minimale.

## 1.2.2 Les codes en blocs

Bien qu'il existe de nombreuses familles de codes en blocs, telles que les codes de Hamming [17], les codes BCH [18] [19], les codes Reed-Solomon [20], les codes LDPC [21], etc., le principe de codage reste le même : le regroupement d'un message d'entrée en blocs fin d'ajouter à chacun des messages une quantité contrôlée de redondance. Cette redondance introduite permettra à la réception de détecter et/ou de corriger d'éventuelles erreurs de transmission.

### codage

Notons  $C$  un  $(n, k)$  code en bloc linéaire de rendement  $r = k/n$ , dans le corps des symboles  $F$ . Dans le reste de notre étude, nous prendrons le corps de Galois à deux éléments ("0" et "1"), noté  $GF(2)$ . La matrice génératrice d'un code en bloc notée  $G$ , est une matrice de taille  $k \times n$  et de rang  $k$  qui permet d'engendrer le code  $C$ . Le nombre de sorties d'un codeur  $n$ , étant strictement supérieur au nombre d'entrées  $k$ , le codeur introduira systématiquement de la redondance au message. Nous appellerons un mot d'information, un vecteur de  $k$  bits d'information introduit simultanément en entrée du codeur et un mot de code, le vecteur de  $n$  bits qui en résulte. Notons  $m(t) = (m_1(t), \dots, m_k(t))$ , le mot d'information transmis à l'instant  $t$  et respectivement  $c(t) = (c_1(t), \dots, c_n(t))$ , le mot de code reçu à l'instant  $t$  de tel sorte que :

$$c_t = m_t \cdot G \text{ pour } t = 0, 1, \dots \quad (1.3)$$

Lors d'une transmission numérique, l'information qui est transmise ne correspond pas uniquement à un mot d'information mais à une séquence de mots d'information. En notant  $L$  la longueur de la séquence à coder, la séquence d'entrée notée  $m$  et la séquence de sortie notée  $c$  sont telles que :

$$m = (m(0), \dots, m(L-1)) = (m_1(0), \dots, m_k(0), \dots, m_1(L-1), \dots, m_k(L-1)) \quad (1.4)$$

$$c = (c(0), \dots, c(L-1)) = (c_1(0), \dots, c_n(0), \dots, c_1(L-1), \dots, c_k(L-1)) \quad (1.5)$$

La matrice de codage liant ces  $L$  mots de code aux  $L$  mots d'information est une matrice blocdiagonale constituée de  $L$  matrices  $G$ . Cette matrice de codage étant composée de nombreux "0", ces éléments ont été omis afin de simplifier la notation. L'équation de codage (1.5) devient :

$$c = m. \begin{pmatrix} G & & & \\ & G & & \\ & & \ddots & \\ & & & G \end{pmatrix} \quad (1.6)$$

Cette matrice de codage est une matrice de taille  $k.L \times n.L$  et chaque mot de code dépend uniquement du mot d'information présent en entrée du codeur. Pour un même rendement, il existe un ensemble de matrices génératrices pouvant générer un même code.

**Définition 1.1.** Deux codes sont équivalents si ils ont la même distribution de distances entre mots. Mathématiquement, deux matrices génératrices  $G$  et  $G'$  sont équivalentes s'il existe une matrice inversible quelconque  $A$  et une matrice de permutation  $B$  vérifiant  $G = A.G'.B$ . matrice de permutation.

La notion de codeur équivalent signifie que les séquences codées par deux codeurs équivalents appartiendront à la même famille ou au même code. Parmi l'ensemble des matrices génératrices d'un code, il existera toujours une forme systématique équivalente de la matrice génératrice.

**Définition 1.2.** Un code  $C(n, k)$  est un code systématique si tous les mots de code contiennent les  $k$  bits d'information non modifiés et les  $n - k$  bits restant sont appelés bits de parité.

D'après la définition 1.2, la matrice génératrice d'un code systématique peut être de la forme  $G = (I_k | P)$  ou  $G = (P | I_k)$ , avec  $I_k$  la matrice identité de taille  $k \times k$  et  $P$  une  $(k \times n - k)$  matrice.

**Exemple 1.1.** Prenons l'exemple d'un code de Hamming de paramètres  $n = 7, k = 4$

et de matrice génératrice :

$$G = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix} \quad (1.7)$$

Soit  $m(t) = (m_1(t) \ m_2(t) \ m_3(t) \ m_4(t))$ , un mot d'information composé de 4 bits et  $c(t) = (c_1(t) \ c_2(t) \ c_3(t) \ c_4(t) \ c_5(t) \ c_6(t) \ c_7(t))$  le mot de code de 7 bits qui en résulte.

D'après l'équation de codage (1.5), nous pouvons écrire que :

$$\begin{cases} c_1(t) = m_1(t) \\ c_2(t) = m_2(t) \\ c_3(t) = m_3(t) \\ c_4(t) = m_4(t) \\ c_5(t) = m_2(t) + m_3(t) + m_4(t) \\ c_6(t) = m_1(t) + m_3(t) + m_4(t) \\ c_7(t) = m_1(t) + m_2(t) + m_4(t) \end{cases}$$

La matrice génératrice de ce code étant systématique, un mot de code est un vecteur composé des 4 bits du mot d'information  $(m_1(t), m_2(t), m_3(t), m_4(t))$  et de 3 bits de parité  $(c_5(t), c_6(t), c_7(t))$ . En prenant le mot d'information  $m(t) = (1 \ 0 \ 1 \ 1)$ , le mot de code sera :

$$c(t) = (1 \ 0 \ 1 \ 1).G = (1 \ 0 \ 1 \ 1 \ 0 \ 1 \ 0) \quad (1.8)$$

En permutant l'ordre des lignes de la matrice génératrice  $G$ , il est possible d'obtenir une matrice génératrice équivalente  $G'$  :

$$G' = L.G = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix} \quad (1.9)$$

Un code  $C(n, k)$  permet d'engendrer  $2^k$  mots de code de  $n$  bits. Avec notre exemple, nous obtenons une famille de 16 mots de code de 7 bits. La figure 1.3



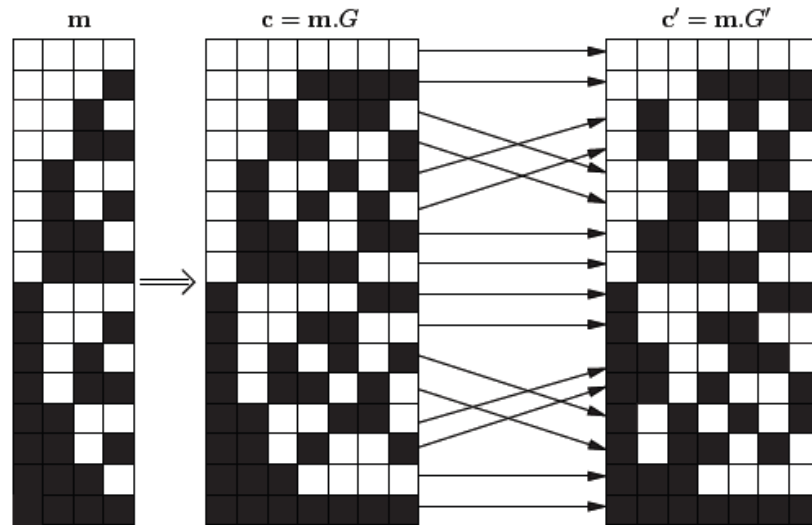


FIGURE 1.3 – Mots de code

représente les 16 mots d'information et les 16 mots de code obtenus avec la matrice génératrice  $G$  et ceux obtenus avec  $G'$ . Sur cette figure, les bits à "0" sont représentés par les cases blanches et ceux à "1" par les cases noires.

Tout d'abord, nous remarquons qu'un mot d'information codé par  $G$  et  $G'$  ne donne pas nécessairement le même mot de code. En revanche, en regardant tous les mots de code obtenus avec la matrice  $G$ , on remarque que l'on obtient les mêmes avec la matrice  $G'$  mais dans un ordre différent. Ces deux codes sont équivalents puisque leurs mots de code appartiennent à la même famille.

### 1.2.3 Les codes convolutifs

#### Présentation des codes convolutifs

Les codes convolutifs ont été introduit par Elias [22] en 1955, mais sans algorithme de décodage rapide et efficace ce type de code a été laissé de coté. Le premier algorithme capable de décoder de tels codes, appelé décodage séquentiel a été introduit par Wozencraft [23] en 1957 puis amélioré par Fano [24] en 1963. Il faudra encore attendre quatre ans pour qu'un algorithme de décodage particulièrement intéressant apparaisse, le décodage de Viterbi [25]. Dès lors, de nombreux travaux ont porté sur la construction des codeurs convolutifs.

L'enjeu de ces travaux étant d'obtenir des codes au pouvoir de correction le plus élevé possible. En 1970, puis en 1973, Forney [26] et [27] démontre les propriétés algébriques des bons codes convolutifs, soit des codes ayant de forts pouvoirs de correction. En parallèle avec ces avancées sur les propriétés des bons codes, un nouvel algorithme de décodage est apparu.

Tout comme les codes en blocs, le principe d'un code convolutif est d'ajouter de la redondance au message émis. Un mot d'information est également un vecteur de  $k$  bits et un mot de code un vecteur de  $n$  bits, mais nous allons introduire un nouveau paramètre, noté  $K$ , qui correspond à la longueur de contrainte du codeur. Chaque mot de code dépend du mot d'information présent en entrée du codeur, comme dans le cas d'un code en bloc, mais également des  $(K - 1)$  mots d'information ayant été introduits précédemment. De ce fait, un code convolutif peut être représenté, comme sur la figure 1.4, par un ensemble de registres à décalage, tel que chaque sortie du codeur est une combinaison linéaire des  $k \times K$  cellules du registre à décalage. La longueur de contrainte d'un codeur représente le nombre de blocs de  $k$  bits présents à l'intérieur du registre à décalage. Intuitivement, on peut dès à présent voir que de par la mémoire d'un code convolutif, ces codes engendrent des séquences codées de longueurs infinies, qui ne peuvent pas être traitées par paquets de bits disjoints comme dans le cas des codes en blocs.

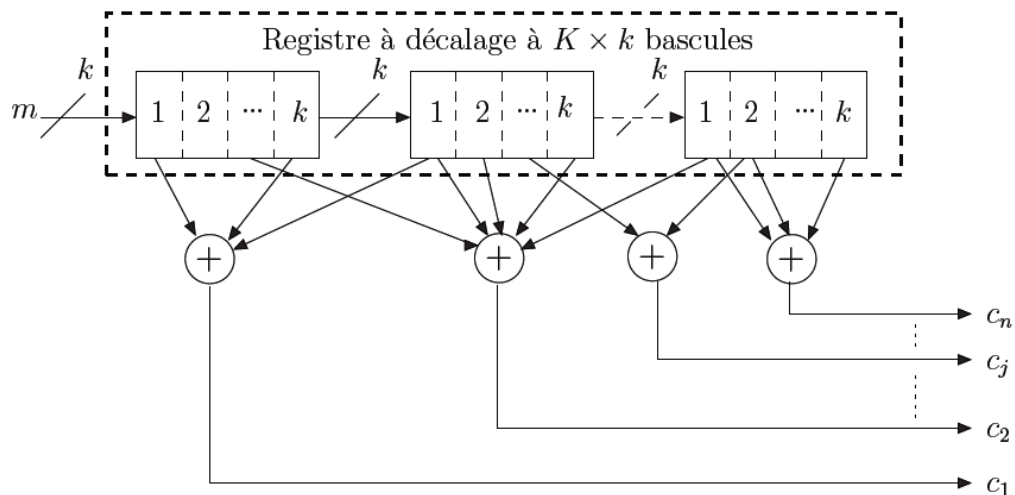


FIGURE 1.4 – Schéma d'implémentation d'un code convolutif à l'aide de registres à décalage

De même que les codes en blocs, il existe différentes familles de codes convolutifs. Ces différentes familles peuvent être distinguées par deux termes : systématique et récursif.

- Le terme systématique signifie que l'on retrouve les  $k$  bits d'entrée en sortie du codeur.

- Le terme récursif signifie que les sorties du codeur peuvent dépendre des sorties précédentes. Dans cette configuration, le codeur possède des registres avec rétroaction (rebouclage de la sortie en entrée).

Dans le cadre de notre étude, nous nous sommes intéressé aux cas des codeurs de types Non Récursifs et Non Systématiques, noté NRNSC, et des codeurs de types Récursifs et Systématiques, noté RSC. Ces deux types de codes étant les plus répandus dans les standards radio-mobiles.

Un code convolutif peut être entièrement défini par ses 3 paramètres entiers ( $n$ ,  $k$  et  $K$ ) et par sa matrice génératrice. La matrice génératrice d'un code convolutif est composée de  $k \times n$  polynômes générateurs qui sont soit de simples polynômes dans le cas d'un codeur de type NRNSC, soit des fractions rationnelles polynômiales dans le cas d'un RSC. Ces polynômes générateurs pourront également être représentés sous forme vectorielle ou encore en octal.

### Les polynômes générateurs

Notons  $a(D)$  un polynôme d'indéterminée en  $D$  de degré  $L$  défini par :

$$a(D) = \sum_{t \geq 0}^L a(t) \cdot D^t = a(0) + a(1) \cdot D + \dots + a(L) \cdot D^L \quad (1.10)$$

ou  $a(t)$  est le  $t$ -ième coefficient du polynôme. Alors, ce polynôme pourra également être décrit par un vecteur, noté  $a$ , qui sera composé des  $(L+1)$  coefficients du polynôme qui dans le cas présent sont des coefficients à valeur dans  $GF(2)$ , donc des éléments binaires :

$$a = (a(0) \ a(1) \ a(2) \ \dots \ a(l)) \quad (1.11)$$

Ce vecteur  $a$  pourra ensuite être converti en octal en regroupant les bits par paquets de trois.

**Exemple 1.6.** Conversion du nombre octal (542) en binaire :

$$\left\{ \begin{array}{l} 5 \Rightarrow (101) \\ 4 \Rightarrow (100) \\ 2 \Rightarrow (010) \end{array} \right. \quad (1.12)$$

Le nombre (542) en octal s'écrira (101100010) en binaire.

### 1.2.4 Code reed-solomon

Les codes de Reed-Solomon sont basés mathématiquement sur les corps finis de Galois. Les champs de Galois finis sont des ensembles d'éléments fermés sur eux-mêmes. Les codes de Reed - Solomon permettent de corriger des erreurs et des effacements grâce à des symboles de contrôle ajoutés après l'information.

#### Avantages des codes de reed-solomon

- Pour des corps de Galois  $GF(q)$ , lorsque l'on a besoin de coder des messages de longueur inférieure à  $q$ , l'encodage de RS est facilement utilisable. Puisque ce sont des codes MDS, la distance minimum est maximisée.
- Ils peuvent être combinés ou ajoutés à autres codes afin de réaliser des codes plus efficaces.
- L'encodage est assez facile.
- Ils sont très efficaces pour la correction d'erreurs consécutives, qu'ils soient utilisés seuls ou en conjonction avec d'autres codes. Bien entendu, cela est valide pour les cas où le nombre d'erreurs demeure en deçà de la capacité de détection-corrrection du code employé.
- Leurs algorithmes de décodages sont très développés.
- Pour des corps de Galois  $GF(q = 2^r)$ , on peut les représenter par des codes binaires puisque chacun des éléments de ce corps de Galois peuvent être représentés par une séquence binaire de longueur  $r$ .

**Propriétés** Nous poserons  $\alpha$  tel qu'étant un élément primitif dans  $GF(2^m)$ . Notez que l'on travaillera avec des codes  $RS$  ayant leurs symboles de  $GF(2^m)$ . Pour tout entier positif  $t$ , tel que  $t \leq 2m - 1$ , il existe un code RS en mesure de corriger  $t$  symboles, pour des symboles parmi  $GF(2^m)$ , et ayant les paramètres suivant :

$$n = 2^m - 1 \quad (1.13)$$

$$n - k = 2t \quad (1.14)$$

$$k = 2m - 1 - 2t \quad (1.15)$$

$$d_{min} = 2t + 1 = n - k + 1 \quad (1.16)$$

où  $n$  correspond au nombre de symbole d'un mot code,  $k$  correspond au nombre de symbole du segment de parité, et enfin, où  $d_{min}$  correspond à la distance minimale.

## Polynômes générateurs

### Polynôme générateur de mot code :

Générateur de mots code sous forme algébrique ( $g(x)$ ), sous représentation polynômiale, qui une fois multiplié par la représentation polynômiale du message à transmettre ( $m(x)$ ), génèrera le mot code correspondant au message ( $c(x)$ ). Ainsi nous avons donc la formulation suivante :

$$c(x) = g(x)m(x) \quad (1.18)$$

### Polynôme générateur pour RS :

L'équation (1.19) correspond au polynôme générateur de Reed-Solomon ainsi qu'à la manière de le construire. En effet, il n'y a qu'à prendre les produits des binômes pour  $(x + \alpha^i)$ , pour les  $i$  allant de 1 jusqu'à  $2t$ . On note que dans cette construction, les  $\alpha^i$  correspondront aux racines du polynôme  $g(x)$ , et où les  $g_i$  sont des symboles membre de  $GF(2^m)$ .

$$g(x) = (x + \alpha^1)(x + \alpha^2) \dots (x + \alpha^{2t}) \quad (1.19)$$

$$g(x) = g_0 + g_1x + g_1x^2 + \dots + g_{2t}x^{2t-1} + x^{2t} \quad (1.20)$$

**Encodage de codes de reed-solomon** Pour un message  $m(x) = m_0 + m_0x + \dots + m_{k-1}x^{k-1}$  où nous avons la représentation polynômiale d'un message à encoder. Nous avons les  $m_i$ ,  $GF(2^m)$  ainsi que  $k = n - 2t$ .

Nous devons d'abord multiplier  $m(x)$ , le message à transmettre, par  $x^{(2t)}$  procédant ainsi à un décallage. Ce décallage permettra d'insérer le polynôme de parité. Une fois que l'on a en main  $x^{(2t)}m(x)$ , nous procédons à sa division par le polynôme générateur de mots codes  $g(x)$  trouvé préalablement.

Nous avons donc maintenant l'équation 1.21, où  $b(x)$  représente le reste.

$$x^{2t}m(x) = a(x)g(x) + b(x) \quad (1.21)$$

$$b(x) = b_0 + b_1x + \dots + b_{2t-1}x^{2t-1} \quad (1.22)$$

Ce reste que l'on vient de trouver représente en fait le polynôme de parité recherché. Pour le message  $m(x)$ , le polynôme de parité est donc représenté par  $b(x)$ , et le message codé, soit le mot de code,  $c(x)$  correspondra à :

$$b(x) + x^{2t-1}m(x). \quad (1.23)$$

**Décodage de codes de reed-solomon** Toutes les versions d'algorithme principal de décodage des codes de Reed-Solomon sont similaires à l'énumération suivante. Ce sera suivi d'une version un peu plus mathématique. Enfin, dans les sections subséquentes nous reverrons le tout avec quelques détails supplémentaires.

1. Calculer les  $2t$  syndrômes  $s_0, s_1, \dots, s_{2t-1}$  où l'on a que les  $s_i = r(\alpha^i)$ , où  $r(x)$  est le polynôme reçu. Notez que  $s_0 = 1$ , et que le polynôme des syndrômes  $s(x) = s_0 + s_1x + \dots + s_{2t-1}x^{2t-1}$ .
2. Le rang de la matrice  $P$  nous donne le degré du polynôme localisateur d'erreur. La matrice  $P$  étant la matrice des coefficients lorsque l'équation du polynôme des syndrômes est vue matriciellement en relation avec les  $\lambda_i$ .
3. Évaluer le polynôme localisateur d'erreur à partir des informations en main.
4. Évaluer les racines du polynôme localisateur d'erreur. Ces racines, lorsque l'on a leur réciproque, nous donnent les positions des erreurs.

5. La matrice des positions d'erreurs par la matrice de leur intensité nous donnera les syndrômes, il nous faudra donc résoudre ce système matriciel pour obtenir l'intensité des erreurs.

Considerant le code  $c(x) = c_0 + c_1x + \dots + c_{n-1}x^{n-1}$ , où  $c_i \in GF(2^m)$  que l'on envoie dans un canal de communication, puis le code reçu  $r(x) = r_0 + r_1x + \dots + r_{n-1}x^{n-1}$ , où  $r_i \in GF(2^m)$ .

Le polynôme d'erreur est donc alors  $e(x) = r(x) - c(x)$  où  $e(x) = e_0 + e_1x + \dots + e_{n-1}x^{n-1}$  où  $e_i = r_i - c_i \in GF(2^m)$ .

Faisant l'hypothèse que  $r(x)$  comporte  $l$  erreurs, le polynôme  $e(x)$  comportera les erreurs aux locations  $x^{j_1}, x^{j_2}, \dots, x^{j_l}$ , et donc  $e(x) = e_{j_1}x^{j_1} + e_{j_2}x^{j_2} + \dots + e_{j_l}x^{j_l}$ .

Les positions des locations d'erreurs sont alors  $Z_{j_i} = \alpha^{j_i}$  pour des  $i$  allant de 1 à  $l$ . Les valeurs de ces erreurs sont de  $e_{j_i}$ , encore une fois pour des  $i$  allant de 1 à  $l$ . Les codes de Reed-Solomon ont donc en plus des procédures des codes BCH à déterminer la valeur des erreurs trouvées.

S'il y a  $p$  erreurs d'effacement et  $q$  erreurs dans le polynôme reçu  $r(x)$ , alors un decodeur  $RS(n, k)$  pourra corriger ces erreurs si on a  $2q + p \leq d - 1 = n - k$ . Pour un polynôme d'erreurs d'effacement  $e^*(x)$ , nous aurons donc en réception le polynôme  $r(x) = c(x) + e(x) + e^*(x)$ .

## 1.3 Modulation

L'opération de modulation consiste à faire varier un paramètre de l'onde porteuse qui est normalement une sinusoïde. Les paramètres modulés peuvent être l'amplitude, la phase ou la fréquence. Dans le cadre de ce projet nous considérons les modulations par déplacement de phase, les modulations par déplacement d'amplitude sur deux porteuses en quadrature et les modulations par déplacement d'amplitude et de phase.

### 1.3.1 Définition

Dans le schéma de la chaîne de transmission numérique présenté sur la figure 1.1, le codeur de source délivre des éléments binaires (ou bits)  $d_k$  toutes les  $T_b$  secondes. Le débit binaire de la source est défini par  $D_b = 1/T_b$  (bit/s). Les bits sont groupés en  $n$ -

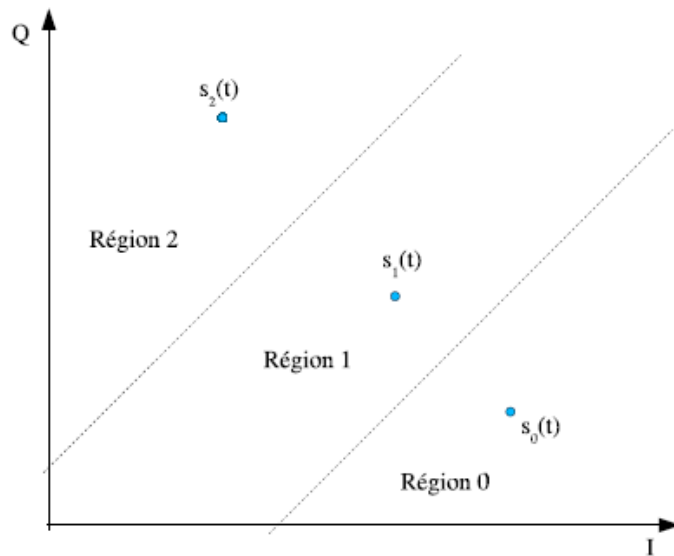


FIGURE 1.5 – Exemple de régions de décision pour une modulation à 3 points alignés tuples ou symboles pour affectés à des signaux de la modulation. Si la constellation est à  $M$  points (i.e. la modulation est d'ordre  $M$ ) :  $n = \log_2 M$ . La rapidité de la modulation, notée  $D_S$ , est définie comme le nombre de symboles émis par unité de temps. Elle s'exprime en bauds et vaut :

$$D_S = \frac{1}{T_s} = \frac{D_b}{\log_2 M} \quad (1.24)$$

où  $T_s$  est la durée d'émission d'un signal, encore appelé temps symbole. L'efficacité spectrale d'un système de communications est le débit utile transmis par unité de bande spectrale occupée, ou autrement dit, le nombre d'éléments binaires d'information transmis par unité de temps et par unité de bande occupée. Si l'on considère une modulation linéaire à  $M$  points et qu'un filtre de Nyquist idéal est utilisé pour la transmission, l'efficacité spectrale du système non codé, notée  $\eta$ , vaut :

$$\eta = \log_2 M (\text{bit/s/Hz}) \quad (1.25)$$

Si un code de rendement  $R$  est associé à la modulation, l'efficacité spectrale de l'association est exprimée comme suit :

$$\eta = R * \log_2 M (\text{bit/s/Hz}) \quad (1.26)$$



En tenant compte du fait que le spectre de fréquences est une ressource très demandée et en même temps très limitée, il est crucial de concevoir des systèmes de communications à très grande efficacité spectrale. On tente ainsi de maximiser la quantité d'information transmise en utilisant une largeur de bande fixe.

### 1.3.2 Modulations par déplacement de phase (MDP, en anglais PSK, Phase Shift Keying)

Dans ce type de modulation, des signaux de même fréquence et amplitude sont émis avec  $M$  phases différentes possibles pour désigner les  $M$  symboles. La constellation est constituée de  $M$  points déployés sur une circonférence. Il s'agit d'une modulation à enveloppe constante car tous les points de la constellation ont la même amplitude. Cette caractéristique la rend très intéressante pour des canaux non linéaires ou à évanouissements non connus car l'amplitude ne comporte pas d'information.

Une séquence de signaux modulés par déplacement de phase s'exprime sous la forme :

$$S_k(t) = A \sum_k h(t - kT_s) \cos(2\pi f_0 t + \phi_k + \phi_0) \quad (1.27)$$

où :

- $\phi_k$  prend ses valeurs de l'ensemble  $(2m + 1)/M$  avec  $0 \leq m \leq M - 1$ ,
- $\phi_0$  est une constante,
- $T_s$  est le temps symbole,
- $h(t)$  représente la fonction créneau (ou porte) :

$$h(t) = \begin{cases} 1 & \text{si } t \in [0, T_s[, \\ 0 & \text{si ailleurs.} \end{cases} \quad (1.28)$$

La figure 1.6 montre la constellation d'une modulation MDP-8. Tous les signaux ont la même énergie  $E_s$ , l'énergie moyenne de la modulation étant  $E_m = E_s$ . Le rayon du cercle où se placent les signaux vaut  $\sqrt{E_s}$ . La relation entre l'amplitude  $A$  et l'énergie

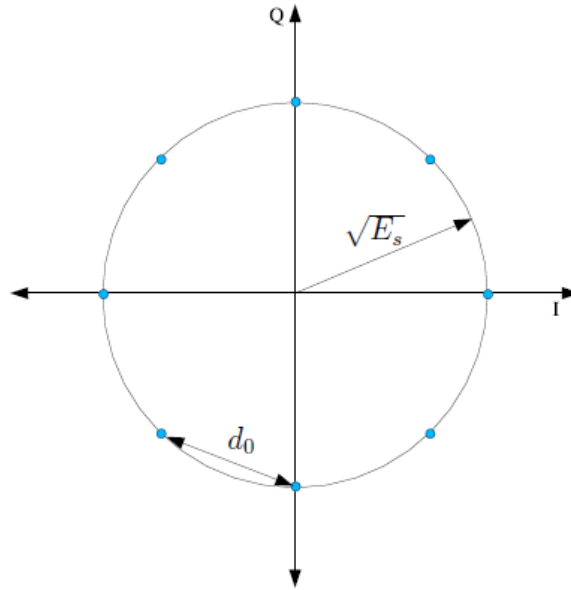


FIGURE 1.6 – Constellation de la modulation MDP-8

Es est la suivante :

$$A = \sqrt{\frac{2E_s}{T_s}} \quad (1.29)$$

La distance euclidienne minimale,  $d_0$ , s'exprime comme :

$$d_0 = 2\sqrt{E_s} \cos\left(\frac{\pi}{M}\right) \quad (1.30)$$

### 1.3.3 Modulation d'Amplitude sur deux porteuses en quadrature (MAQ, en anglais QAM, Quadrature Amplitude Modulation)

Une séquence de signaux modulés en amplitude sur deux porteuses en quadrature s'écrit sous la forme :

$$s(t) = \sum_k a_i h(t - kT_s) \cos(2\pi f_0 t) - \sum_k b_i h(t - kT_s) \sin(2\pi f_0 t) \quad (1.31)$$

Chaque signal dans la séquence est la somme de deux porteuses en quadrature, modulées en amplitude par les signaux  $\sum_k a_i h(t - kT_s)$  et  $\sum_k b_i h(t - kT_s)$ .

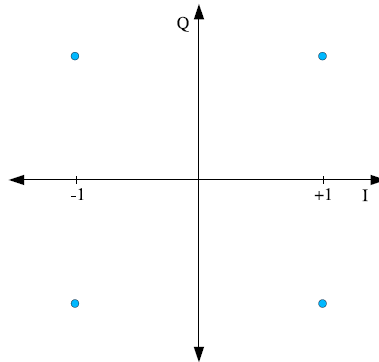


FIGURE 1.7 – Constellation de la modulation MAQ-4/MDP-4

Dans notre étude, les modulations MAQ utilisées sont telles que les symboles  $a_i$  et  $b_i$  sont indépendants et les alphabets ou ils prennent leurs valeurs sont identiques. Concrètement, nous considérons les modulations MAQ suivantes : MAQ-4/MDP-4 , MAQ-16, MAQ-64. Pour ces trois modulations  $n = \log_2 M$  est pair et les constellations correspondantes sont carrées. Par contre, le cas de la MAQ-32 est différent car  $n$  est impair et la constellation est dite en croix. Les figures 1.7 à 1.8 montrent les différentes constellations  $MAQ - 2^n$  pour  $n$  allant de 2 à 6.

Considérant la représentation sur une base orthonormale, les coordonnées des signaux dans le cas des constellations carrées sont de la forme  $(A_i, B_i) \in (\pm 1, \pm 3, \dots, \pm \log_2 M - 1)$  avec :

$$A_i = \sqrt{\frac{T_s}{2}} a_i \quad (1.32)$$

$$B_i = \sqrt{\frac{T_s}{2}} b_i \quad (1.33)$$

L'énergie moyenne de chaque modulation peut être calculée à partir de l'expression 1.33. Le tableau 1.1 fournit les valeurs d'énergie moyenne et de distance euclidienne minimale pour les différentes modulations MAQ. Ces résultats seront utilisés dans les chapitres suivants.

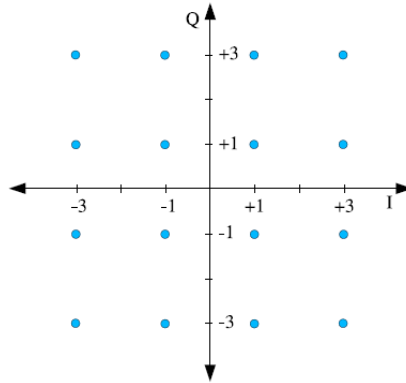


FIGURE 1.8 – Constellation de la modulation MAQ-16 (constellation carrée)

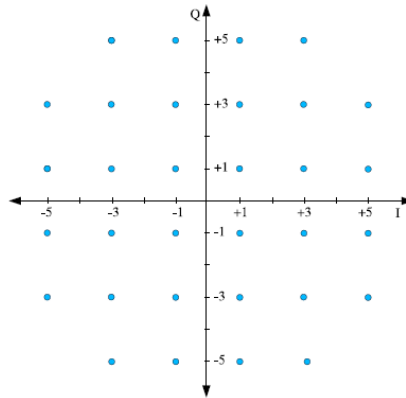


FIGURE 1.9 – Constellation de la modulation MAQ32 (constellation en croix)

	MAQ-4/MDP-4	MAQ-16	MAQ-32
$E_m$	2	10	20
$d_0$	$\sqrt{2E_m}$	$\sqrt{2/5E_m}$	$\sqrt{1/5E_m}$

TABLE 1.1 – Energie moyenne et distance euclidienne minimale des différentes modulations

### 1.3.4 Modulation par déplacement d'Amplitude et de phase (MDAP, en anglais APSK, Amplitude Phase Shift Keying)

Dans ce type de modulation, les  $M$  symboles sont distribués sur  $u$  cercles de rayons  $R_1, R_2, \dots, R_u$ , le nombre de symboles sur chaque cercle étant respectivement  $N_1, N_2, \dots, N_u$ . Logiquement,  $N_1 + N_2 + \dots + N_u = M$ . La notation utilisée pour nommer ces modula-

tions est  $MDAP - N_1 - N_2 - \dots - N_u$ . Une modulation MDAP peut être vue comme un ensemble de  $u$  modulations MDP sur  $u$  cercles différents. Les paramètres précisant les rapports entre ces MDPs sont :  $p_i = R_{i+1}/R_1$ , où  $R_i$  est le rayon du cercle  $i$ .  $\varphi_i$  : déplacement de phase relatif entre la constellation MDP du cercle  $i + 1$  et celle du cercle 1. Une séquence de signaux modulés suivant une modulation MDAP s'écrit sous la forme :

$$S(t) = \sum_k h(t - kT) \operatorname{Re} \{ s_k \exp(j2\pi f_0 t) \} \quad (1.34)$$

avec :

$$W_k = \begin{cases} R_1 \exp(j\varphi_i^1) , \varphi_i^1 = \frac{2\pi i}{N_1} & , i = 0, \dots, N_1 - 1, \text{ dans le cercle } R_1 \\ R_2 \exp(j\varphi_i^2) , \varphi_i^2 = \frac{2\pi i}{N_2} + \varphi_1 & , i = 0, \dots, N_2 - 1, \text{ dans le cercle } R_2 \\ \vdots \\ R_u \exp(j\varphi_i^u) , \varphi_i^u = \frac{2\pi i}{N_u} + \varphi_{u-1} & , i = 0, \dots, N_u - 1, \text{ dans le cercle } R_u \end{cases} \quad (1.35)$$

Les figures 1.9 et 1.10 montrent deux exemples de modulations MDAP. Dans la littérature, le cas de la MDAP-16 a été le plus étudié [29] [30] [31].

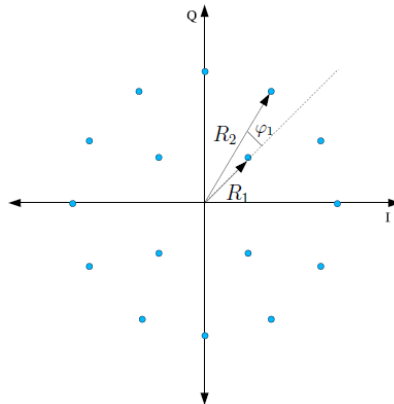


FIGURE 1.10 – Constellation de la modulation MDAP-4-12

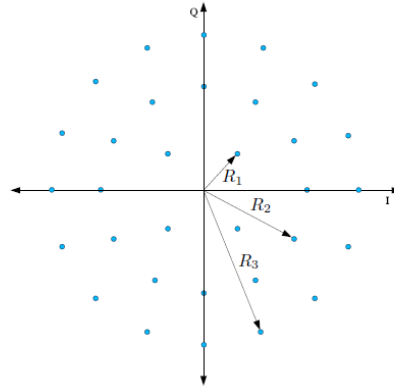


FIGURE 1.11 – Constellation de la modulation MDAP-4-12-16

## 1.4 Canal de transmission

Le canal de transmission, appelé également canal de propagation, représente le support physique qui permet de transmettre le signal. La forme physique du signal sera adaptée au milieu, par exemple : une onde électromagnétique pour le milieu ambiant, un signal électrique pour un câble, une onde lumineuse pour une fibre optique, etc. Ces milieux provoquent certaines modifications du signal que l'on y transmet, telles que des atténuations, des retards, des interférences, des distorsions, ...etc. Afin de modéliser au mieux ces différents phénomènes, de nombreux travaux de recherche ont porté sur la modélisation des canaux de propagation. Les canaux de transmissions peuvent être classés en deux groupes, les canaux stationnaires dont les paramètres sont fixes au cours du temps (fibres optiques, câbles métalliques, etc.) et les canaux non stationnaires dont les paramètres évoluent au cours du temps (communications radio-mobiles).

### 1.4.1 Le canal stationnaire

Les canaux stationnaires sont en règle générale modélisés par un canal appelé BBAG (pour Bruit Blanc Additif Gaussien). Ce canal, représenté sur la figure 1.12, est l'un des plus utilisés pour la simulation de transmissions numériques car il permet de représenter assez fidèlement les phénomènes physiques qui interviennent dans les canaux de transmission radioélectrique tout en gardant une simplicité de mise en oeuvre et d'analyse.

A chaque instant  $t$ , il modifie le symbole émis,  $s(t)$ , par l'addition d'un bruit blanc gaussien,  $n(t)$  :

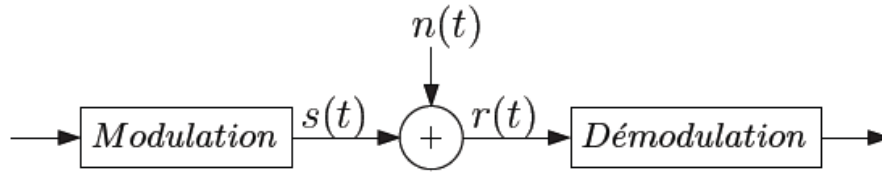


FIGURE 1.12 – Modélisation du canal BBAG

$$r(t) = s(t) + n(t) \quad (1.36)$$

Le BBAG,  $n(t)$ , est caractérisé par un processus aléatoire gaussien de moyenne nulle, de variance  $\sigma_n^2$  et de densité spectrale de puissance  $\frac{N_0}{2}$ .

### 1.4.2 Le canal non stationnaire

Lors d'une transmission radio-mobile, les mouvements de l'émetteur et du récepteur engendrent des fluctuations du canal. Ces fluctuations du canal vont entraîner des erreurs de transmission qui arriveront par rafales. De plus, dans ce type de canaux, l'environnement est souvent riche en échos ce qui se caractérise par de nombreux trajets multiples. La modélisation de ces types de canaux ne peut plus se faire avec un simple BBAG : il faut en effet utiliser des modèles plus élaborés qui tiendront compte des différences de propagation du milieu, appelés encore atténuations ou évanouissements. La sortie d'un canal à évanouissement comportant  $N$  trajets multiples s'exprime alors par :

$$r(t) = \sum_{n=0}^{N-1} \alpha_n \cdot s(t - \tau_n) + n(t) \quad (1.37)$$

ou  $n(t)$  représente le BBAG,  $\tau_n$  le retard affectant chaque trajet et  $\alpha_n$  l'atténuation complexe. L'atténuation  $\alpha_n$  est une variable aléatoire gaussienne qui suit une loi uniforme pour la phase et une loi de Rayleigh ou de Rice pour le module.

### 1.4.3 Le canal binaire

En sortie du canal de transmission, le signal bruité est démodulé afin d'obtenir une suite d'éléments binaires. Le taux d'erreur binaire, appelé couramment TEB ou BER en anglais (pour *BitErrorRate*) va permettre d'évaluer l'impact du bruit du canal sur le signal en comparant le message en entrée du modulateur et celui obtenu après démodulation. Il est possible de représenter l'ensemble des parties modulation, canal de transmission et démodulation par un canal binaire. Ce modèle de canal qui sera à entrée et à sortie ferme (c-à-d des suites de "0" et de "1") permettra d'analyser les performances du système (modulation, canal de transmission et démodulation). Le principe d'un canal binaire, représenté sur la figure 1.13, est d'associer à chaque bit d'entrée une certaine probabilité pour que le bit reçu soit erroné. Cette probabilité, qui représente le TEB, dépendra d'une part des erreurs engendrées par le canal de propagation et d'autre part des erreurs du à la démodulation du signal :

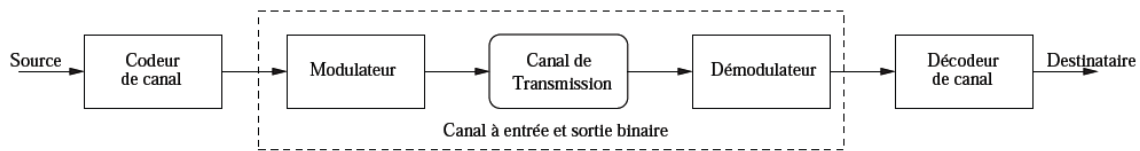


FIGURE 1.13 – Description d'un canal binaire

**Description d'un canal binaire** Le modèle le plus simple est le canal binaire symétrique appelé BSC (pour "*BinarySymmetricChannel*"). Un BSC est défini par sa probabilité d'erreur, notée  $p$ . La valeur de cette probabilité qui dépend du canal et de la modulation correspond au TEB obtenu en sortie du démodulateur. Si l'on note  $c$  et  $y$  les éléments en entrée et en sortie du BSC, alors la probabilité pour que le symbole reçu soit erroné sera égale à  $p$  et inversement la probabilité pour que le symbole reçu soit correcte sera de  $1 - p$  :

$$P_r(y = 0, c = 1) = P_r(y = 1, c = 0) = p \quad (1.38)$$

$$P_r(y = 0, c = 0) = P_r(y = 1, c = 1) = 1 - p \quad (1.39)$$

La figure 1.5 représente le fonctionnement d'un BSC. Chaque élément binaire en sortie du canal dépend uniquement de l'élément binaire d'entrée, on parle de canal sans mémoire.



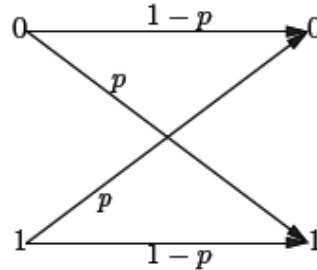


FIGURE 1.14 – Canal binaire symétrique

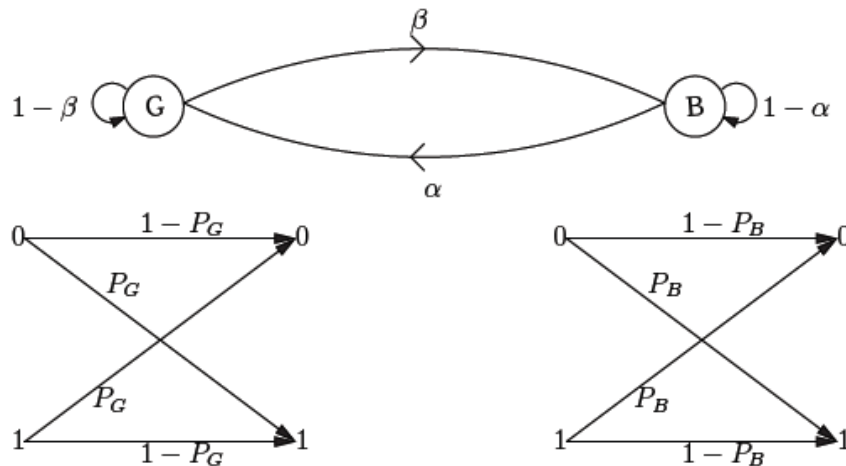


FIGURE 1.15 – Modèle du canal de Gilbert-Elliot

**Le canal burst** Pour certains canaux, tels que les canaux satellitaires, les conditions de propagation (fréquence de transmission, environnement,...) évoluent au cours de la transmission. Dans ce cas, il faut considérer un modèle de canal plus fin afin de représenter les erreurs qui arrivent par paquets. Le modèle de Gilbert-Elliot [32] permet de caractériser ces canaux appelés de type burst. Ce modèle de canal, représenté sur la figure 1.6 comprend deux états : l'état bon noté  $G$  et l'état mauvais noté  $B$ . L'état  $G$  correspond à une plage où les données seront très peu bruitées alors que l'état  $B$  représente une plage où les erreurs arriveront par rafale.

Les probabilités de transitions  $Pr(B \rightarrow G)$  et  $Pr(G \rightarrow B)$  sont respectivement  $\alpha$  et  $\beta$ . La probabilité d'erreur dans l'état  $B$  est  $P_B$  et celle dans l'état  $G$  est  $P_G$ . Le TEB moyen, noté  $p$ , est donné par :

$$p = \frac{1}{\alpha + \beta} [\alpha \cdot P_B + \beta \cdot P_G] \quad (1.40)$$

## 1.5 Conclusion

Dans ce chapitre, après avoir succinctement décrit les différents éléments d'une chaîne de transmission numérique, nous avons établi les bases théoriques permettant de décrire l'ensemble des fonctions concernées par notre étude. Nous avons ainsi présenté les notions fondamentales du codage correcteur d'erreurs et des modulations numériques car c'est l'association de ces deux fonctions qui permet de concevoir des systèmes à grande efficacité spectrale présentant de bonnes performances vis-à-vis du bruit. Une description des modèles de canal de transmission utilisés dans l'étude a également été donnée.

# Chapitre 2

## STRUCTURE DE LA CHAÎNE DE TRANSMISSION NUMÉRIQUE

### 2.1 Introduction

Un système de communications numériques comprend une source, un codeur de source, un codeur de canal, un modulateur, un canal de transmission, un démodulateur, un décodeur de canal, un décodeur de source et le destinataire. Le rôle de ces éléments est d'associer à un message émis par la source, un signal de nature compatible avec l'entrée du canal de transmission et de fournir au destinataire à partir du signal observé à la sortie du canal de transmission, un message aussi proche que possible du message émis par la source.

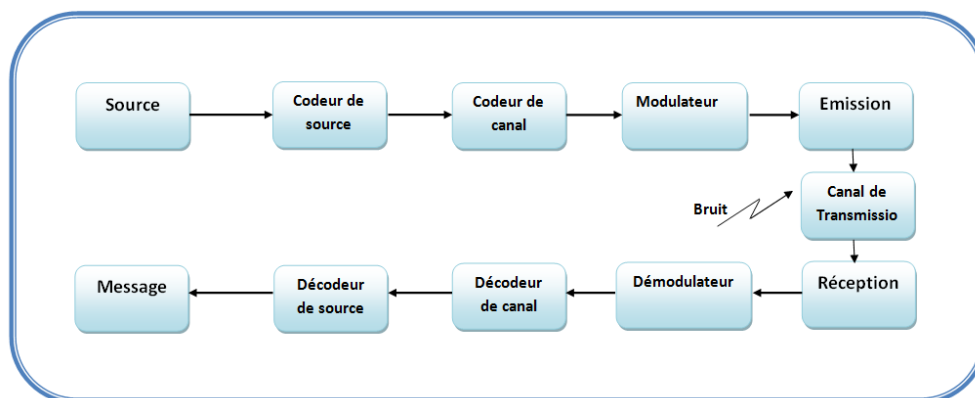


FIGURE 2.1 – Le système de communications numériques

## 2.2 L'émetteur

### 2.2.1 La source

Une source d'information génère des messages qui sont à transmettre au destinataire. La source délivre des symboles puisant leurs valeurs, dans un alphabet fini généralement binaire ne contenant que deux éléments notés, par convention 0 et 1. Si cet alphabet comporte  $M$  éléments, on parlera de source  $M$ -aire. La source est caractérisée par un débit alphabétique exprimé en symboles par seconde (ou Bauds) et un débit d'information en Shannon par seconde (ou bit/s) . Si une source délivre des symboles indépendants dont les valeurs sont équiprobables, on dira que la source est idéale dans le sens où elle maximise le débit d'information. Aussi une source binaire idéale, est caractérisée par un débit d'information égal au débit alphabétique ; Dans la suite on traitera le cas de la source binaire.

**Simulation** Dans le système de transmission y'a beaucoup de source d'information , dans le cadre de notre recherche on a utilisé le générateur bernoulli comme une source binaire.

**Bernoulli** Le bloc générateur de Bernoulli a été employé comme une source dans notre système pour générer des impulsions binaires aléatoires en utilisant une distribution de Bernoulli.

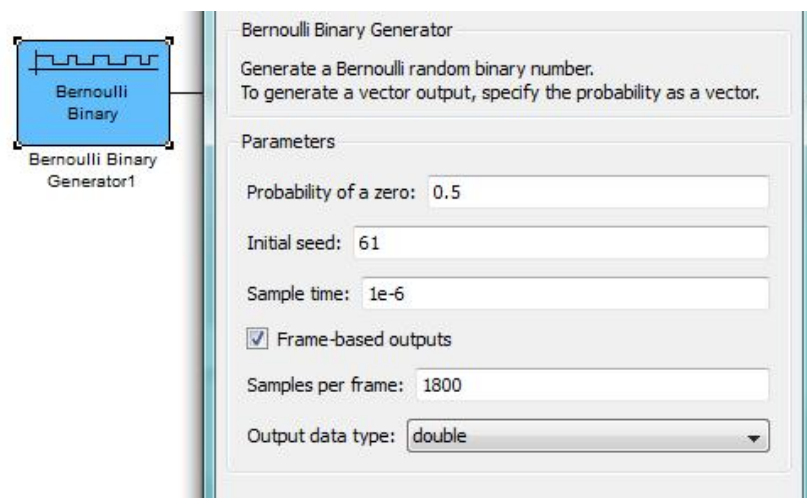


FIGURE 2.2 – Le bloc generateur Bernoulli

### 2.2.2 Codage canal

Le codage canal a pour rôle de protéger l'information émise contre les perturbations du canal de transmission susceptible de modifier son contenu. Il s'agit donc de rajouter de la redondance de manière à détecter et éventuellement corriger les erreurs lors de la réception si la stratégie adoptée le permet. L'information  $D(x)$  issue du codage source est transformée en séquence codée  $C(x)$ . Comme le décrit le théorème fondamental du codage canal, pour se rapprocher de la capacité du canal de transmission, il est nécessaire de coder l'information avant de la transmettre ; Les opérations de codage de canal ont pour objet de préparer le signal avant l'émission. Elles comprennent essentiellement la dispersion d'énergie (brassage), le codage correcteur d'erreur et l'entrelacement. Les signaux codés, des différents composants (sons, données et images) de plusieurs programmes, sont alors multiplexés. Le signal binaire résultant, transporte toutes les informations de synchronisation des différentes composantes des programmes. L'entrelacement est une opération dont le but est de rendre le signal le plus aléatoire possible, l'intérêt est d'éviter les longues suites des « zéros » ou des « uns » qui créent une raie à forte énergie dans le spectre. Pour le codage correcteur d'erreur, plusieurs algorithmes sont disponibles suivant l'application. Parmi les codes correcteurs d'erreur on trouve le codage **RS (Reed-SOLOMON)**, le codage **CC (codage convolutif)**.

#### FEC (Forward Error Correction)

Le codage du canal introduit de la redondance afin de rendre la transmission plus robuste. En effet, cette redondance pourra être utilisée en réception pour faire face aux différentes sources de bruit et d'interférence afin de détecter, voire corriger les erreurs commises. Afin de quantifier cette redondance, on définit le rendement du code, c'est à dire le rapport entre le nombre d'éléments binaires en entrée et le nombre d'éléments binaires en sortie ; Cette valeur est donc toujours inférieure ou égale à 1 (le cas particulier d'une valeur égale à 1, signifie qu'il n'y a pas de codage, c'est à dire aucune redondance). Il existe deux grandes familles de codes correcteurs d'erreurs ; les codes en bloc et les codes convolutifs [25]. Dans le cadre de notre étude on utilise un code en bloc de type **Reed-Solomon** suivi d'un code convolutif.

**Simulation** : pour effectuer cette simulation On a utilisé la technique de concaténation entre les codes **CC** et **RS** pour perfectionner cette dernière .

Dans notre projet nous avons utilisé les codes correcteurs d'erreurs suivants :

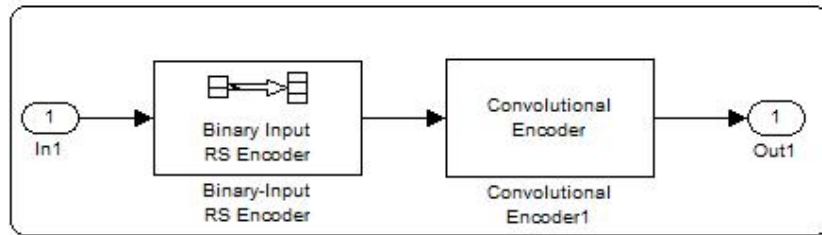


FIGURE 2.3 – Le bloc FEC

### Code reed-solomon

Le bloc Binary-input Encoder est un code détecteur et correcteur. Ce code est basé sur les corps de Galois dont le principe est de construire un polynôme formel à partir des symboles à transmettre et de le sur-échantillonner. Le résultat est alors envoyé au lieu des symboles originaux. La redondance du sur-échantillonnage permet au récepteur du message encodé de reconstruire le polynôme même s'il y a eu des erreurs pendant la transmission. Ces codes ont une propriété importante, ils sont linéaires et font partie des codes BCH. Le codeur prend  $k$  symboles de donnée (chaque symbole contenant  $s$  bits) et calcule les informations de contrôle pour construire  $n$  symboles, ce qui donne  $n-k$  symboles de contrôle. Le décodeur peut corriger au maximum  $t$  symboles, où  $2t = (n - k)$ .

**Simulation** Avec ce bloc on obtien une meilleur détection et correction d'erreur comise par le système.

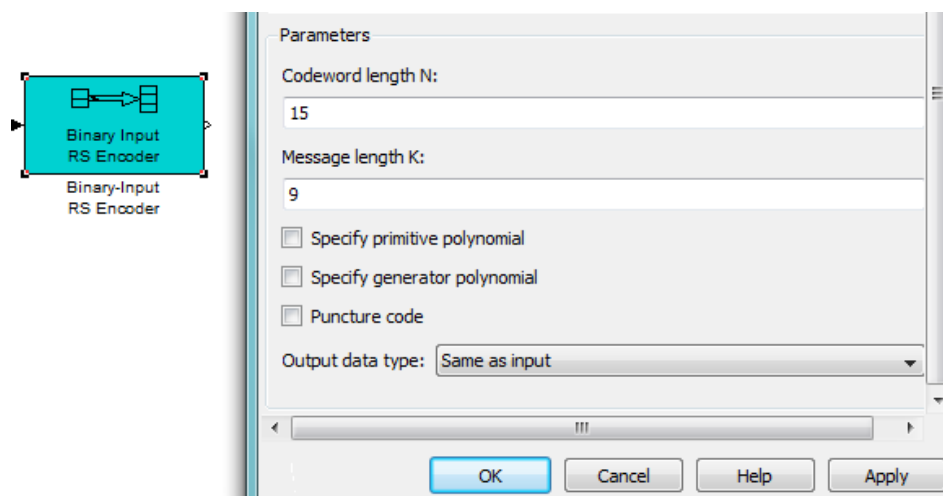


FIGURE 2.4 – Le bloc de codage reed-solomon

### Code convolutif

Après le processus du codage *RS*, les bits de données sont encore codés par un codeur convolutif binaire, qui a un taux de  $1/2$  et une longueur de contrainte égale à 7. Les polynômes générateurs utilisés pour dériver les bits du codage des deux sorties X et Y, sont donnés par les expressions suivantes :

$$G_1 = 171_{oct} \quad \text{pour } X, \quad (2.1)$$

$$G_2 = 133_{oct} \quad \text{pour } Y, \quad (2.2)$$

Un codeur convolutif, accepte des messages de longueur  $k_0$  bits et produit des mots de code de  $n_0$  bits. Généralement, il se compose d'un registre à décalage de L segments, où L est la longueur de contrainte du codeur.

pour faciliter l'algorithme de codage on prend un exemple d'un code convolutif non récursif non systématique de  $\mathbf{R} = 1/2$  et de polynôme générateur (7, 5) et schématisé par la figure 2-5 :

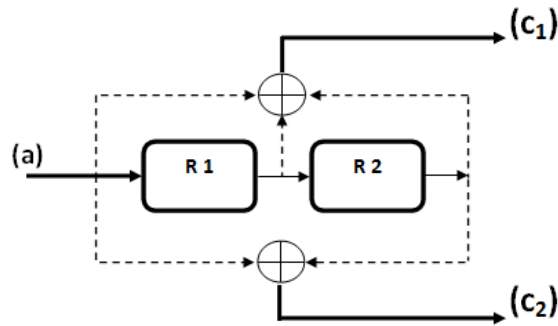


FIGURE 2.5 – Code convolutif NRNSC  $R = 1/2$ ,  $K = 3$

$$c_1 = (a) \oplus R_{(1)} \oplus R_{(2)} \quad (2.3)$$

$$c_2 = (a) \oplus R_{(2)} \quad (2.4)$$

Bits d'entrée (a)	Etat du registre R1 et R2	Bits de sortie (c <sub>1</sub> )	Bits de sortie (c <sub>2</sub> )
0	0 0	0	0
1	0 0	1	1
0	0 1	1	1
1	0 1	0	0
0	1 0	1	0
1	1 0	0	1
0	1 1	0	1
1	1 1	1	0

TABLE 2.1 – Table d'état du code convolutif NRNSC R=1/2,K=3

**Diagramme d'Etat** on déduit à partir du tableau 2-1 de l'exemple précédent le diagramme d'état dans lequel le temps n'intervient pas et qui représente les différents transitions possibles entre les états.

- Noeud : 1 état particulier des registres du codeur
- Branche : changement d'état du codeur en fonction de l'arrivée d'un nouveau d'un nouveau bit

Notre exemple : 2 registres à 4 état possibles (a=00 , b=01 ,c=10, d=11)

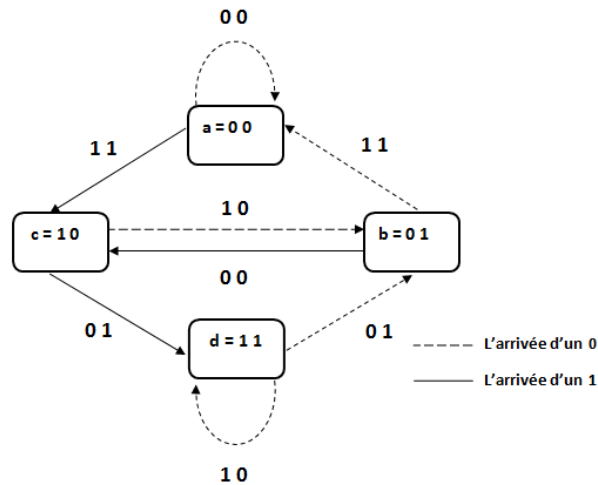


FIGURE 2.6 – Diagramme d'état d'un code convolutif



**Diagramme en treillis** cette représentation correspond à la façon dont communiquent les états entre eux en fonction du temps. sur la figure 2-7, les liens (branches) entre états représentés en pointillés et en traits pleins correspondent respectivement à la présence d'un bit d'information égale à **0** et **1**.

chaque de ces branches fournit un couple de bits pour la sortie du codeur un couple de bits pour la sortie du codeur le treillis est formé de noeuds reliés par des branches : les noeuds représentent les différents états possibles : il y en a  $q^{2^{(m-1)k}}$  s'il y a  $k$  entrées, les branches représentent les différents transitions possibles d'un noeud à un autre (ou d'un état du codeur au suivant) lors de l'arrivée d'un bit d'entrée.

la figure 2-6 présente le treillis du code précédent : les états sont **00,01,10,11**.

partant, par exemple de l'état **00**, l'arrivée d'un **0** mène le codeur à l'état **00** (transition en pointillé pour l'arrivée d'un **0**) et l'arrivée d'un **1** mène le codeur à l'état **10** (transition en trait plein pour l'arrivée d'un **1**).

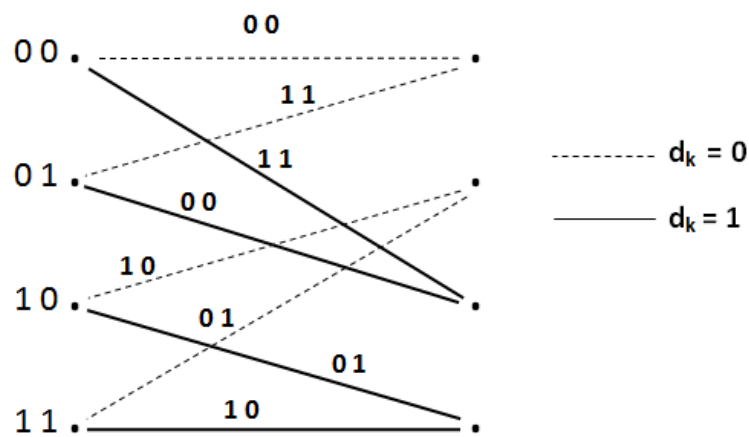


FIGURE 2.7 – Représentation en treillis d'un code convolutif

**Simulation** Dans cette étape on a implémenter un bloc détecteur correcteur d'erreur qui est défini le bloc ci-dessus

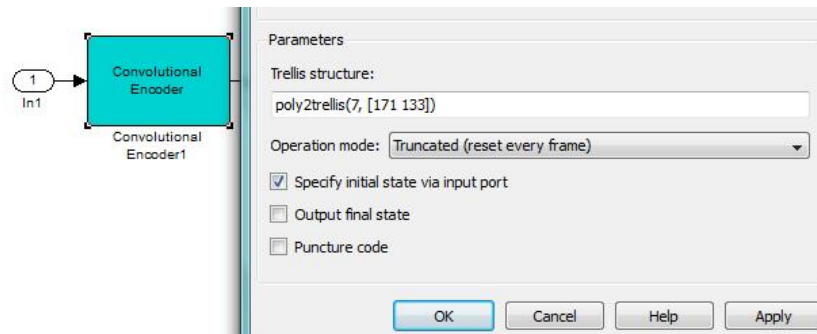


FIGURE 2.8 – Le bloc de codage convolutif

## Poinçonnage

Le processus de poinçonnage consiste à éliminer systématiquement, les bits du flux à faible taux de la sortie du codeur, afin de réduire la quantité de données à transmettre. Ceci conduit à un code à taux élevé. Les bits sont effacés selon une matrice de poinçonnage, où la moyenne du bit rejeté s'annule.

Le processus du poinçonnage est employé, pour créer les taux variables du codage requis pour fournir divers niveaux de protection d'erreur aux utilisateurs du système. Les différents taux qui peuvent être utilisés, sont  $1/2$ ,  $2/3$ ,  $3/4$  et  $5/6$ . Les vecteurs de poinçonnage pour ces taux, sont donnés dans le tableau 2-2 :

Taux de codage	Vecteur de poinçonnage
$1/2$	[1 1]
$2/3$	[ 1 0 1 1 ]
$3/4$	[1 0 1 1 1 0]
$5/6$	[1 0 1 0 1 0 1 0 1 1 ]

TABLE 2.2 – Vecteurs de poinçonnage

## Codes concaténés

La concaténation de codes permet d'augmenter la puissance des systèmes de codage au prix d'une augmentation de la complexité globale, autrement dit, moins complexe qu'un codeur "simple" mais de capacité équivalente. La concaténation peut se faire de trois façons : parallèle, série ou hybride (parallèle et série) et sur deux ou plusieurs niveaux [38]. Dans le cas d'une structure série à deux codes, l'information est codée deux fois. Une première fois par le premier code appelé code externe, puis une seconde fois par le deuxième, dit code interne [8]. Les deux codes utilisés sont en général complémentaires : les codes convolutifs par exemple sont inadaptés aux erreurs qui apparaissent par paquets alors que les codes *RS* sont adéquats pour ce type d'erreurs. Dans ce cas le décodeur convolutif s'occupera des erreurs aléatoires pour de faibles rapports signal sur bruit tandis que le décodeur *RS* s'occupera des erreurs par paquets pour des rapports signal sur bruit élevés. Ce type de concaténation série a été proposée par David Forney en 1966 [14] puis standardisé en 1987 pour les communications spatiales dans les réseaux DSN (Deep Space Network) par les agences spatiales NASA (National Aeronautics and Space Agency) et ESA (European Space Agency), préconisant l'utilisation d'un code *RS* ( $q = 3$  ;  $k = 9$  ;  $n = 15$  ;  $t = 6$ ). Il est aussi utilisé dans les systèmes de diffusion par satellite (norme DVB-S2) [39], la diffusion hertzienne terrestre (norme DVB-T) et la diffusion vidéo numérique portable DVB-H (Digital Video Broadcasting -Handhelds) qui est normalisée par le groupe européen ETSI (European Telecommunications Standards Institute [40]). Les codes produits, inventés par P. Elias en 1954 [41], sont construits par la concaténation série de deux ou plusieurs codes en blocs linéaires à faible pouvoir de correction. En général, les codes utilisés sont les codes BCH et Hamming.

### 2.2.3 Entrelacement

A l'émission, les octets d'un paquet sont répartis dans d'autres paquets, ce qui permet d'éviter d'avoir à corriger une longue suite de bits (ou octets) faux consécutifs. L'entrelacement permet de répartir les erreurs sur plusieurs paquets, facilitant ainsi la détection et la correction d'erreurs du décodeur *RS*. Le principe de l'entrelaceur convolutif utilise plusieurs registres à décalages qui vont induire un retard. L'entrelaceur utilise deux paramètres : le nombre de branches  $K$  et la profondeur  $T$  (en octet) du registre à décalage de base.

**Simulation** pour la protection d'information de notre chaîne il faut une technique d'entrelacement

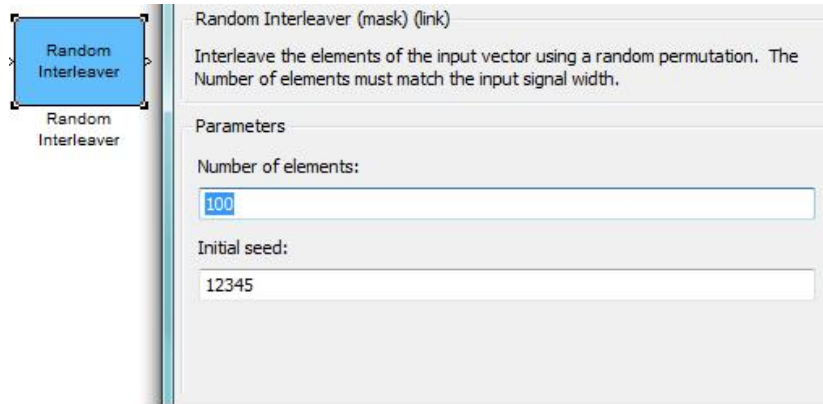


FIGURE 2.9 – Le bloc d’entrelacement

## 2.2.4 Modulation

Pour des raisons techniques ou de rentabilité, l’acheminement d’une information numérique, ne peut pas toujours se faire en bande de base. L’utilisation d’une fréquence porteuse est alors nécessaire. La modulation est l’opération qui fait correspondre à chaque niveau du signal numérique, un état d’amplitude, de fréquence ou de phase d’une onde porteuse. Le choix d’une modulation numérique dépend de : La modulation agit sur les paramètres d’un signal porteur afin de transmettre les données codées. Dans le cas de la modulation numérique, le message codé est transformé à partir d’un alphabet dont l’entrée correspond à une partie du signal à transmettre (i.e. un symbole). Le signal porteur est une sinusoïde dont on peut faire varier l’amplitude, la fréquence ou la phase indépendamment (ASK (Amplitude Shift Keying), FSK (Frequency Shift Keying), PSK (Phase Shift Keying)) ou simultanément QAM (Quadrature Amplitude Modulation), en fonction de l’information à émettre.

**Simulation** Après l’entrelacement des données binaires, les données rentrent au bloc de modulation, notre système définit l’utilisation de plusieurs modulations : BPSK, QPSK, QAM, quelques valeurs de constellation mapping sont données par le tableau 2-3.

modulation	Constellation mapping
BPSK	[0 1]
QPSK	[0 2 3 1]
16QAM	[13 12 14 15 9 8 10 11 1 0 2 3 5 4 6 7]

TABLE 2.3 – Vecteurs de poinçonnage

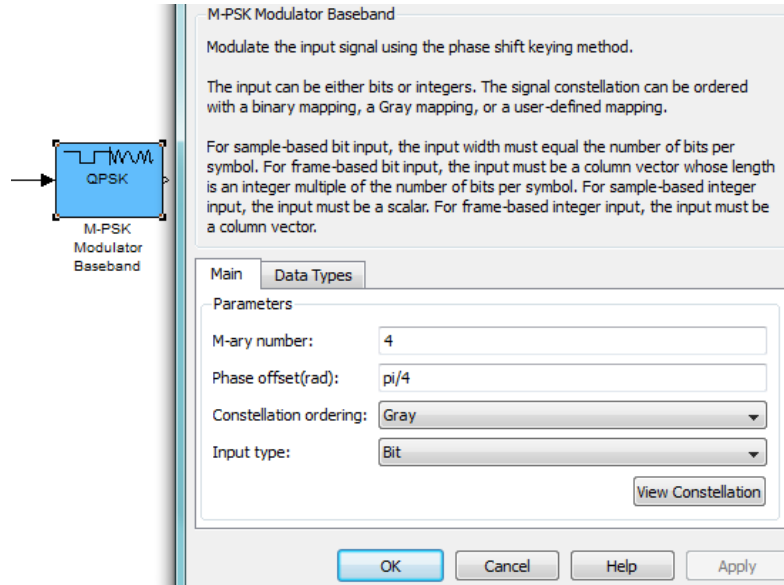


FIGURE 2.10 – LE bloc de modulation QPSK

## La modulation QAM

La modulation QAM (Quadrature Amplitude Modulation) ou modulation d'amplitude en quadrature de phase est une technique qui emploie une combinaison de modulation de phase et d'amplitude. Elle est largement employée par les modems pour leur permettre d'offrir des débits binaires élevés. Prenons par exemple un signal modulé QAM avec 3 bits transmis par baud. Une telle modulation requiert donc 23 soit 8 combinaisons binaires différentes. Dans notre exemple nous prendrons 2 amplitudes combinées avec 4 décalages de phase différents. La table de correspondance pourra être du type : Après l'entrelaceur, les bits de données entrent en série dans le modulateur numérique. Tous les systèmes de transmission sans fil, utilisent une méthode de modulation pour représenter les bits codés sous une certaine forme qui peut être, effectivement transmise sur un canal de transmission. Ainsi, les bits des sous porteuses, sont représentés par une amplitude et une phase, données par un vecteur complexe en phase et en quadrature de phase (Q.I.). La représentation graphique de Q.I, montre le vecteur transmis pour toutes les combinaisons de mots contenant les données de la modulation. Le codage de Gray, est une methode pertinente pour cette allocation, puisque les points adjacents dans la constellation, différent seulement par un simple bit. Ce codage aide à réduire au minimum, le taux global d'erreur des bits comme il réduit les erreurs multiples occurrentes, d'une erreur simple des symboles.

**Simulation** Dans notre chaîne on a utilisé la modulation QAM

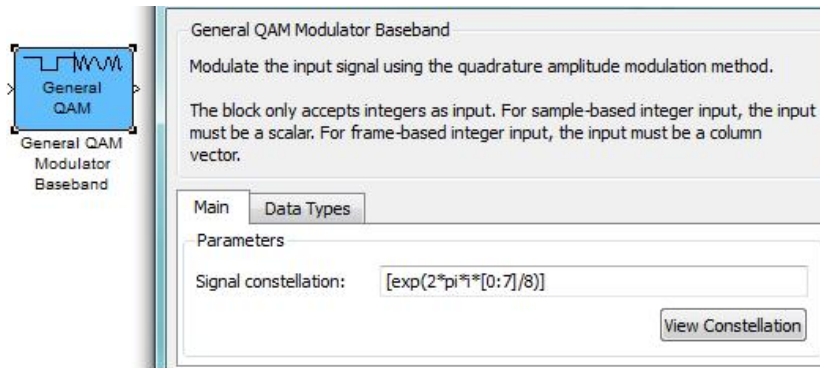


FIGURE 2.11 – Le bloc de modulation QAM

## 2.3 Canal de transmission

Il représente la liaison entre l'émetteur et le récepteur et peut être de différentes natures selon le type de grandeur qu'il permet de véhiculer. Le canal de transmission est caractérisé par sa capacité et sa bande passante. Il existe plusieurs modèles théoriques du canal de transmission en fonction des types d'erreurs les plus fréquents, nous nous limiterons au canal à Bruit Blanc Additif Gaussien (BBAG) et à un canal Rayleigh.

### 2.3.1 Canal à bruit blanc additif gaussien (Canal AWGN)

Le canal à Bruit Blanc Additif Gaussien (BBAG ou AWGN, Additive White Gaussian Noise) est le modèle le plus fréquemment utilisé pour caractériser un canal de transmission. Dans ce modèle, le signal reçu  $r(t)$  est la somme du signal transmis  $s(t)$  et d'un bruit gaussien  $n(t)$  :  $r(t) = s(t) + n(t)$ . Cette modélisation de canal a une grande importance théorique et pratique. Elle constitue en effet un modèle très précis pour certains types de canaux de transmission, pour les communications satellites et spatiales en particulier. Le bruit additif gaussien est la modélisation d'un bruit dont les sources sont internes ou externes. Les sources de bruit internes sont générées à l'intérieur des composants électroniques des équipements du système. Dans la majorité des cas c'est la source la plus significative et la plus facile à caractériser. Le bruit thermique est la plus importante d'entre elles. Les sources de bruit externes sont générées à l'extérieur du système, causées par les intempéries, le bruit industriel, etc. Elles sont difficiles à

modéliser ; Calcul de la variance du bruit gaussien en fonction du rapport signal bruit  
 Le bruit additif gaussien est caractérisé par un processus aléatoire gaussien centré de densité spectrale de puissance bilatérale égale à  $N_0/2$ .

**Simulation** pour transmettre une information d'un émetteur a un destinataire il nous faut un canal de transmission .

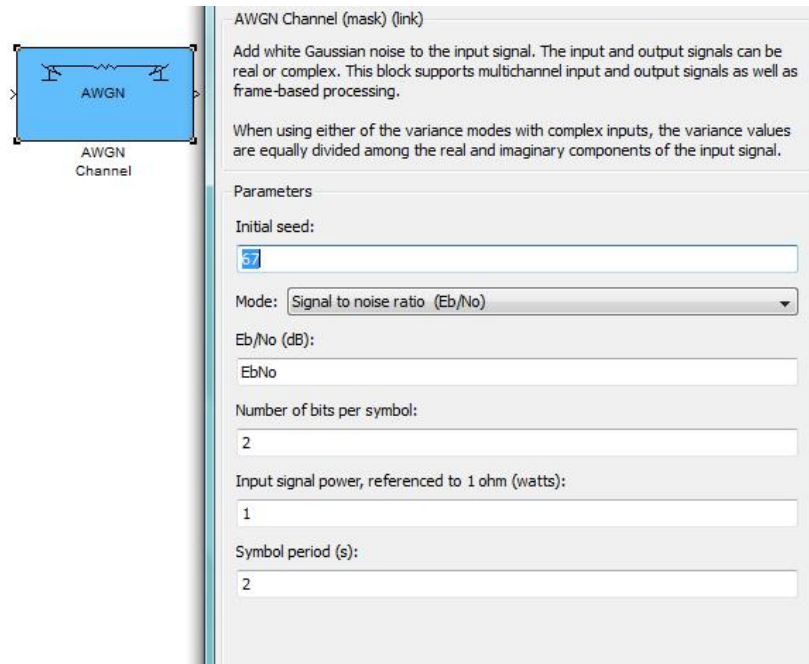


FIGURE 2.12 – LE bloc de canal de transmission AWGN

### 2.3.2 Canal de Rayleigh

Ce modèle de canal fait l'hypothèse simplificatrice que le signal reçu par le récepteur est composé d'un grand nombre de trajets. D'autre part, ces derniers sont non corrélés 0 et  $2\pi$ . on peut alors, d'après le théorème de la limite centrale, supposer que les composantes en phase ( $I$ ) et en quadrature ( $Q$ ) de l'enveloppe du signal reçu suivent deux lois gaussiennes indépendantes et corrélées temporellement .

Des lors , on en déduit que l'enveloppe  $r(t)$  du signal suit une loi de Rayleigh.

Dans le modèle de Rayleigh , la moyenne de la réponse impulsionnelle du canal  $h(t)$  est considérée comme nulle . IL est à noter que la valeur quadrature définit la puissance moyenne reçue par le récepteur après la prise en compte des phénomènes d'affaiblissement et d'ombrage.



FIGURE 2.13 – Le bloc de canal Rayleigh

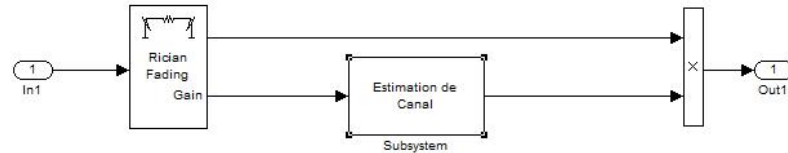


FIGURE 2.14 – Le bloc de canal rayleigh

## 2.4 Recepteur

### 2.4.1 Démodulateur

La démodulation est l'opération inverse de la modulation, on utilise pour cela la fonction inverse du PSK-MOD (PSK-DEMOM. Cette étape n'a été effectuée que pour comparer le décodage hard (avec démodulation préalable) et le décodage soft (sans démodulation préalable). Le démodulateur joue le rôle dual du modulateur et transforme donc le signal reçu en un train binaire. Avec les nouvelles techniques (notamment avec l'étalement de spectre) on module souvent plusieurs caractéristiques en même temps .

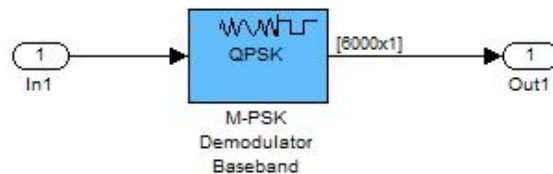


FIGURE 2.15 – Le bloc de demodulation QPSK



## 2.4.2 Désentrelacement

Le désentrelacement réarrange les bits de chaque salve, d'une manière correcte par ordonnancement consécutive, comme avant le procédé d'entrelacement. Il se compose de deux blocs ; un bloc général de désentrelacement et d'un bloc de désentrelacement de matrice. Ces blocs fonctionnent identiquement comme l'entrelaceur utilisé dans l'émission.

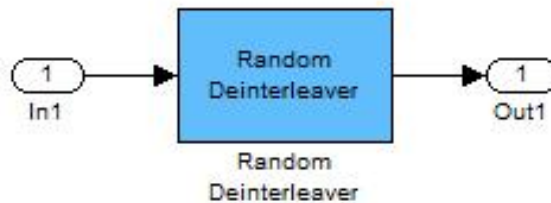


FIGURE 2.16 – Le bloc de désentrelacement

## 2.4.3 Décodage de canal

le décodage canal consiste dans un premier temps à détecter la présence d'erreurs dans l'information et puis dans un deuxième temps de les corriger. Les codes correcteurs d'erreurs ont été utilisés pour la détection et la correction des erreurs induites par le canal la transmission. Différentes types de codes ont été utilisés ; parmi ceux on distingue les codes en blocs et les codes convolutifs. Pour un code en blocs, la trame d'entrée de  $k$  symboles d'information est convertie en une séquence de sortie sur  $n$  symboles (avec  $k < n$ ), Le bloc de sortie sur  $n$  symboles dépend uniquement des  $k$  symboles de la trame d'entrée et du rendement de code  $r = k/n$ . Par contre, le codeur d'un code convolutif a un effet mémoire et prend pour entrée un symbole de  $k$  bits et fournit en sortie un symbole de  $n$  bits. Les  $n$  bits en sortie sont calculés par une combinaison linéaire entre les  $k$  bits en entrée et les  $m$  blocs mémorisés (précédents). Le rendement du code convolutif  $r = k/n$  et sa longueur  $K_c$  est le nombre maximum de bits associés à une sortie qui peuvent être affectés par un bit quelconque à l'entrée. Par la suite, nous nous sommes intéressés au cas du code de rendement  $r = 1/n$ .

## Decodeur viterbi

ce bloc est basé sur le principe du maximum de vraisemblance. Cet algorithme est une méthode optimale de décodage pour les codes convolutifs, ses performances dépendant de la qualité du canal de transmission utilisé. Par contre la complexité des systèmes de décodage augmentant exponentiellement avec la longueur de contrainte du code utilisé restreint leur emploi aux applications où le code a une petite longueur de contrainte. Il est utilisé dans les applications des systèmes sans fil

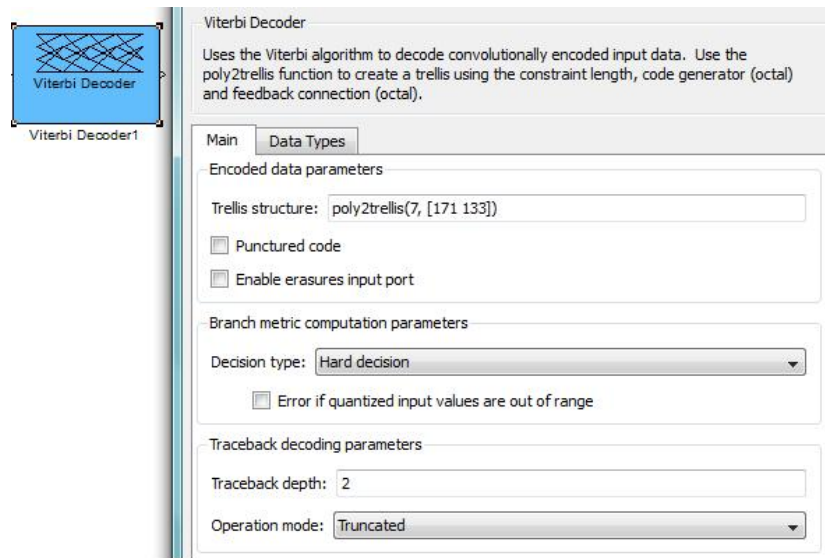


FIGURE 2.17 – Le bloc de décodeur viterbi

## Décodeur reed solomon

Binary-output RS Decoder : Comme pour le décodage des codes BCH binaires, le décodage revient à résoudre l'équation-clé 1.20. Dans ce cas, la résolution de cette équation nécessite quatre étapes élémentaires : le calcul des composantes du syndrome  $S$  ; le calcul du polynôme localisateur d'erreur ( $x$ ) et du polynôme évaluateur d'erreur ( $x$ ) ; la détermination des positions des erreurs ; le calcul des amplitudes des erreurs. La valeur de l'erreur est fournie sur  $q$  bits, il s'agit de la correction à apporter au symbole erroné. Classiquement, la correction des codes RS intervient sur un ou plusieurs octets ( $t = 8$  et  $GF(28)$ ). Les processus de décodage utilisent les mêmes algorithmes tels que celui de Berlekamp pour calculer le polynôme localisateur d'erreurs et de Chien pour déterminer les racines. L'algorithme PGZ est le plus adapté pour ces faibles pouvoirs de correction (un ou deux symboles). Les symboles d'un code Reed-Solomon étant  $q$ -aires,

la localisation des erreurs n'est pas suffisante comme dans le cas des codes BCH. Il est nécessaire de pouvoir estimer également leurs amplitudes afin d'effectuer les corrections associées. Un autre algorithme similaire à Berlekamp c'est l'algorithme d'Euclide. C'est un algorithme récursif qui permet de localiser les erreurs et de calculer leurs amplitudes, il est basé sur le calcul d'un PGCD (plus grand diviseur commun) entre deux polynômes  $r_0(t)$  et  $r_1(t)$  qui sont initialisés à  $x_2(t)$  et  $S(x)$  respectivement. Une suite de divisions successives est réalisée dans le «champ de Galois»  $GF(q)$  jusqu'à ce que le degré du reste soit inférieur à  $t$ . La procédure de calcul du polynôme de localisation des erreurs et le polynôme d'amplitude selon l'algorithme d'Euclide se trouvent dans [208].

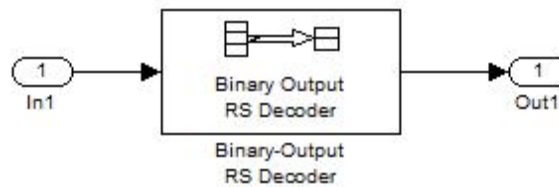


FIGURE 2.18 – Le bloc de décodeur reed-solomn

#### 2.4.4 Décodeur source

Le décodage source réalise l'opération duale, le message d'information est décomprimé afin de retrouver son équivalent à partir de la séquence de substitution «  $D(x)$  » avant la transmission. Il est à noter que les limites théoriques du codage source sont fixées par le premier théorème de Shannon.

#### 2.4.5 Bits reçus

En fin de chaîne s'effectue la réception des données mais on peut aussi calculer le TEB final de la chaîne de transmission

### 2.5 Conclusion

Dans ce chapitre, on a bien défini les différents blocs de la chaîne de transmission ; cette chaîne est devenue de plus en plus performante avec l'utilisation des techniques modernes de codage comme les codes convolutifs, en blocs dans le codage canal et les

différents types de modulation. Nous avons exposé dans ce chapitre quelques notions sur les codes modernes qui sont plus performants que les codes classiques. Pour cela, nous réalisons dans le prochain chapitre une simulation qui implémente la chaîne avec une modulation sans codage et une autre chaîne avec un code convolutif simple et une troisième chaîne avec un code concaténé d'un code convolutif avec Reed-Solomon et voir par la suite l'influence de ces dernières sur la qualité de transmission en termes de probabilité.

# Chapitre 3

## SIMULATIONS ET PERFORMANCES

### 3.1 Introduction

La simulation de systèmes de communications est un moyen efficace et rapide pour mettre en lumière les performances et les principales difficultés de conception de ces derniers. Parmi l'ensemble des logiciels de simulation, nous avons utilisé l'outil de simulation appelé Matlab/Simulink permet de modéliser des chaînes de transmissions, pour en analyser les performances en termes de taux d'erreur binaire (*TEB*). C'est ainsi qu'on propose de modéliser et de simuler, sous Matlab/Simulink, une chaîne de transmission, dans le but d'illustrer les notions abordées en communications numériques, tel que la modulation, le codage de canal, ainsi que l'Entrelacement, etc... . La démarche consiste à implémenter, en plusieurs étapes ; dans la première étape on implémente une chaîne de transmission simple sans codage (source-canal, modulation, destinataire), en traçant le TEB en fonction de  $E_b/N_0$  pour les différentes modulations et canaux (AWGN, Rayleigh) en seconde étape on essaye d'améliorer les performances de la chaîne en ajoutant les différentes techniques de codages (CC, RS et concaténation CC-RS) pour les différents canaux et modulations. En dernière étape nous allons tenter d'améliorer grâce à la technique d'entrelacement les performances du système. La chaîne doit être validée et ses performances doivent être comparées avec la chaîne de l'étape précédente afin de mettre en évidence l'apport des blocs ajoutés. Les résultats de simulation du système sont comparés avec les différents types de modulation de sous porteuses et de codage. À l'aide de la simulation, nous allons tester et valider nos différents choix d'architecture. Ce qui nous permettra de mettre en évidence l'amélioration

des performances du système que la concaténation des deux codes peut apporter selon l'algorithme de Viterbi.

## 3.2 Implementation de La chaîne de transmission

Après avoir définis les différents blocs de la chaîne de transmission lors de chapitre 2, nous pouvons expliciter l'implémentation et la simulation de notre chaîne de transmission sur simulink de MATLAB . voir la figure 3.1

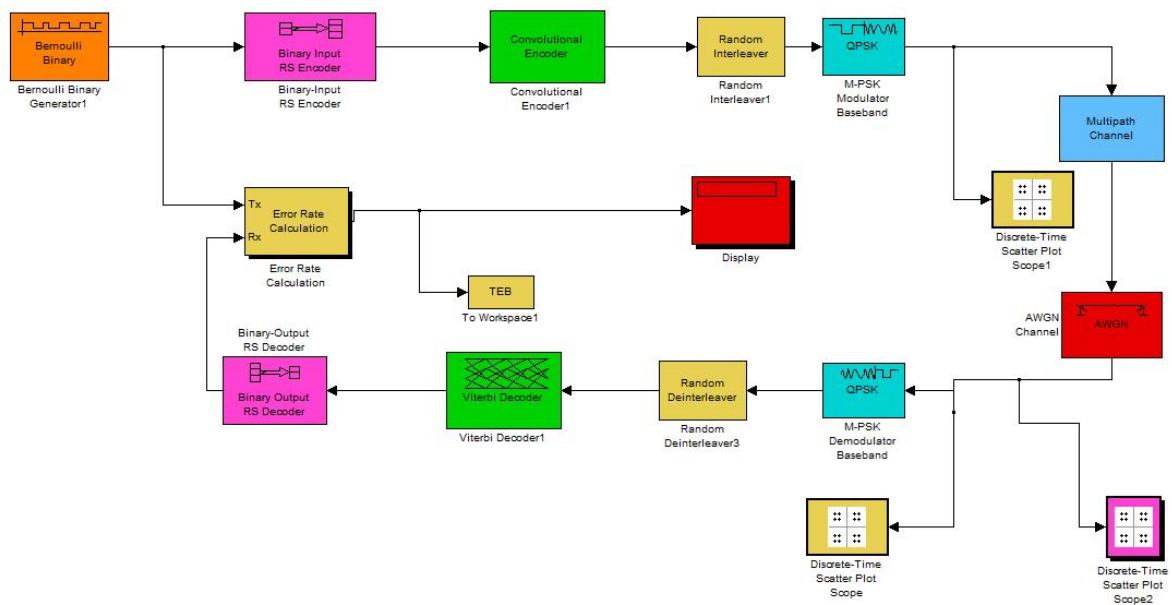


FIGURE 3.1 – Schéma global de la chaîne de transmission

## 3.3 Simulation de la chaîne de transmission sans codage

En premier temps nous avons simulé une chaîne de transmission comportant une modulation et démodulation (M-PSK) autour d'un canal AWGN et un canal Rayleigh cette chaîne peut être vue comme une chaîne référence afin d'évaluer les performances (entre de TEB) apportées par la suite. Des blocs seront mis en parallèle du dispositif d'émission-réception afin de calculer le nombre de bits reçus et celui des bits défectueux. D'autre part, des dispositifs de visualisation nous permettront pendant la simulation de voir le signal reçu en même temps .

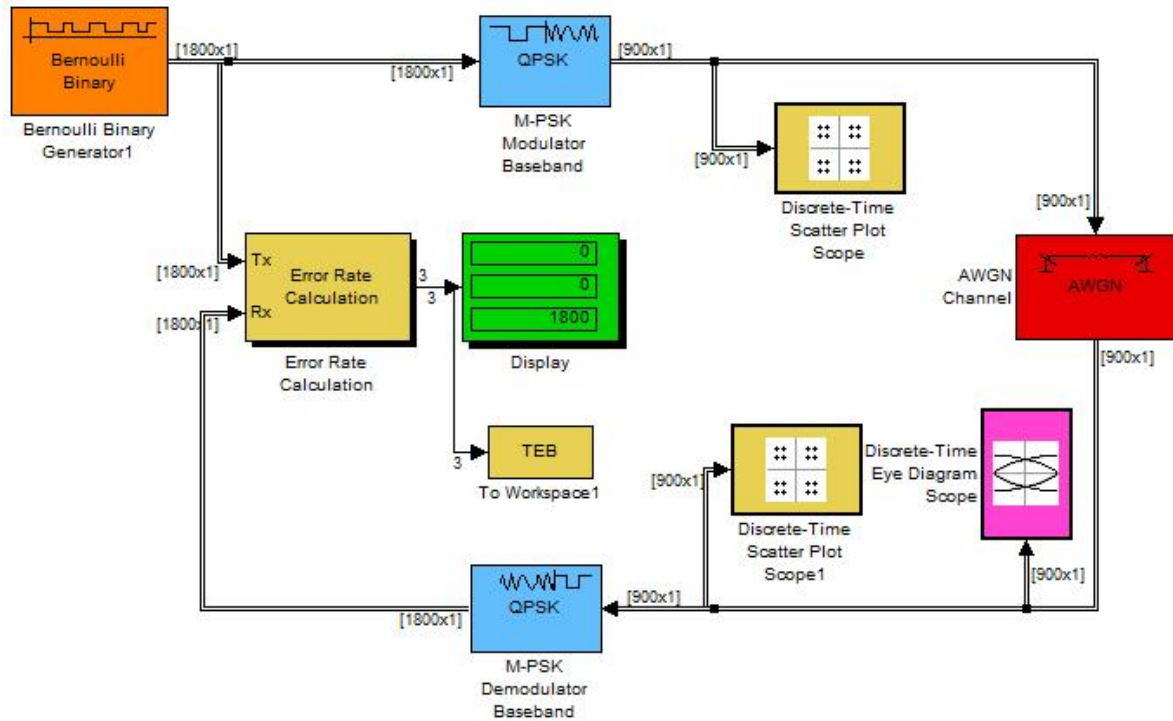


FIGURE 3.2 – Modélisation de la chaîne de transmission sans codage

pour ce faire ,nous avons représenté :

- Les diagrammes de constellations avant et après le canal de transmission
- La courbe du TEB en fonction du  $E_b/N_0$  et c'est ce qui nous servira de référence par la suite

Pour évaluer les performances de notre chaîne de transmission, il est important de déterminer les diagrammes de constellation des modulations. Les diagrammes de constellation des modulations que nous avons utilisés sont décrits sur les figures ci-dessus :

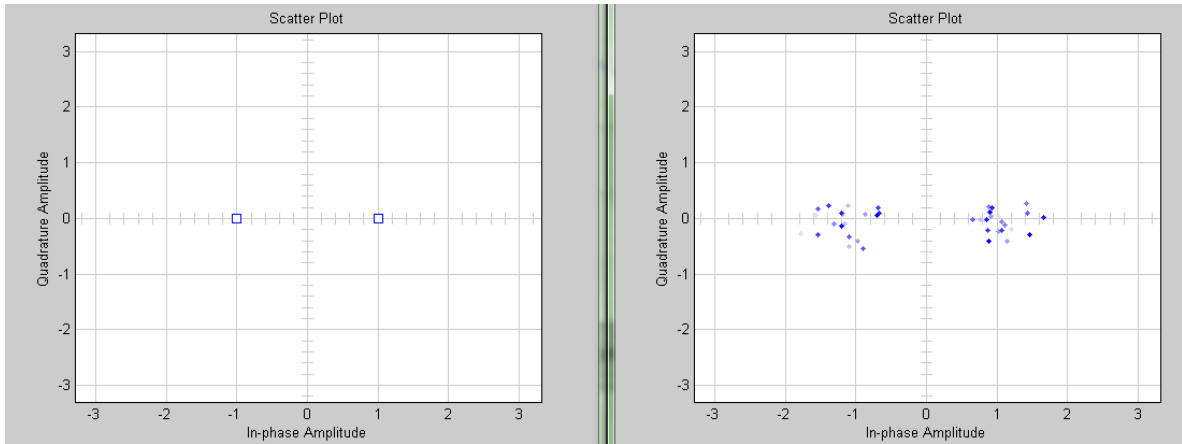


FIGURE 3.3 – Diagramme de constellation avant et après le canal de transmission BPSK

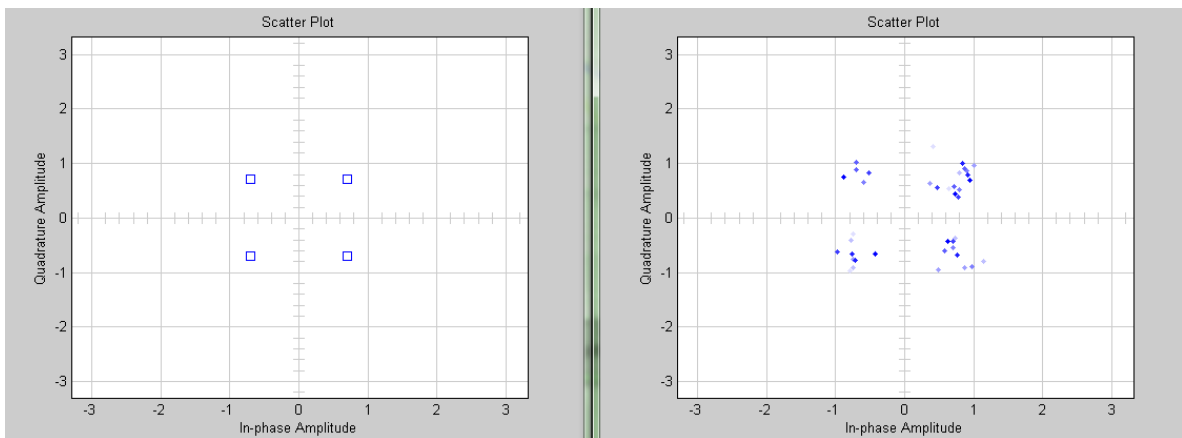


FIGURE 3.4 – Diagramme de constellation avant et après le canal de transmission QPSK



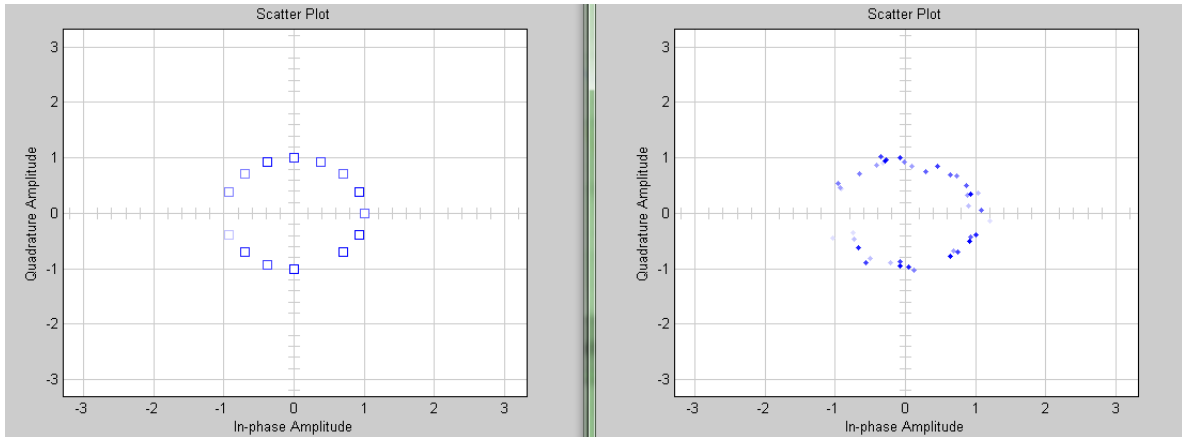


FIGURE 3.5 – Diagramme de constellation avant et après le canal de transmission 8PSK

Nous remarquons d'après ces diagrammes de constellation l'effet du bruit sur le signal tel que chaque symbole de la constellation avant l'émission a perdu son emplacement après le canal AWGN ; d'où la probabilité de perte d'information au niveau de la réception.

### 3.3.1 Comparaison entre les méthodes de modulation sans codage

Les figures 3.6 et 3.7 représente une comparaison entre la partie theorique et experimental de la chaine :

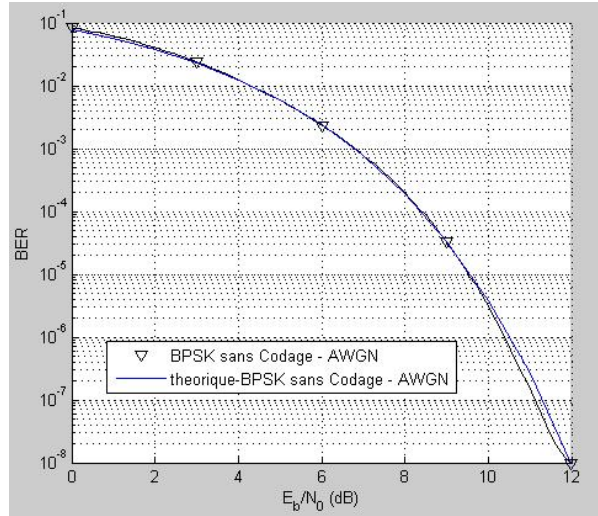


FIGURE 3.6 – comparaison entre la partie théorique est expérimentale d'une modulation BPSK

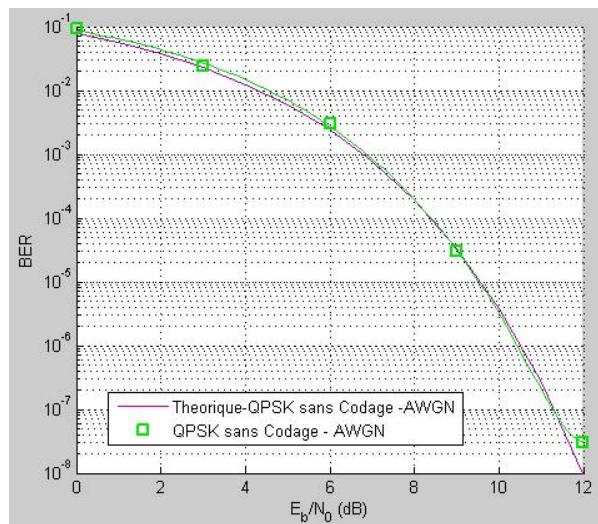


FIGURE 3.7 – Comparaison entre la partie théorique est expérimentale d'une modulation QPSK

Lors de simulation on constate la superposition des courbes BPSK et QPSK sur la courbe théorique; ce qui confirme l'exactitude de la simulation du système. Ainsi nous pouvons poursuivre la simulation avec l'insertion de divers bloc de codage afin d'améliorer les performances du système .

la figures 3.8 illustre les courbes de performances obtenus dans le cas d'une chaine sans codage en utilisant les modulation BPSK, QPSK, 8PSK,...64PSK.

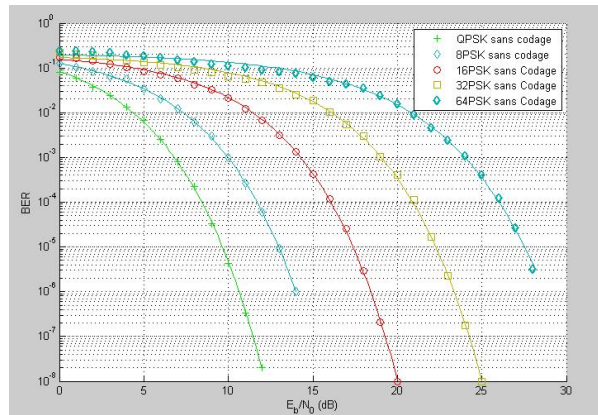


FIGURE 3.8 – Performances du système pour différentes modulation sans codage

### 3.3.2 Comparaison entre les méthodes de modulation avec canal rayleigh

Dans cette phase de simulation ,nous avons effectuè une comparaison entre les différents modulation avec un canal AWGN et un canal Rayleigh , les figures 3.9 et 3.10 montre les performances des modulation BPSK et QPSK dans un canal Rayleigh :

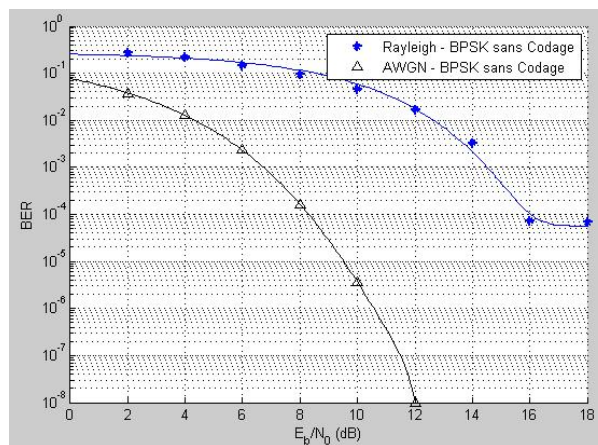


FIGURE 3.9 – Performances du TEB du système dans une modulation BPSK avec dans un canal Rayleigh

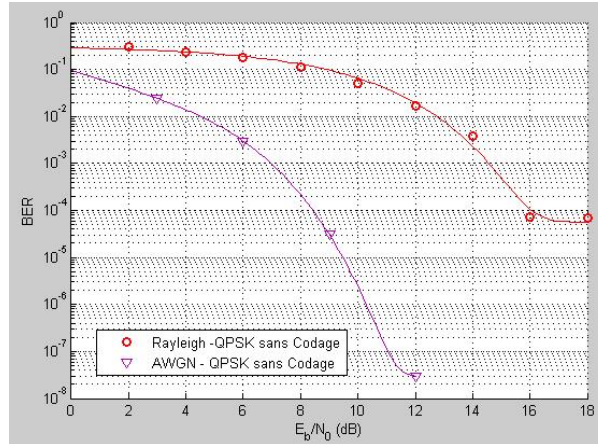


FIGURE 3.10 – performances du TEB du système dans une modulation QPSK dans un canal Rayleigh

Nous remarquons d'après les figures 3.9 et 3.10 dans le cas d'un canal AWGN associé à un canal Rayleigh que pour obtenir un TEB de l'ordre  $10^{-4}$  nous devons fournir une énergie assez importante en comparaison avec celle d'un canal AWGN simple et cela est dû à l'effet d'évanouissement provoqué par un canal Rayleigh.

## 3.4 Insertion des blocs de codage et de décodage

Dans cette partie on conserve la chaîne réalisée précédemment, avec un bruit blanc gaussien et additif. ce qui n'est pas performant quand on sait que le but d'une transmission numérique est d'avoir un TEB assez faible avec le minimum d'énergie. D'où la nécessité d'apporter une protection à la transmission, pour ce faire nous avons rajouté un codeur-décodeur reed-solomon ( $n=15, k=9, T=6$ ), La figure 3.9 illustre cette insertion.

### 3.4.1 Insertion de codage convolutif

La variation de la performance du système avec utilisation du codage convolutif est présentée Comme indiqué au niveau du chapitre 2 (section 2.2.2), selon un taux de codage de  $1/2$ , d'une longueur de contrainte  $K = 3$  du code convolutif.

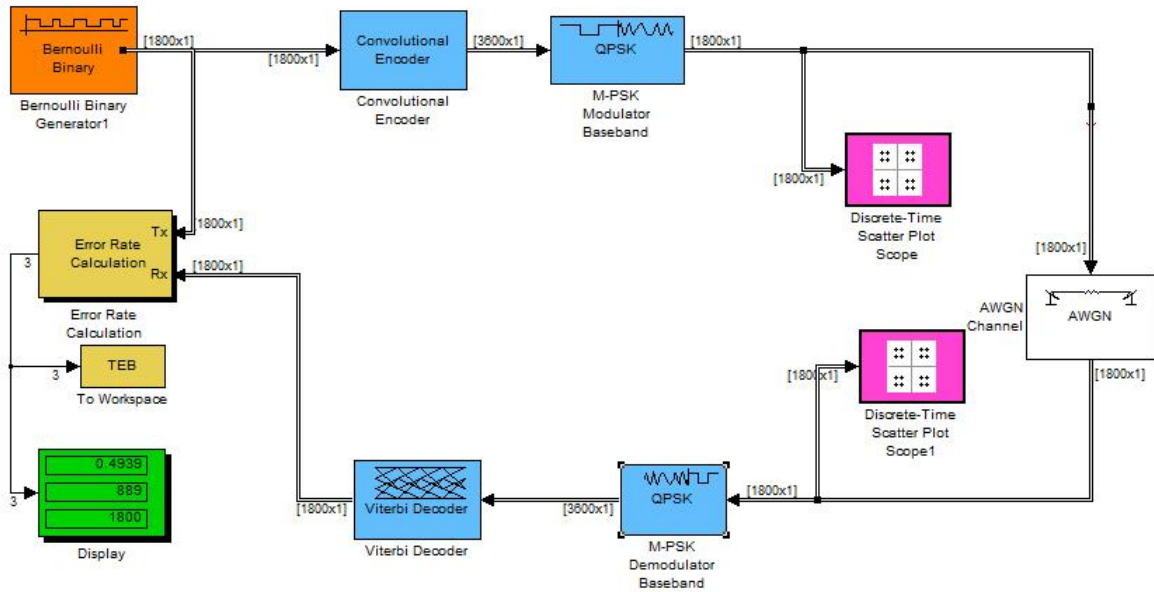


FIGURE 3.11 – la chaîne de transmission avec codage convolutif

Dans les figures 3.12 et 3.13 nous avons effectué une comparaison entre les modulation BPSK et QPSK avec et sans codage convolutif utilisé au niveau d'émission et le décodage de viterbi à la réception.

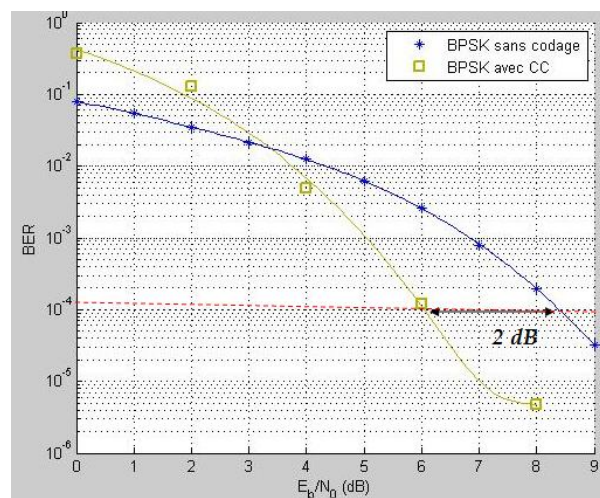


FIGURE 3.12 – Performances du T8B du système dans une modulation BPSK avec CC

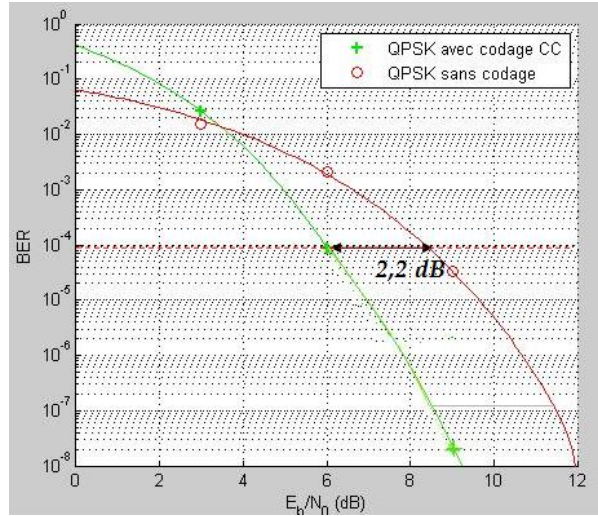


FIGURE 3.13 – Performances du TEB du système dans une modulation QPSK avec CC

D'après les figures ci-dessus, nous constatons lorsque le code convolutif est employé, on peut observer un gain d'énergie de 2 dB dans la modulation BPSK et 2,2 dB dans la modulation QPSK est obtenu pour un TEB de l'ordre de  $10^{-4}$ .

### 3.4.2 Comparaison entre les méthodes de modulation avec codage convolutif dans un canal rayleigh

Dans cette étape de simulation, nous allons présenter les performances du code convolutif avec les différents modulation dans un canal Rayleigh, les figures 3.14 et 3.15 illustrent cette étape :



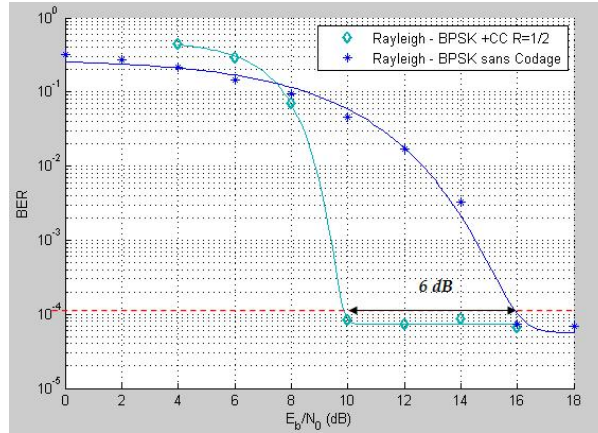


FIGURE 3.14 – Performances du TEB du système dans une modulation BPSK avec CC dans un canal Rayleigh

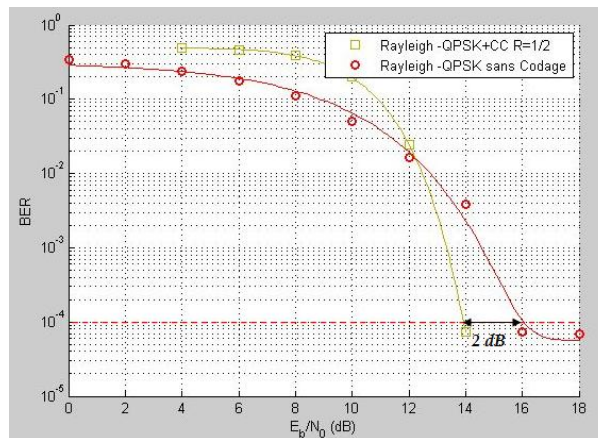


FIGURE 3.15 – Performances du TEB du système dans une modulation QPSK avec CC dans un canal Rayleigh

Dans cette partie de simulation, nous constatons lorsque le code convolutif est employé dans un canal Rayleigh on peut observer que la modulation BPSK est la plus robuste de notre système. Elle permet une diminution importante du TEB et cela avec un petit rapport  $E_b/N_0$ . Elle est donc la plus intéressante pour garantir la qualité de la transmission, mais avec un débit théorique assez faible. Et inversement pour modulation QPSK.

### 3.4.3 poinçonnage

pour obtenir des différents taux de codage ; on élimine certains bits. la fonction de poinçonnage est intégrée dans le bloc convolutif illustré par le tableau 3.1.

modulation	Taux de codage	Vecteur de poinçonnage
BPSK	1/2	pas de poinçonnage
QPSK	1/2	pas de poinçonnage
QPSK	3/4	reshape ([1 0 1;1 1 0],)

TABLE 3.1 – Vecteurs de poinçonnage

### 3.4.4 Comparaison entre les méthodes de modulation avec codage

Dans les figures 3.16 et 3.17 nous avons effectué une comparaison total entre les différentes modulation et les différents taux de codage qui sont définis dans notre système

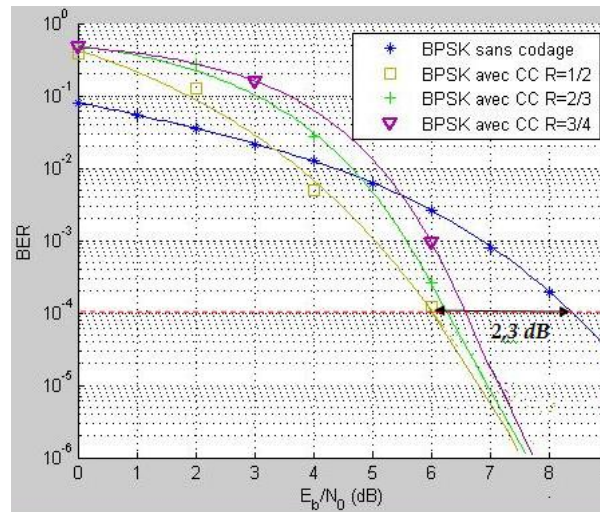


FIGURE 3.16 – Comparaison des performances du système pour la modulation BPSK et taux de codage CC



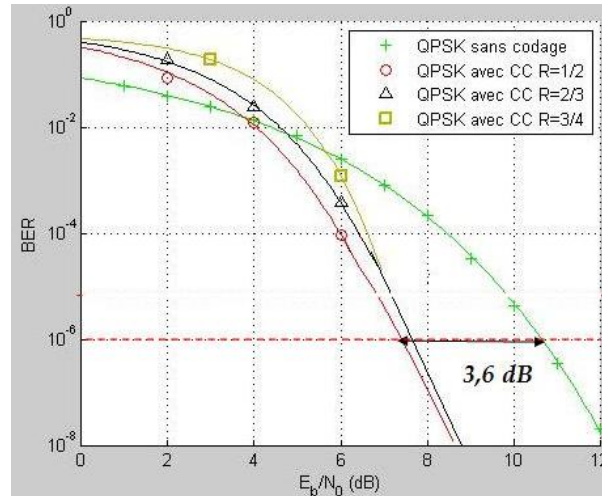


FIGURE 3.17 – Comparaison des performances du système pour la modulation QPSK et taux de codage CC

Dans cette étape en fait varier le taux de codage afin de constater que le rendement 1/2 est le plus performant dans les différents types de modulation et surtout avec la modulation QPSK avec un gain d'énergie de 3,6 dB pour un TEB de l'ordre de  $10^{-6}$ .

### 3.4.5 Insertion de codage reed-solomon et l'entrelaceur

Le code Reed-Solomon est employé en concaténation avec le code convolutif, pour fournir d'avantage de gain. Un code systématique de RS ( $n=15; k=9$ ) où les éléments du champ de Galois GF (Galois Field), sont de  $(2^4)$ . La figure 3.18 montre l'évolution du système quand le codage Reed-Solomon est concaténé au codage convolutif. On peut observer un gain additionnel par rapport au codage convolutif est obtenu.

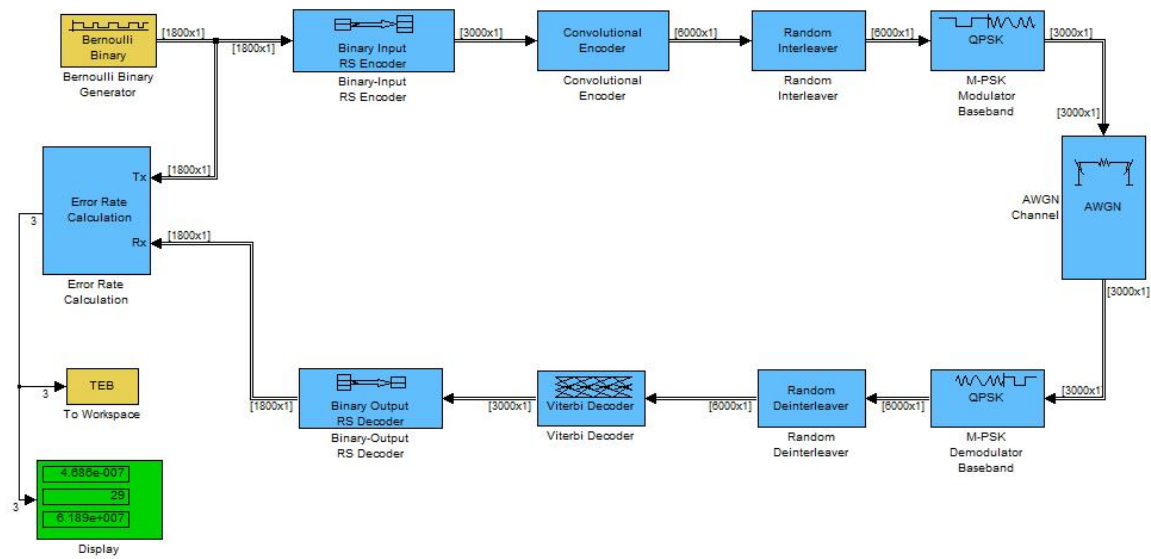


FIGURE 3.18 – Modélisation de la chaîne de transmission avec codage concaténé (CC+RS)

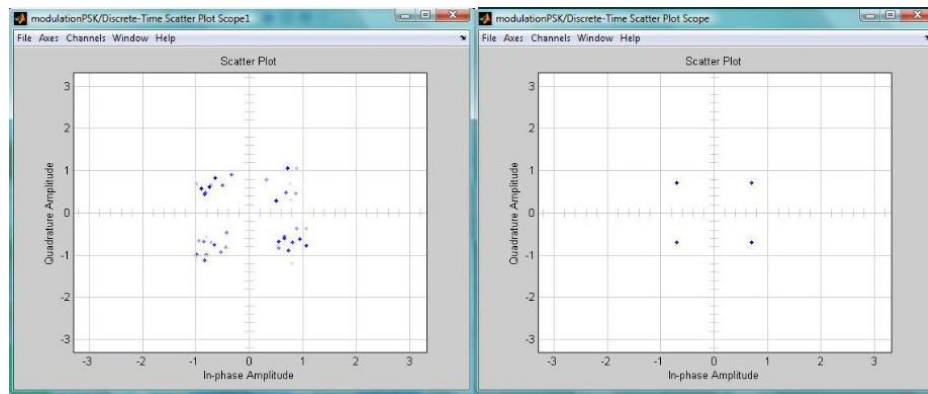


FIGURE 3.19 – Le diagramme de constellation de codage concaténé

Nous remarquons tout comme la représentation des diagrammes de constellation de la première chaîne qu'il y a une déformation par le canal de transmission, par contre on peut constater que le diagramme de constellation après le canal de cette chaîne est moins bruité que celui de la précédente et ceci grâce au code **RS** appliqué qui apporte une robustesse au signal initial figure 3.19.

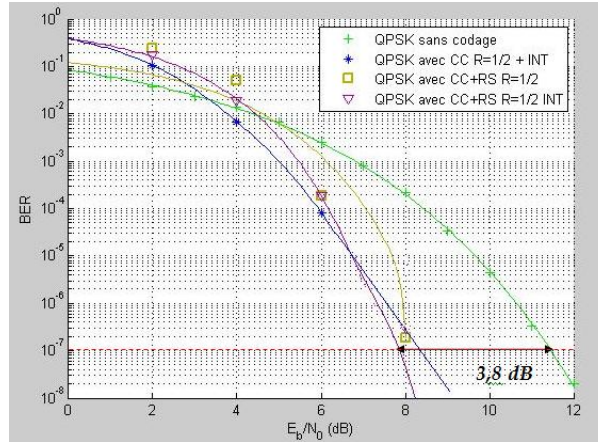


FIGURE 3.20 – Performances du système pour une modulation QPSK en utilisant un codage concaténé convolutif et reed-solomon avec un entrelaceur

En dernière partie, nous avons apporté une chaîne de transmission avec un codage reed-solomon concaténé avec un codage convolutif (**RS+CC**) on à observé une amélioration remarquable a obtenu de l'ordre de 3.8 dB pour un TEB de  $10^{-7}$

### 3.5 Conclusion

Dans ce chapitre, nous avons présenté la simulation d'une chaîne de transmission. Nous avons évalué les performances d'un code reed-solomon concaténé avec un code convolutif. Une comparaison a été effectuée pour les différents types de modulation et taux de codage, selon les résultats obtenu, nous pouvons dire que l'intégration d'un code concaténé entre le code **RS+CC** ainsi qu'un entrelaceur permet d'améliorer performances grâce à sa capacité et efficacité de connection.

# Chapitre 4

## Conclusion

Ce projet a été consacré à l'étude des performances d'un système de transmission numérique sous Simulink de Matlab, en utilisant des codes convolutifs et de codes de Reed-Solomon ainsi que les différents types de modulations M-PSK dans deux canaux de propagation différents AWGN et Rayleigh. Le but de cette recherche est de proposer un système de codage concaténé introduisant les codes mentionnés ci-dessus. Comme nous l'avons constaté, les codes de Reed-Solomon améliorent les performances du système d'environ 0,5dB par rapport à des codes convolutifs classiques.

Au premier chapitre nous avons présenté une introduction aux systèmes de transmission numérique, exposé les caractéristiques principales de transmission telles que : le codage canal, les différents types de modulation, les diagrammes de constellation de chaque type, canal de propagation, ...etc.

Au deuxième chapitre nous avons décrit la chaîne de transmission et ces différents blocs constituant : source, codage canal où nous avons étudié les codes convolutifs et leur decodeur Viterbi ainsi que le codeur Reed-Solomon, technique FEC, entrelacement, les différents modulations BPSK, QPSK, ...etc,

D'autre part on a pu valider notre étude théorique en réalisant une chaîne de transmission qui implémente deux types de codes ; le premier est un code convolutif classique associé à un algorithme de Viterbi, le deuxième est un FEC. Pour cela on a réalisé trois chaînes de simulation, sans codage, avec code convolutif et avec un code concaténé (RS+CC) pour différents types de modulation et taux de codage.

En comparant à l'aide des simulations, les performances en termes de taux d'erreurs binaires de notre système utilisant les différents types de modulation de sous-porteuses,

nous avons constaté qu'avec la modulation 64 PSK les dégradations de performances du système deviennent importantes par rapport à une modulation QPSK.

Le présent travail détermine les aspects pratiques qui peuvent améliorer les systèmes de communication numérique, comme l'utilisation d'une concaténation en série de deux codes convolutionnel et de Reed-Solomon qui conduit à de meilleurs résultats de performance du système. on conclut que les performances du système ont tendance à augmenter lorsque le rapport  $E_b/N_0$  est élevé.

Par conséquent, nous suggérons que d'autres recherches à venir puissent approfondir davantage l'analyse de la concaténation en série des différents types de codage permettant ainsi d'obtenir une robustesse accrue et une amélioration des performances du système.

# Bibliographie

# Bibliographie

- [1] F. Lustenberger, « On the Design of Analog VLSI Iterative Decoders », Dissertation ETH N<sup>o</sup>. 13879, Serie in Signal and Information Processing : Volume 2, Hartung Gorre, Konstanz, Allemagne, novembre 2000.
- [2] Thitimaishima, « Les codes Convolutifs Récursifs Systématiques et leur application à la concaténation parallèle », Thèse de Doctorat en Electronique, Université de Bretagne Occidentale, France, 1993.
- [3] F. Monteiro, A. Dandache, A. M'sir and B. Lepley, « A fast CRC implementation on FPGA using a pipelined architecture for the polynomial division », Proc. 8th ICECS, St Julian, Malta, Sept. 2001, pp.12311234.
- [4] L. Lampe, M. Jain, R. Schober, « Improved decoding for Bluetooth systems », Proc. ICC 2005, 16-20 May 2005, pp. 25112515, Vol. 4.
- [5] Y. Fei Guo, Z. C. Li, Q. Wang, « An area-efficient reed-solomon decoder for HDTV channel demodulation », Proc. of the 2nd IEEE ASME06, pp. 15, Aug. 2006.
- [6] S.S. Alekseev, A.V. Krivosheikin, « Digital implementation of DVB-C reverse channel receiver », IEEE ISCE06, pp. 13, 2006.
- [7] J. Jin, C-Y. Tsui, « A low power viterbi decoder implementation using scarce state transition and path runing scheme for high throughput wireless applications », Proc. IEEE ISLPED06, pp. 406411, Tegernsee, Hong Kong, 46 Oct. 2006.
- [8] A. Cardenal-López, C. García-Mateoa, L. Docío-Fernández, « Weighted viterbi decoding strategies for distributed speech recognition over IP networksstar », Speech Communication, vol. 48, pp. 14221434, Nov. 2006.
- [9] C. Shannon. A mathematical theory of communication. The Bell System Technical Journal, 27 :379423, 623656, July, October 1948.
- [10] 3GPP TS 05.03 v8.9.0. Channel coding (release 1999). The 3rd Generation Partnership Project, Technical Specification Group GSM/EDGE Radio Access Network, January 2005. [http ://www.3gpp.org](http://www.3gpp.org).

- [11] 3GPP2 C.S0002-E v0.99. Physical layer Standard for CDMA2000 Spread Spectrum Systems (Revision E). The 3rd Generation Partnership Project 2, February 2009. <http://www.3gpp2.org>.
- [12] 3GPP TS 25.212 v6.5.0. Multiplexing and channel coding (FDD) (release 6). The 3rd Generation Partnership Project, Technical Specification Group Radio Access Network, June 2005. <http://www.3gpp.org>.
- [13] 3GPP TS 36.212 v8.3.0. Multiplexing and channel coding (FDD) (release 8). The 3rd Generation Partnership Project, Technical Specification Group Radio Access Network, May 2008. <http://www.3gpp.org>.
- [14] R. W. Hamming, Error detecting and error correcting codes, Bell Syst. Tech. J., vol. 29, pp. 147169, Avril 1950.
- [15] P. Elias, Error-free coding, I.R.E Trans. Inform. Theory, vol. PGIT-4, pp. 2937, Sept.1954.
- [16] P. Elias, Coding for noisy channels, I.R.E Conv. Rec., vol. 3, pp. 3746, Sept. 1955.
- [17] R. W Hamming. Error detecting and error correcting codes. The Bell System Technical Journal, XXIX(2) :147160, April 1950.
- [18] AR. C Bose and D. K Ray-Chaudhuri. On a class of error correcting binary group codes. Information and Control, 3(1) :6979, March 1960.
- [19] A. Hocquenghem. Codes correcteurs d'erreur. Chiffres, 2 :147156, September 1959.
- [20] I. S Reed and G. Solomon. Polynomial codes over certain finite fields. Journal of the Society for Industrial and Applied Mathematics (SIAM), 8(2) :300304, 1960.
- [21] R. G Gallager. Low-Density Parity-Check codes. M.I.T Press, 1963.
- [22] P. Elias. Coding for noisy channels. IRE International Convention Record, 4 :3746, 1955.
- [23] J. M Wozencraft. Sequential decoding for reliable communication. IRE International Convention Record, 5(2) :11 25, 1957.
- [24] R. M Fano. A heuristic discussion of probabilistic decoding. IEEE Transactions on Information Theory, 9(2) :64 74, April 1963.
- [25] A. J Viterbi. Error bounds for convolutional codes and asymptotically optimum decoding algorithm. IEEE Transaction on Information Theory, 13 :260269, April 1967.



- [26] G. David Forney. Convolutional codes I : Algebraic structure. IEEE Transactions on Information Theory, 16(6) :720738, November 1970.
- [27] G. David Forney. Structural Analysis of Convolutional codes via Dual Codes. IEEE Transactions on Information Theory, 19(4) :5125188, July 1973.
- [28] L. R Bahl, J. Cocke, F. Jelinek, and J. Raviv. Optimal decoding of linear codes for minimizing symbol error rate. IEEE Transaction on Information Theory, 20(2) :284 287, March 1974.
- [29] J. G. Smith, Odd bit symmetric QASK, IEEE Trans., vol. COM-23, pp. 385389, Mars 1975.
- [30] C. M. Thomas, Digital amplitude phase keying with m-ary alphabets, IEEE Trans., vol. COM-22, pp. 168 179, Fév. 1974.
- [31] M. K. Simon et J. G. Smith, Hexagonal multiple phase and amplitude shift-keyed signal sets, IEEE Trans., vol. COM-21, pp. 1108 1115, Oct. 1973.

# Annexe A

## LISTE DES SYMBOLES ET DES ABBREVIATIONS

### ABBREVIATIONS

AWGN : Additive White Gaussian Noise.

APSK : Amplitude Phase Shift Keying.

BPSK : Binary phase shift keying .

BBAG : Bruit Blanc Additif Gaussien.

CC : Code Convolutif.

FEC : Forward Error Correction.

GF : Corps de galois.

GSM : Global System for Mobile Communications.

LDPC : Low Density Parity Check.

MDP : Modulations par déplacement de phase.

MDAP : Modulation par déplacement d'Amplitude et de phase.

NRNSC : Non Recursive and Non Systematic Code.

PSK : Phase Shift Keying.

QAM : Quadrature Amplitude Modulation.

QPSK : Quadrature phase shift keying.

RSC : Récursifs Systématiques codes.

RS : Reed-Solomon.

TEB : Taux d'Erreur Binaire.

UMTS : Universal Mobile Telecommunications System.

# Annexe B

## QUELQUES DÉFINITION

### Définition du logiciel MATLAB

Matlab est un logiciel de calcul numérique produit par MAT-Works (voir le cite web [http :](http://www.mathworks.com/)

[www.mathworks.com/](http://www.mathworks.com/)). Il est disponible sur plusieurs plateformes. Matlab est un langage simple et très efficace, optimisé pour le traitement des matrices, d'où son nom. Pour le calcul numérique, Matlab est beaucoup plus concis que les vieux langages (C, Pascal, Fortran, Basic). Un exemple : plus besoin de programmer des boucles modifier de un a un les éléments d'une matrice. On traiter la matrice. On traiter la matrice comme une simple variable. Matlab contient également une interface graphique puissante, ainsi qu'une grande variété d'algorithmes scientifique. On peut enrichir Matlab en ajoutant des boites a outils (toolbox) qui sont les ensembles de fonctions supplémentaires, profilées pour les applications particulières (traitement de signaux, analyses statistiques, optimisation, etc.). Matlab est relativement couteux car il contient un nombre impressionnant de fonction. Il existe une version étudiant a un prix abordable et un clone (Octace), disponible en freeware, dont la compatibilité avec Matlab est assez bonne : ([http :// www.octave.com/](http://www.octave.com/) ou dans la distribution SuSE de Linux).

### SIMULINK

Simulink (simulation and Link) est une prolongation de MATLAB par Mathworks Inc. Cela fonctionne avec MATLAB pour offrir une modélisation, simulation et analyse des systèmes dynamiques sous un environnement graphique de l'interface utilisateur (GUI).

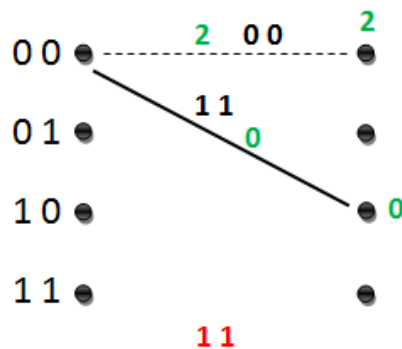
L'algorithme de plus utilisé pour le décodqge des codes convolutionneles est l'algo-

rithme de viterbi. celui-ci s'appuie sur le schéma en treillis. Le principe est de chercher une séquence sans erreur le plus proche de celle reçue (et donc la plus probable). on recherche donc une trame qui ait la distance la plus petite avec celle reçue.

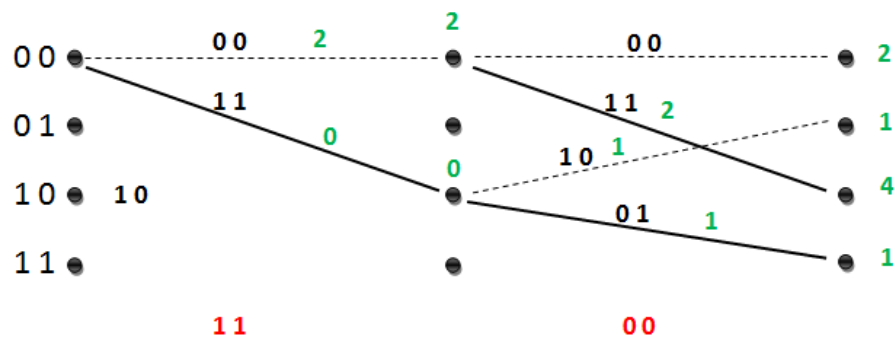
Alors après codage : 11 10 11 11 présence d'erreur de transmission ; données reçues : 11 00 11 11

### Les étapes de décodage

1ère étape : on part de l'état 00. on trace toutes branches possible à partir de cet état. pour chacune de ces branches ,on calcule la distance entre la valeur de la branche et le codé reçu (c1). On affecte à l'état d'arrivée le poids de la branche rattaché.

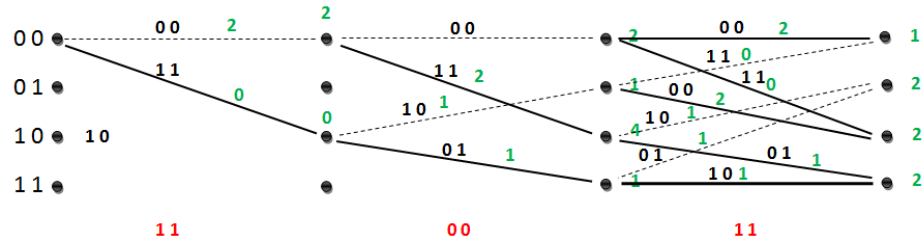


2ème étape : A partir des états d'arrivée précédents, on trace toutes les branches possibles comme a l'étape 1, on calcule la distance entre les valeurs des branches et le code reçu (c2) pour chaque état d'arrivée ,on effectue la somme entre le poids de la branche et le poids de l'état d'origine



3ème étape : A partir des états d'arrivée précédents, on trace toutes les branches possibles. Comme aux étapes précédentes , on calcule la distance entre les valeurs des

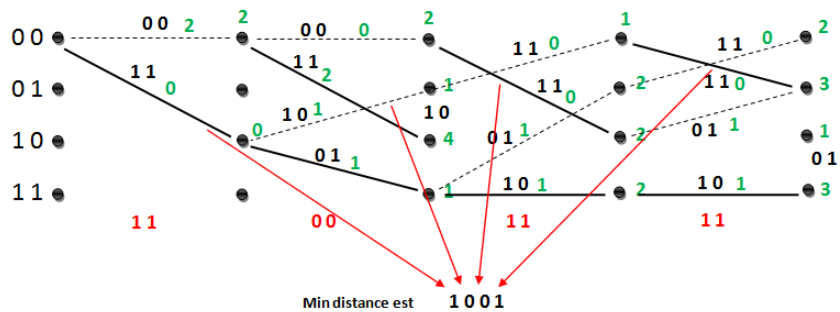
branches et le code reçu (c3). Pour chacune des braches arrivant à un état, on effectue la somme entre le poids de la branche et le poids de l'état d'origine. On affecte le résultat le plus petit a l'état d'arrivée.



4ème étape : Une fois ce travail effectué, pour chaque état d'arrivée, on ne garde que la branche qui a permis l'affectation du poids de cet état (c'est-à-dire celle qui a la somme entre son poids et le poids de l'état d'origine le plus petite). Les branches sont éliminées.

5ème étape : A cette étape, on regard les poids d'arrivée :

1er cas : le poids minimal des états d'arrivée n'est affecté qu'à un seul état dans ce cas le décodage peut avoir lieu le message est la suite des valeurs des braches (0 ou 1) du seul chemin possible pour atteindre cet état à partir de l'état initial. 2ème cas : le poids minimal des états d'arrivée est affecté à plusieurs états. Dans ce cas, le décodage aura lieu lors des étapes suivantes.

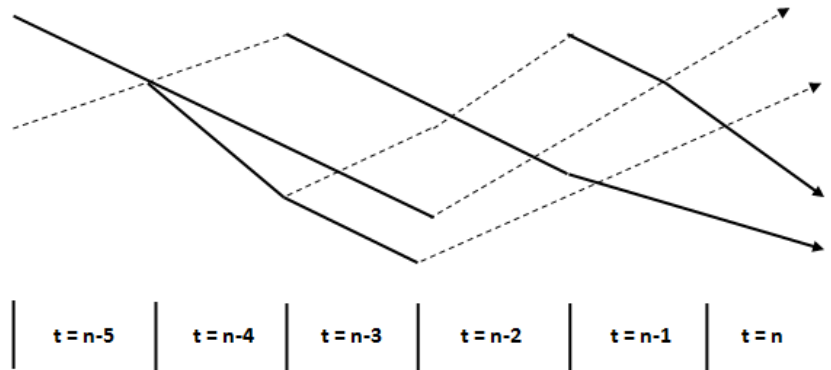


Afin d'arriver à la séquence reçue avec le minimum de bits erronés, on constate que le schéma le plus court est celui de la séquence reçue 11 10 11 11 ou la sortie est 1001.

**L'algorithme de Bahl et al.** L'algorithme de Viterbi minimise pas nécessairement la probabilité d'erreur pas symbole ou par bit. L'algorithme de Bahl et al (à

entrées et sorties souples SISO) minimise la probabilité d'erreur par symbole. Il n'est guère intuitif et sa complexité a fait qu'il soit oublié jusqu'à l'arrivée des Turbo Codes.

**Limitation du décodage à une fenêtre** En pratique, on n'attend pas que toute la séquence binaire émise soit reçue pour commencer le décodage (trop de mémoire). On s'aperçoit qu'au bout d'un certain temps  $t=n$ , les chemins survivants convergent vers un même chemin en amont en  $t = n - x$ . On garde tous les chemins survivants de  $n - x$  à  $t = n$ , on décide de décoder l'échantillon de  $t = n - x$ .



Pour un rendement  $1/2$   $n \cong 5$  à  $6$  fois la longueur de contrainte.