
# COM105

# Communications Numériques et Théorie de l'Information

Philippe Ciblat, Michèle Wigger

15 février 2021

# Table des matières

# Notations

**Outils mathématiques :**

- Nombre imaginaire pur : $i = \sqrt{-1}$
- Transformée de Fourier : $X(f) = \mathrm{FT}(x(t)) = \int_{t=-\infty}^{\infty} x(t) e^{-2\pi i f t} \mathrm{d}\, t$
- Transformée de Fourier Inverse : $x(t) = \mathrm{IFT}(X(f)) = \int_{f=-\infty}^{\infty} X(f) e^{2\pi i f t} \mathrm{d}\, f$
- Alphabet d'une variable aléatoire $X$ : $\mathcal{X}$
- Loi de probabilité d'une variable aléatoire discrète $X$ (*prob. mass func. -pmf-*) : $x \mapsto P_X(x)$
- Loi de probabilité conjointe d'une paire de variables aléatoires discrètes $(X, Y)$ : $(x, y) \mapsto P_{XY}(x, y)$
- Loi de probabilité conditionnelle d'une variable aléatoire discrète $X$ par rapport à une deuxième variable $Y = y$ : $x \mapsto P_{X|Y}(x|y)$
- Densité de probabilité d'une variable aléatoire continue $X$ (*prob. density func. -pdf-*) : $x \mapsto p_X(x)$
- Densité de probabilité conditionelle d'une variable aléatoire continue $X$ par rapport à une deuxième variable $Y$ : $x \mapsto p_{X|Y}(x|y)$
- Probabilité de l'événement $\omega$ : $\Pr(\omega)$

**Signaux et suites :**

- Autocorrélation d'un signal aléatoire $x(t)$ : $r_{xx}(t, \tau) = \mathbb{E}[x(t+\tau)x(t)]$
- Densité spectrale de puissance d'un signal aléatoire $x(t)$ : $f \mapsto \mathcal{S}_{xx}(f)$
- Variance d'une suite aléatoire $\{s_k\}_k$ : $\sigma_s^2 = \mathbb{E}[(s_k - m_s)^2]$ avec $m_s = \mathbb{E}\big[s_k\big]$
- Puissance moyenne du signal $x(t)$ : $P_x$

**Codage et modulation :**

- Bits de la source ou Données : $\{d_m\}_m$ (typiquement i.i.d. Bernouilli de probabilité 1/2)
- Bits codés : $\{c_m\}_m$
- Longueur d'un mot de code : $n$ (attention, $n$ sera parfois une variable muette de sommation mais cela ne devrait créer aucune ambiguïté)
- Nombre de bits d'information dans un mot de code : $k$ (idem, $k$ sera parfois une variable muette de sommation mais cela ne devrait créer aucune ambiguïté)
- Rendement d'un code correcteur d'erreur : $R = k/n$
- Symboles émis : $\{s_k\}_k$
- Signal émis : $x(t) = \sum_k s_k g(t - kT_s)$
- Filtre de mise en forme : $g(t)$
- Durée d'un symbole : $T_s$
- Temps-bit : $T_b$
- Débit (bit utile) : $D_b = R/T_b$
- Signal reçu : $y(t) = x(t) + b(t)$
- Bruit blanc gaussien de moyenne nulle et de niveau spectral $N_0/2$ : $b(t)$
- Filtre de réception : $g_r(t)$
- Signal après filtrage de réception : $z(t) = (y \star g_r)(t)$
- Signal après échantillonnage : $z_k = (y \star g_r)(kT_s)$ (quand $g \star g_r$ satisfait le critère de Nyquist, $z_k = s_k + w_k$ avec $w_k$ une gaussienne i.i.d. de moyenne nulle et de variance $N_0/2$)

**Théorie de l'information**

- Entropie d'une variable aléatoire $X$ : $H(X)$
- Entropie conditionelle d'une variable aléatoire $X$ sachant $Y$ : $H(X|Y)$
- Entropie jointe d'une paire de variable aléatoires $(X, Y)$ : $H(X, Y)$
- Fonction d'entropie binaire : $H_{\mathrm{b}}(\cdot)$
- Information mutuelle de $X$ et $Y$ : $I(X; Y)$

— Capacité de canal : $C$

**Paramètres du système :**

— Largeur de bande : $B$
— Facteur d'excès de bande : $\rho$
— Energie par symbole : $E_s$
— Energie par bit utile : $E_b$
— Probabilité d'erreur symbole : $P_e$
— Probabilité d'erreur bit : $P_b$
— Rapport Signal-à-Bruit (RSB) : $\text{RSB} = 2E_s/N_0$

# Abréviations

— BEC : *Binary Erasure Channel.* ..... Canal Binaire à Effacement
— BER : *Bit Error Rate.* ..... Taux d'Erreur Bit
— BSC : *Binary Symmetric Channel.* ..... Canal Binaire Symétrique
— DMC : *Discrete Memoryless Channel.* ..... Canal Discret sans Mémoire
— FEC : *Forward Error Correction.* ..... Code Correcteur d'Erreur
— FT : *Fourier Transform.* ..... Transformée de Fourier
— IFT : *Inverse Fourier Transform.* ..... Transformée de Fourier Inverse
— i.i.d. : *Independent and identically distributed.* ..... indépendant et identiquement distribué
— ISI : *Intersymbol Interference.* ..... Interférence Entre Symboles
— MIMO : *Multi-Input Multi-Output Antenna.* ..... Multi entrées- Multi sorties
— OOK : *On-Off Keying.* ..... Marche-Arrêt
— PAM : *Pulse Amplitude Modulation.* ..... Modulation d'Amplitude
— QoS : *Quality of Service.* ..... Qualité de Service
— SIMO : *Single-Input Multi-Output Antenna.* ..... Une entrée - Plusieurs sorties
— SNR : *Signal-to-Noise Ratio.* ..... Rapport Signal-à-Bruit

# Introduction générale

Sur la figure 1, nous avons représenté sur un même schéma un ensemble de réseaux de communication ayant des supports très différents (DVD, câble de cuivre, fibre optique, sans fil) et offrant des fonctionnalités très différentes : téléphone, transfert de données, flux vidéo (*video streaming*, en anglais), télévision, stockage dans des centres de données (*data centers*, en anglais). D'autres supports de communication, non dessinés sur la figure 1, existent tels que la clef USB, les molécules (via les communications moléculaires), le système cérébral (que certains chercheurs tentent d'expliquer via les principes et outils de communications numériques), etc.

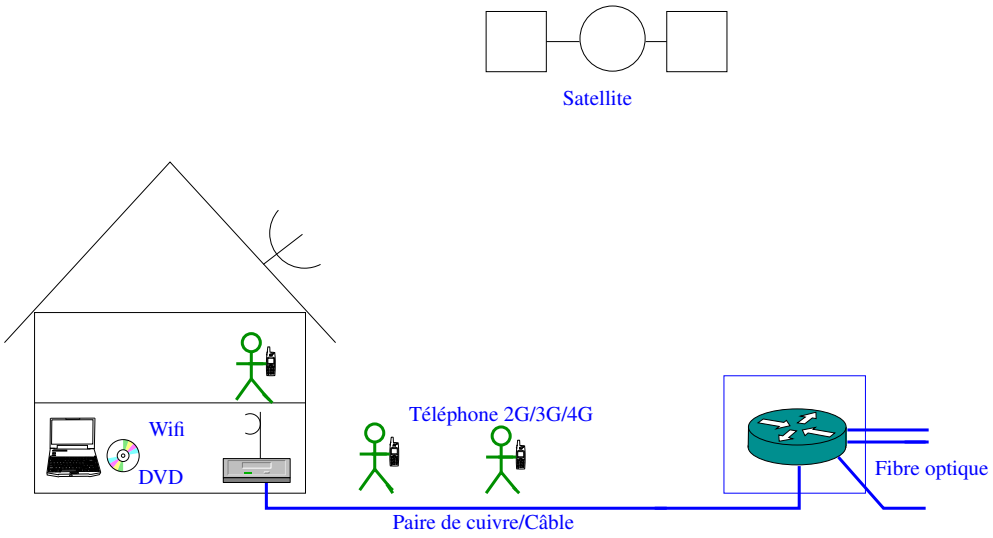



FIGURE 1 – Un ensemble de réseaux de communication

Néanmoins tous ces éléments ont un point commun : les données sont numérisées et doivent être transmises à travers un canal de propagation. Le domaine des communications numériques regroupe de ce fait les problématiques suivantes

- **comprimer**
- **transmettre**
- **sauvegarder**

**l'information de manière fiable avec un coût faible** (par coût, on peut entendre, consommation énergétique, occupation spectrale, complexité raisonnable, etc).

Dans ce polycopié, afin de simplifier, nous allons nous intéresser uniquement à des communications dites point-à-point, c'est-à-dire, entre un seul émetteur et un seul récepteur. Nous passerons donc sous silence la dimension multipoints-à-multipoints de tout réseau de communication, c'est-à-dire, quand plusieurs utilisateurs sont présents dans le système ou bien quand le réseau a plusieurs flux de données à gérer de manière simultanée. Cette dimension « réseau » des communications numériques est évidemment abordée dans les cours du cycle Master et la recherche académique ainsi qu'industrielle est actuellement très active sur cette dimension. En effet, les outils et techniques développés en communications numériques point-à-point offrent des perspectives très prometteuses pour optimiser les réseaux dans leur entier. Historiquement, la stratification OSI a découplée les deux domaines : la science des communications numériques développe les technologies pour bien transmettre UN paquet ; la science des réseaux apprend à manipuler efficacement LES

paquets. Ce découplage est en fait clairement sous-optimal et tend à s'estomper dans les systèmes en voie de standardisation. La conséquence en est l'insertion de plus en plus forte des techniques de communications numériques dans le cœur des réseaux.

Comme déjà dit, ce polycopié offre seulement quelques prolégomènes à l'étude des communications numériques. Classiquement, les communications sont subdivisées en trois sous-domaines que nous retrouvons dans notre découpage en trois chapitres :

- Dans le chapitre 1, vous apprendrez comment **protéger** les données contre des erreurs éventuelles via l'outil du **codage correcteur d'erreur**.

- Dans le chapitre 2, vous apprendrez à **envoyer et recevoir** les données protégées sur les supports de communication (qui peuvent être de tout type : fibre, câble, atmosphère, etc) via l'outil des **modulations numériques**.

- Dans le chapitre 3, vous apprendrez que les communications numériques ont des **performances-limites** (par exemple, en terme de débit) via l'outil de la **théorie de l'information**. Vous définirez en autre la notion (de quantité) d'information.

Tout chapitre se termine par une section, dite « Bilan », dans laquelle nous synthétisons dans deux boîtes les **concepts de base** à retenir et les **savoir-faire** à acquérir (qui sont en lien avec les TDs). Les cours de 2A associés à ce polycopié sont dans les filières

— Algèbre appliquée (ACCQ)

— Télécommunications : des données aux systèmes (TELECOM)

et les cours de 3A associés à ce polycopié sont dans le master suivant de l'Institut Polytechnique de Paris

— Information Processing : Machine Learning, Communications, and Security (MICAS)

accessible à partir des deux filières mentionnées ci-dessus.

# Chapitre 1

# Codage correcteur d'erreur

## 1.1 Introduction

Qu'est-ce-que le codage dit codage de canal ? A quoi cela sert-il ? Dans certains applications générant des erreurs comme la communication sur un canal bruité ou le stockage dans des grands serveurs, le codage de canal peut servir à réduire le nombre d'erreurs voire à les éliminer totalement. Il peut également deviner des effacements (c'est-à-dire, décider si un bit effacé est égal à 0 ou à 1).

Dans ce polycopié, nous allons nous intéresser uniquement aux codes en bloc qui, couplés parfois avec d'autres types de codes, sont utilisés dans tous les systèmes actuels (2G, 3G, 4G, 5G, ADSL, TNT, Wifi, etc).

Le codage en bloc est dessiné pour le système de communication présenté dans la figure 1.1 et expliqué dans la suite.

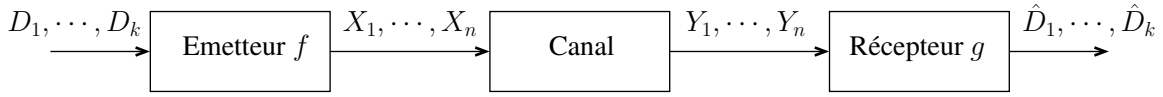



FIGURE 1.1 – Modèle simple d'un système de communication

1. *Bits d'information :* Le but de la communication est de transmettre $k$ bits d'information $\mathbf{D} = (D_1, \ldots, D_k)$ qui représentent les données. Ces bits d'information sont supposés complètement aléatoires et donc i.i.d. Bernoulli(1/2).
2. *Transmission en bloc :* On considère une transmission en bloc où les $k$ bits d'information sont envoyés sur un bloc de $n$ utilisations du canal. Dans la suite, nous utilisons les majuscules $\mathbf{D} = (D_1, \ldots, D_k)$ pour noter les bits d'information aléatoires et les minuscules $\mathbf{d} = (d_1, \ldots, d_k)$ pour une réalisation spécifique.
3. *Emetteur (aussi appelé codeur) :* L'émetteur associe à chaque mot de $k$ bits d'information $\mathbf{d} = (d_1, \ldots, d_k)$ un mot de longueur $n$, noté $\mathbf{c} := (c_1, \ldots, c_n)$ de façon bijective. Donc chaque mot d'information est associé à un mot codé différent. En notant cette fonction de codage par $f$, la suite des symboles codés s'écrit donc comme $\mathbf{C} := (C_1, \ldots, C_n) = f(\mathbf{D})$. Dans ce chapitre on suppose un système sans modulation pour lequel la suite codée est immédiatement transmise sur le canal, donc $\mathbf{X} := (X_1, \ldots, X_n) = \mathbf{C}$. Des liens plus subtils entre $\mathbf{X}$, $\mathbf{C}$ et $\mathbf{D}$ seront vus au chapitre 2.
4. *Récepteur (aussi appelé décodeur) :* Le récepteur essaie de retrouver à partir du mot reçu $\mathbf{y} = (y_1, \ldots, y_n)$ les valeurs des bits d'informations $\mathbf{d}$ transmis. Il produit un mot de bits devinés/détectés $\hat{\mathbf{d}} = (\hat{d}_1, \ldots, \hat{d}_k)$.
   Nous nous limitons à des récepteurs qui travaillent de façon déterministe et donc peuvent être décrits par une fonction de décodage $g(\cdot)$, c'est-à-dire, que $\hat{\mathbf{D}} = g(\mathbf{Y})$.
5. *Erreur :* Il y a erreur dans la transmission si
$$(\hat{d}_1, \ldots, \hat{d}_k) \neq (d_1, \ldots, d_k).$$

Le codage s'intéresse à la question comment concevoir la fonction de codage $f(\cdot)$ à l'émetteur et la fonction de décodage $g(\cdot)$ au récepteur. Commençons par donner un exemple très simple de ces deux fonctions.

**Exemple 1.1** *Considérons un code à répétition avec $k = 1$ et $n = 3$. Il transmet un seul bit d'information $d_1 \in \{0, 1\}$ auquel on associe le mot transmis $\boldsymbol{c}$ de longueur 3. Ces $\boldsymbol{c}$ ne prennent que deux valeurs qui sont en l'occurrence $(0,\ 0,\ 0)$ et $(1,\ 1,\ 1)$. Le tableau 1.1 donne la correspondance entre le bit d'information $d_1$ et les deux mots transmis.*

| Donnée $d_1$ | Mot codé $\mathbf{c}$ |
|---|---|
| 0 | 000 |
| 1 | 111 |

TABLE 1.1 – Fonction de codage $f(\cdot)$ du code à répétition de longueur $n = 3$

*Le tableau 1.2 donne la correspondance entre les mots reçus $\mathbf{y}$ et le bit $\hat{d}_1$ détecté par le décodeur dit à majorité. Notons que si le mot transmis $\mathbf{x}$ et le mot reçu $\mathbf{y}$ diffèrent au maximum dans une position, alors le décodeur retrouve toujours la bonne valeur pour $d_1$. Ceci n'est pas le cas si les deux mots diffèrent dans 2 ou plus positions. On dit alors que le code utilisé a une capacité de correction d'erreur de 1.*

| Mot reçu $\mathbf{y}$ | Donnée estimée $\hat{d}_1$ |
|---|---|
| 000 | 0 |
| 001 | 0 |
| 010 | 0 |
| 100 | 0 |
| 110 | 1 |
| 011 | 1 |
| 101 | 1 |
| 111 | 1 |

TABLE 1.2 – Une fonction de décodage $g(\cdot)$ du code à répétition de longueur $n = 3$

Dans ce chapitre nous allons voir d'autres exemples et surtout des méthodes pour choisir ces deux fonctions $f(\cdot)$ et $g(\cdot)$. Le meilleur choix dépendra du critère de performances et du canal de transmission. C'est pourquoi, nous serons dans l'obligation de discuter des modèles de canaux et des critères de performances possibles.

Le chapitre est organisé de la manière suivante :

- en section 1.2 nous introduisons la notion des codes en bloc et leurs caractéristiques principales.
- en section 1.3, nous nous intéressons aux codes en bloc linéaires qui sont une classe très répandue de codes car offrant des facilités de manipulations tant pratique que théorique.
- en section 1.4, nous introduisons des modèles de canaux aléatoires. Ceci permettra d'étudier l'efficacité des codes en fonction des canaux de transmission sur lesquels ils sont utilisés.
- en section 1.5, nous nous intéressons aux résultats de la théorie de la détection qui nous permettent de décrire un décodeur optimal.
- en section 1.6, nous analysons théoriquement les performances des codes.

En section 1.7, nous faisons un bilan de ce chapitre en rappellant les éléments essentiels à retenir et les outils à maîtriser.

## 1.2 Codes en bloc

En théorie du codage, ce qui est important c'est l'ensemble des suites produites par $f(\cdot)$, plutôt que la fonction de codage en elle-même. Les suites produites par $f(\cdot)$ sont appelées les mots de code, et l'ensemble de tous les mots de code est simplement appelé le code $\mathcal{C}$. Nous avons la définition suivante.

**Définition 1.1 (Codes en bloc)** *Un code en bloc $\mathcal{C}$ de longueur $n$ sur un alphabet $\mathcal{X}$ est un sous-ensemble de $\mathcal{X}^n$. Les éléments de $\mathcal{C}$ sont appelés les mots de code de $\mathcal{C}$.*

Si $|\mathcal{X}| = 2$, on appelle le code *binaire*.

**Définition 1.2 (Dimension d'un code)** *La dimension d'un code $\mathcal{C}$ de longueur $n$ sur un alphabet $\mathcal{X}$ est donné par*

$$k = \log_2 |\mathcal{C}|. \tag{1.1}$$

La dimension $k$ d'un code $\mathcal{C}$ détermine donc le nombre de bits d'information que le code peut coder en utilisant des mots de code différents pour des suites de bits différentes. Le nombre de bits d'information que le code envoie en moyenne par symbole codé, est connu comme le rendement du code.

**Définition 1.3** *Le rendement $R$ d'un code en bloc est donné par*

$$R = \frac{k}{n}. \tag{1.2}$$

### 1.2.1 Distance de Hamming et distance minimale

Un autre paramètre très important d'un code est sa *distance minimale*. Comme nous verrons dans la suite, on préfère en général des codes qui ont une distance minimale grande. Car, en utilisant un code avec une distance minimale plus élevée, on peut corriger des configurations d'erreurs (changement de bits) qui ne peuvent pas être corrigées avec un code de distance minimale plus faible.

Avant de donner la définition de la distance minimale, nous introduisons le poids et la distance de Hamming.

Le poids de Hamming mesure le nombre de positions qui ne sont pas nulles dans un mot. Le poids de Hamming d'un mot $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ est

$$w_H(\mathbf{x}) := \sum_{i=1}^{n} \mathbb{1}(\{x_i \neq 0\})$$

où $\mathbb{1}(\cdot)$ est la fonction indicatrice qui vaut 1 si l'argument est vérifié et zéro autrement.

La distance de Hamming mesure le nombre de positions pour lesquels deux mots diffèrent. La distance de Hamming entre deux mots $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ et $\tilde{\mathbf{x}} = (\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_n)$ est

$$d_H(\mathbf{x}, \tilde{\mathbf{x}}) := \sum_{i=1}^{n} \mathbb{1}(\{x_i \neq \tilde{x}_i\}).$$

La distance de Hamming satisfait bien les propriétés d'une distance ce qui justifie son nom.

**Résultat 1.1 (Propriétés de la distance de Hamming)** *Soient $\mathbf{x} = (x_1, \ldots, x_n)$, $\hat{\mathbf{x}} = (\hat{x}_1, \ldots, \hat{x}_n)$, et $\breve{\mathbf{x}} = (\breve{x}_1, \ldots, \breve{x}_n)$ trois mots dans $\mathcal{X}^n$. La distance de Hamming satisfait :*

1. $d_H(\mathbf{x}, \hat{\mathbf{x}}) = d_H(\hat{\mathbf{x}}, \mathbf{x})$. *(symétrie)*
2. $d_H(\mathbf{x}, \hat{\mathbf{x}}) \geq 0$ *avec égalité si et seulement si* $\mathbf{x} = \hat{\mathbf{x}}$. *(positivité)*
3. $d_H(\breve{\mathbf{x}}, \hat{\mathbf{x}}) \leq d_H(\mathbf{x}, \hat{\mathbf{x}}) + d_H(\mathbf{x}, \breve{\mathbf{x}})$. *(inégalité triangulaire)*

*Preuve :*

1. La définition de $d_H(\cdot, \cdot)$ est symétrique en ces deux arguments.
2. Par définition $d_H(\cdot, \cdot)$ est égale à la somme de valeurs 0 et 1, et donc est positive ou nulle.
3. Pour chaque $i \in \{1, \ldots, n\}$ :

$$\mathbb{1}(\{\breve{x}_i \neq \tilde{x}_i\}) \leq \mathbb{1}(\{x_i \neq \breve{x}_i\}) + \mathbb{1}(\{x_i \neq \tilde{x}_i\}).$$

   Comme $\mathbb{1}(\cdot)$ est soit 0 soit 1, l'inégalité est évidemment satisfaite quand $\mathbb{1}(\{\breve{x}_i \neq \tilde{x}_i\}) = 0$ et donc quand $\{\breve{x}_i = \tilde{x}_i\}$. Quand $\mathbb{1}(\{\breve{x}_i \neq \tilde{x}_i\}) = 1$, alors $x_i$ est différent soit de $\breve{x}_i$ soit de $\tilde{x}_i$ et donc un des deux $\mathbb{1}(\{x_i \neq \breve{x}_i\})$ ou $\mathbb{1}(\{x_i \neq \tilde{x}_i\})$ vaut 1 et l'autre est nul, ce qui finit la preuve en sommant sur tous les indices $i$.

Nous sommes maintenant en mesure de définir la distance minimale d'un code en bloc $\mathcal{C}$.

**Définition 1.4 (Distance minimale)** *La distance minimale $d_{\min}$ du code en bloc $\mathcal{C}$ est*

$$d_{\min}(\mathcal{C}) = \min_{\substack{\mathbf{c},\tilde{\mathbf{c}}\in\mathcal{C} \\ \mathbf{c}\neq\tilde{\mathbf{c}}}} d_H(\mathbf{c},\tilde{\mathbf{c}}). \tag{1.3}$$

*et est donc la plus petite distance de Hamming entre deux mots distincts de $\mathcal{C}$.*

Les trois paramètres : la longueur $n$, la dimension $k$ et la distance minimale $d_{\min}$ caractérisent fortement un code en bloc $\mathcal{C}$, et notamment la classe des codes linéaires que nous verrons dans la suite. Pour cette raison on indique un tel code souvent (de façon ambigue) par $\mathcal{C}(n,k,d_{\min})$. Ainsi, dans l'exemple 1.1, le code à répétition sera noté $\mathcal{C}(3,1,3)$.

### 1.2.2 Capacité de correction d'erreur

Nous nous intéressons ici au nombre d'erreur qu'un code peut corriger. On dit que le canal introduit $\ell$ erreurs, si

$$d_H(\mathbf{c},\mathbf{y}) = \ell,$$

pour $\mathbf{c}$ le mot de code envoyé et $\mathbf{y}$ le mot reçu.

**Définition 1.5 (Correction d'erreur)** *On dit qu'un code en bloc $\mathcal{C}$* corrige $t$ erreurs *s'il existe un décodeur qui retrouve le mot de code envoyé pour toutes les configurations de $t$ ou moins d'erreurs.*

Considérons un code $\mathcal{C}$ et le décodeur du voisin le plus proche qui décide toujours le mot de code $\mathbf{c}$ le plus proche en distance de Hamming du mot reçu $\mathbf{y}$. Donc pour ne pas faire d'erreur au décodage quand $\mathbf{c}$ a été envoyé, il faut qu'il n'y ait pas un autre mot de code $\tilde{\mathbf{c}}$ plus proche ou à la même distance du mot reçu $\mathbf{y}$ que $\mathbf{c}$. Il faut donc :

$$d_H(\mathbf{c},\mathbf{y}) < d_H(\tilde{\mathbf{c}},\mathbf{y}), \quad \forall\tilde{\mathbf{c}}\in\mathcal{C}\backslash\{\mathbf{c}\}. \tag{1.4}$$

Or, par l'inégalité triangulaire et par la définition de la distance minimale $d_{\min}$ :

$$d_{\min} \leq d_H(\mathbf{c},\tilde{\mathbf{c}}) \leq d_H(\mathbf{c},\mathbf{y}) + d_H(\tilde{\mathbf{c}},\mathbf{y}), \quad \forall\tilde{\mathbf{c}}\in\mathcal{C}\backslash\{\mathbf{c}\}. \tag{1.5}$$

Dans le cas où

$$2d_H(\mathbf{c},\mathbf{y}) < d_{\min}, \tag{1.6}$$

alors en l'injectant dans (1.5), on obtient (1.4) ce qui signifie que le décodeur décide toujours le bon mot $\hat{\mathbf{c}} = \mathbf{c}$. On peut bien vérifier, par distinction des cas pair et impair de $d_{\min}$, que la valeur maximale de $d_H(\mathbf{c},\mathbf{y})$ qui satisfait (1.6) est $\left\lfloor \frac{d_{\min}-1}{2} \right\rfloor$. Ceci implique que la capacité de correction d'erreur est au moins

$$t \geq \left\lfloor \frac{d_{\min}-1}{2} \right\rfloor. \tag{1.7}$$

Nous allons maintenant montrer, par l'absurde, que la capacité de correction d'erreur $t$ ne peut pas dépasser $\left\lfloor \frac{d_{\min}-1}{2} \right\rfloor$. Considérons un code $\mathcal{C}$ et n'importe quel algorithme de décodage qui peut corriger $t$ erreurs pour un $t$ qui satisfait

$$d_{\min} > t \geq \left\lfloor \frac{d_{\min}-1}{2} \right\rfloor + 1. \tag{1.8}$$

En examinant les cas pair et impair de $d_{\min}$, on note que (1.8) implique

$$d_{\min} - t \leq t. \tag{1.9}$$

Soient les deux mots de code $\mathbf{c}$ et $\tilde{\mathbf{c}}$ de distance minimale $d_{\min}$ et le mot reçu $\mathbf{y}$ tel que $d_H(\mathbf{c},\mathbf{y}) = t$ et $d_H(\tilde{\mathbf{c}},\mathbf{y}) = d_{\min} - t$. (Il suffit de prendre pour $\mathbf{y}$ tous les $n - d_{\min}$ symboles qui sont communs à $\mathbf{c}$ et $\tilde{\mathbf{c}}$, les $\ell$ premiers symboles de $\mathbf{c}$ qui diffèrent de $\tilde{\mathbf{c}}$, et le reste des symboles de $\tilde{\mathbf{c}}$.) Comme le décodeur peut corriger $t$ erreurs, alors lorsque $\mathbf{y}$ est reçu il devrait décider que le mot $\mathbf{c}$ a été envoyé. Mais comme $d_{\min} - t \leq t$, il devrait aussi décider que le mot $\tilde{\mathbf{c}}$ a été envoyé. Un tel décodeur ne peut évidemment pas exister, ce qui amène à une contradiction.

Nous avons donc démontré le résultat suivant.

**Résultat 1.2 (Capacité de correction d'un code)** *Un code de distance minimale $d_{\min}$ peut corriger $t$ erreurs (au sens de la définition 1.5) si $t$ vérifie*

$$t = \left\lfloor \frac{d_{\min} - 1}{2} \right\rfloor$$

*où $\lfloor u \rfloor$ est la partie entière de $u$.*

Ainsi, pour corriger une erreur simple, il faut une distance minimale au moins égale à 3. Pour 2 erreurs, il faut une distance au moins égale à 5, etc.

### 1.2.3 Capacité de détection d'erreur

Les codes correcteurs d'erreur sont aussi souvent utilisés pour détecter des erreurs (on les retrouve alors dans plusieurs couches OSI, de la couche MAC à la couche TCP). Lorsqu'ils sont utilisés en détection d'erreur, le décodage se limite à répondre à une seule question : est-ce que le mot reçu est bien égal au mot de code envoyé ? Bref, est-ce que $\mathbf{y} = \mathbf{c}$ ?

Nous supposons donc le décodeur qui produit

$$\hat{\mathbf{c}} = \mathbf{y}, \quad \text{si } \mathbf{y} \in \mathcal{C}$$

et un symbole d'erreur $\Delta$ sinon.

**Définition 1.6** *On dit qu'un code en bloc $\mathcal{C}$ est capable de détecter $t'$ erreurs si le canal a introduit n'importe quelle configuration de $t'$ ou moins d'erreur et si $\hat{\mathbf{c}} = \mathbf{c}$ quand $\mathbf{y} \in \mathcal{C}$.*

On en arrive rapidement au résultat suivant.

**Résultat 1.3 (Capacité de détection d'un code)** *Un code en bloc de distance minimale $d_{\min}$ est capable de détecter $t'$ erreurs, si $t'$ vérifie*

$$t' = d_{\min} - 1.$$

En effet, on utilise simplement un décodeur qui déclare $\Delta$ à chaque fois que le mot reçu $\mathbf{y}$ n'appartient pas au code $\mathcal{C}$. Or, le codeur transmet un mot de code. Il y aura donc non-détection des erreurs lorsque le mot reçu est un autre mot de code. Un autre mot de code étant à une distance de Hamming au moins égale à $d_{\min}$ du mot émis, pour pouvoir toujours détecter les erreurs, il suffit que leur nombre reste strictement inférieur à $d_{\min}$.

## 1.3 Codes en bloc linéaires

Dans ce cours on s'intéresse davantage aux codes en bloc dits *linéaires* qui sont définis comme suit.

**Définition 1.7 (Codes en bloc linéaires)** *Un code en bloc $\mathcal{C}$ est dit linéaire si pour tous mots de code $\mathbf{c}, \mathbf{c}' \in \mathcal{C}$ la somme forme à nouveau un mot de code :*

$$\mathbf{c} + \mathbf{c}' \in \mathcal{C}. \tag{1.10}$$

Dans cette définition, l'addition des deux mots se fait par composante et reste à définir plus précisément.

D'habitude on choisit l'alphabet du code $\mathcal{X}$, l'addition par composante $+$ et une multiplication $\times$ de façon à que le triple $(\mathcal{X}, +, \times)$ forme un corps fini $\mathbb{F}_q$, pour $q$ un nombre entier positif indiquant le cardinal du corps. Pour rappel, un corps fini est un alphabet fini $\mathcal{X}$ muni d'une addition et d'une multiplication ayant la propriété algébrique d'un corps. L'élément neutre de l'addition sera noté 0 et l'élément neutre de la multiplication sera noté 1. $(\mathcal{X}, +)$ est un groupe fini commutatif et $(\mathcal{X} \setminus \{0\}, \times)$ est un groupe fini qui est d'emblée commutatif car tout corps fini est commutatif. De plus, la multiplication est distributive pour l'addition. Le cardinal d'un corps fini $q$ est toujours une puissance d'un nombre premier, $q = p^r$. Si $r = 1$, alors l'addition et la multiplication du corps $\mathbb{F}_p$ sont définies de la même façon que sur $\mathbb{Z}/p\mathbb{Z}$, c'est-à-dire, l'addition et la multiplication modulo $p$.

| + | 0 | 1 |
|---|---|---|
| 0 | 0 | 1 |
| 1 | 1 | 0 |

| × | 0 | 1 |
|---|---|---|
| 0 | 0 | 0 |
| 1 | 0 | 1 |

TABLE 1.3 – Addition et multiplication sur $\mathbb{F}_2$

Par simplicité, dans ce cours on se restreint à l'alphabet binaire $\mathcal{X} = \{0, 1\}$ et au corps $\mathbb{F}_2$, où l'addition et la multiplication sont celles des entiers modulo 2 détaillées dans la table de vérité en bas. Mais dans la théorie de codage d'autres corps finis sont très important [1].

Considérons l'exemple suivant.

**Exemple 1.2** *Soit le code* $\mathcal{C} = \{(0\,0\,0\,0), (1\,0\,0\,1), (1\,1\,1\,0), (0\,1\,1\,1)\}$. *Notons d'abord que pour ce code* $n = 4, k = 2$ *et* $d_{\min} = 2$. *On vérifie bien aussi que le code est linéaire avec l'addition modulo 2 dans la table 1.3. En effet :*

$$\begin{aligned}
(0\,0\,0\,0) + (0\,0\,0\,0) &= (0\,0\,0\,0)\\
(0\,0\,0\,0) + (1\,0\,0\,1) &= (1\,0\,0\,1)\\
(0\,0\,0\,0) + (1\,1\,1\,0) &= (1\,1\,1\,0)\\
(0\,0\,0\,0) + (0\,1\,1\,1) &= (0\,1\,1\,1)\\
(1\,0\,0\,1) + (1\,0\,0\,1) &= (0\,0\,0\,0)\\
(1\,0\,0\,1) + (1\,1\,1\,0) &= (0\,1\,1\,1)\\
(1\,0\,0\,1) + (0\,1\,1\,1) &= (1\,1\,1\,0)\\
(1\,1\,1\,0) + (1\,1\,1\,0) &= (0\,0\,0\,0)\\
(1\,1\,1\,0) + (0\,1\,1\,1) &= (1\,0\,0\,1)\\
(0\,1\,1\,1) + (0\,1\,1\,1) &= (0\,0\,0\,0)
\end{aligned}$$

Notons que tout code linéaire binaire doit contenir le mot de code nul, c'est-à-dire le mot de code avec que des 0. En effet, la somme modulo 2 d'un mot de code avec lui-même donne comme résultat toujours le mot nul.

Un code linéaire binaire $\mathcal{C}$ est donc un sous-ensemble de $\{0, 1\}^n$ qui est clos par combinaisons linéaires et contient le mot nul. Ceci nous mène au résultat suivant.

**Résultat 1.4** *Un code en bloc linéaire binaire de longueur* $n$ *est un sous-espace vectoriel de* $\mathbb{F}_2^n$. *Et la dimension* $k$ *du code* $\mathcal{C}$ *est égale à la dimension du sous-espace.*

Ce résultat nous sera utile dans la suite car il nous permet d'utiliser des propriétés bien connues des sous-espace vectoriels.

### 1.3.1 Codeur de codes linéaires : notion de matrices génératrices

Tout sous-espace vectoriel de dimension finie contient une *base*, c'est-à-dire, un ensemble de $k$ vecteurs ($k$ indiquant la dimension du sous-espace), tel que chaque vecteur du sous-espace peut être obtenu par une combinaison linéaire de ces $k$ vecteurs. Comme un code binaire linéaire est un sous-espace de $\mathbb{F}_2^n$, il existe une base du code $\mathcal{C}$,

$$\mathcal{B} = (\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_k), \quad \mathbf{e}_i \in \mathcal{C},$$

tel que

$$\mathbf{c} = \sum_{i=1}^{k} d_i \mathbf{e}_i, \; d_i \in \mathbb{F}_2, \qquad \forall \mathbf{c} \in \mathcal{C}.$$

En général, il y a plusieurs choix possibles pour la base $\mathcal{B}$ et les coefficients $d_1, \ldots, d_k$ qui correspondent.

Ces observations nous permettent de décrire le code par une simple matrice de $k$ lignes et de $n$ colonnes et plus important encore de décrire la fonction de codage sous forme d'un produit matriciel.

1. Les codes de Reed-Solomon sont des codes très utilisés dans de nombreux systèmes. Ils sont construits, dans la plupart des cas, sur le corps $\mathbb{F}_{256}$

**Définition 1.8 (Matrice génératrice)** *Soit $\mathcal{B} = (\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_k)$ une base du code $\mathcal{C}$. On appelle matrice génératrice du code $\mathcal{C}$ toute matrice $G$ à $k$ lignes et $n$ colonnes qui s'écrit sous la forme,*

$$G = \begin{bmatrix} \mathbf{e}_1 \\ \mathbf{e}_2 \\ \vdots \\ \mathbf{e}_k \end{bmatrix}.$$

*Tout mot de code $\mathbf{c} \in \mathcal{C}$ peut s'écrire alors*

$$\mathbf{c} = \mathbf{d} \cdot G \tag{1.11}$$

*où $\mathbf{d}$ est le mot d'information qui, après passage dans le codeur, donnera le mot de code $\mathbf{c} \in \mathcal{C}$.*

Notons que la matrice $G$ est de rang $k$ puisque la base $\mathcal{B}$ engendre le code $\mathcal{C}$, sous-espace vectoriel de dimension $k$. De plus, le produit matriciel (1.11) décrit un *codeur linéaire*. En effet, si les mots d'information $\mathbf{d}$ et $\mathbf{d}'$ sont codés par $\mathbf{c}$ et $\mathbf{c}'$, alors le mot d'information $\tilde{\mathbf{d}} = \mathbf{d} + \mathbf{d}'$ est codé par la somme $\tilde{\mathbf{c}} = \mathbf{c} + \mathbf{c}'$.

Nous discutons maintenant de quelques exemples de codes.

**Exemple 1.3 (Code à répétition)** *Un* code à répétition *de longueur $n$ est l'ensemble du mot nul et du mot avec que des 1. Par exemple, pour $n = 5$, le code à répétition est l'ensemble des deux mots*

$$(0\,0\,0\,0\,0), \qquad (1\,1\,1\,1\,1).$$

*Il est facile de vérifier qu'un code à répétition est toujours linéaire. La dimension du code à répétition est égale à $\log_2 |\mathcal{C}| = 1$ et la distance minimale est égale à $d_{\min} = n$. Le code contient un seul mot de code non-nul qui forme donc une base pour le code. Par conséquence,*

$$G = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

*est la seule matrice génératrice pour le code à répétition de longueur 5.*

**Exemple 1.4 (Code de parité)** *Un* code de parité *de longueur $n$ est l'ensemble de tous les mots de longueur $n$ qui ont un poids de Hamming pair. Par exemple, pour $n = 3$, le code de parité est l'ensemble des quatre mots*

$$(0\,0\,0), \qquad (0\,1\,1), \qquad (1\,0\,1), \qquad (1\,1\,0).$$

*Le code de parité $\mathcal{C}$ est linéaire car :*

1. *Le mot $\mathbf{0}$ est un mot de poids pair donc $\mathbf{0} \in \mathcal{C}$.*
2. *La somme de deux mots de poids pair est un mot de poids pair.*

*Comme il y a autant de mots binaires de longueur $n$ avec poids de Hamming pair qu'avec poids de Hamming impair, un code de parité contient $2^{n-1}$ mots de code. Sa dimension est donc égale à $\log_2 |\mathcal{C}| = n-1$.*

*La distance minimale du code de parité est $d_{\min} = 2$, indépendamment de la valeur de $n$. Pour voir cela, on note que le mot qui commence par deux symboles 1 et puis se poursuit par que des 0 est un mot du code. Sa distance au mot de code nul est égale à deux, et donc $d_{\min} \leq 2$. D'un autre côté, $d_{\min} \geq 2$ car parmi deux mots de code qui sont à distance 1, il y en a forcément un avec un poids de Hamming impair. On conclut donc que $d_{\min} = 2$.*

*Pour l'exemple de $n = 3$, une base $\mathcal{B}$ du code peut être $\mathcal{B} = \{(1\,1\,0), (0\,1\,1)\}$, ce qui correspond à la matrice génératrice*

$$G = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix}.$$

**Codes équivalents et matrice génératrice systématique** Pour un code linéaire donné, $\mathcal{C}$, et parmi toutes les matrices génératrices possibles, il en est une qui est très fréquemment (presque toujours) utilisée. Il s'agit de la *matrice génératrice systématique*. Pour bien comprendre ce qu'elle est, il nous faut, tout d'abord, introduire la notion de codes équivalents.

Soient $\mathcal{C}$ et $\tilde{\mathcal{C}}$ deux codes de même longueur $n$. On dit que $\mathcal{C}$ est *équivalent* à $\tilde{\mathcal{C}}$ si et seulement si il existe une permutation des composantes $\sigma$ telle que, à tout mot $\mathbf{c}$ du code $\mathcal{C}$ correspond un mot $\tilde{\mathbf{c}}$ du code $\tilde{\mathcal{C}}$ vérifiant,

$$(\tilde{c}_1, \tilde{c}_2, \ldots, \tilde{c}_n) = \left(c_{\sigma(1)}, c_{\sigma(2)}, \ldots, c_{\sigma(n)}\right).$$

Evidemment, $\mathcal{C}$ et $\tilde{\mathcal{C}}$ ont la même dimension, la même distance minimale, ..., et seront en fait considérés comme deux codes identiques ce qui est motivé par le fait qu'ils ont la même performance quand ils sont utilisés pour la transmission sur un canal sans mémoire.

Soit maintenant $G$ une matrice génératrice du code $\mathcal{C}$. Donc $G$ correspond à une certaine base $\mathcal{B}$. Une autre matrice génératrice $G'$, du même code, correspond à une autre base $\mathcal{B}'$ du même sous-espace vectoriel. Les lignes de la matrice génératrice sont les vecteurs de la base choisie. On note que si on remplace un vecteur d'une base par une combinaison linéaire de ce même vecteur avec d'autres vecteurs de la base on obtient une nouvelle base. On peut donc facilement changer la matrice génératrice d'un code en remplaçant une ligne de la matrice génératrice par une combinaison linéaire de cette ligne avec d'autres lignes de la matrice. Cette opération sur les lignes d'une matrice génératrice ne change pas le code lui-même.

Dans la suite on s'autorise à changer de code si le nouveau code est équivalent au premier. Or, passer d'un code à un code équivalent à celui-ci revient à trouver une permutation des composantes des mots de code. En terme de matrice génératrice, ceci est équivalent à permuter les colonnes de $G$.

Pour résumer, deux opérations sont permises sur la matrice génératrice $G$ pour trouver une autre matrice génératrice du même code (ou d'un code équivalent) :

- Remplacement d'une ligne par une combinaison linéaire de cette ligne avec d'autres lignes. [2]
- Permutations de colonnes.

Or, il existe un algorithme (le pivot de Gauss) qui autorise ces manipulations sur une matrice quelconque et dont la sortie est la forme réduite de la matrice sous forme échelonnée. Si le rang de la matrice initiale est $k$ (ce qui est notre cas, puisque $G$ engendre un sous-espace de dimension $k$), alors la matrice échelonnée obtenue en sortie de l'algorithme sera la suivante

$$\left[\begin{array}{c|c} I_k & P \end{array}\right]$$

où $I_k$ est la matrice identité de rang $k$ et $P$ est une matrice dépendant du code considéré. Cette forme échelon est la matrice génératrice systématique du code $\mathcal{C}$ (en fait il s'agit bien souvent d'un code équivalent à $\mathcal{C}$).

**Définition 1.9 (Matrice génératrice systématique)** *Soit $\mathcal{C}$ un code linéaire de longueur $n$ et de dimension $k$. On appelle matrice génératrice systématique du code $\mathcal{C}$ toute matrice obtenue à la sortie du pivot de Gauss appliqué à une matrice génératrice $G$ quelconque du code $\mathcal{C}$. Une matrice génératrice systématique est sous la forme*

$$G_s = \left[\begin{array}{c|c} I_k & P \end{array}\right]$$

*où $P$ dépend du code $\mathcal{C}$.*

Une matrice génératrice systématique permet en fait de simplifier le codeur et le décodeur de $\mathcal{C}$. Examinons ici le codeur (sous forme systématique). Si $\mathbf{d}$ est le mot d'information en entrée du codeur systématique, alors le mot de code en sortie sera

$$\begin{aligned} \mathbf{c} &= \mathbf{d} \cdot G_s \\ &= \mathbf{d} \cdot \left[\begin{array}{c|c} I_k & P \end{array}\right] \\ &= \left[\underbrace{\mathbf{d}}_{\text{information}} \; \underbrace{\mathbf{d} \cdot P}_{\text{parité}}\right]. \end{aligned}$$

Ainsi si on utilise un codeur systématique, le mot de code correspondant à $\mathbf{d}$ comporte deux parties. Les $k$ premiers bits sont les bits d'information (égaux aux bits du mot d'information $\mathbf{d}$) alors que les $(n-k)$ bits

2. Il est nécessaire que dans la combinaison linéaire le coefficient de la ligne à remplacer est non-nul.

restants dépendent de $\mathbf{d}$ et du code et sont appelés bits de parité. En pratique, $(n-k)$ est petit devant $n$. Le codeur ne doit plus calculer que $(n-k)$ bits au lieu de $n$ pour un codeur quelconque.

### 1.3.2 Code dual et matrice de contrôle de parité

Deux mots $\mathbf{x} = (x_1, \ldots, x_n)$ et $\tilde{\mathbf{x}} = (\tilde{x}_1, \ldots, \tilde{x}_n)$ de longueur $n$ sont dits **orthogonaux** si et seulement si

$$\mathbf{x} \cdot \tilde{\mathbf{x}}^\top = \sum_{i=1}^{n} x_i \tilde{x}_i = 0.$$

Attention, les symboles sont dans $\mathbb{F}_2$ et l'orthogonalité ici est bien différente du cas de l'espace euclidien. Tout mot de poids de Hamming pair est orthogonal à lui-même !

**Définition 1.10** *Le* code dual *du code $\mathcal{C}$ sur l'alphabet $\mathcal{X}$, que l'on note $\mathcal{C}^\perp$, est l'ensemble des mots de longueur $n$ orthogonaux à tous les mots de $\mathcal{C}$ :*

$$\mathcal{C}^\perp := \left\{ \mathbf{x} \in \mathcal{X}^n : \ \mathbf{x} \cdot \mathbf{c}^\top = 0, \ \forall \mathbf{c} \in \mathcal{C} \right\}.$$

Si $\mathcal{C}$ est un code linéaire de longueur $n$ et de dimension $k$, alors $\mathcal{C}^\perp$ est un code linéaire de longueur $n$ aussi et de dimension $n-k$. Mais contrairement au cas de l'espace euclidien, le code dual peut avoir une intersection non nulle avec le code de départ. Ainsi, il existe des codes auto-duaux, c'est-à-dire, des codes qui vérifient $\mathcal{C} = \mathcal{C}^\perp$.

A titre d'exercice, vous pouvez vérifier que le code dual d'un code de parité de longueur $n$ est tout simplement le code à répétition de longueur $n$.

**Définition 1.11 (Matrice de contrôle de parité)** *Soit $\mathcal{C}$ un code de longueur $n$ et de dimension $k$. Une matrice de contrôle de parité de $\mathcal{C}$ est toute matrice $H$ qui est matrice génératrice de $\mathcal{C}^\perp$. Ainsi $H$ est une matrice à $n-k$ lignes et $n$ colonnes de rang $n-k$.*

**Résultat 1.5** *Soit $G$ une matrice génératrice de $\mathcal{C}$. Toute matrice $H$ de rang $n-k$, à $n-k$ lignes et $n$ colonnes, qui vérifie*

$$G \cdot H^\top = \boldsymbol{0} \tag{1.12}$$

*est une matrice de contrôle de parité de $\mathcal{C}$.*

L'idée de la preuve de ce résultat est la suivante :

- Tout mot de $\mathcal{C}$ s'écrit $\mathbf{d} \cdot G$ où $\mathbf{d}$ est un mot d'information, c'est-à-dire une matrice ligne à $k$ colonnes.
- Tout mot de $\mathcal{C}^\perp$ s'écrit $\tilde{\mathbf{d}} \cdot H$ où $\tilde{\mathbf{d}}$ est un mot d'information, c'est-à-dire une matrice ligne à $n-k$ colonnes.
- Tout mot de $\mathcal{C}$ est orthogonal à tout mot de $\mathcal{C}^\perp$,

$$\mathbf{d} \cdot G \cdot H^\top \cdot \tilde{\mathbf{d}}^\top = 0, \ \forall \mathbf{d}, \tilde{\mathbf{d}}.$$

- On en déduit donc $G \cdot H^\top = 0$.

Une conséquence immédiate du résultat 1.5 et de la définition 1.11 est la suivante.

**Résultat 1.6** *Pour $H$ une matrice de contrôle de parité du code $\mathcal{C}$, on a*

$$\mathbf{c} \cdot H^\top = \mathbf{0} \quad \Longleftrightarrow \quad \mathbf{c} \in \mathcal{C}. \tag{1.13}$$

*A partir de la matrice de contrôle de parité $H$ on peut donc facilement tester si un mot $\mathbf{c}$ appartient au code $\mathcal{C}$ ou pas.*

Il existe une matrice de contrôle de parité qui se déduit facilement de la matrice génératrice systématique du code $\mathcal{C}$. Il s'agit de la *matrice de contrôle de parité systématique*. La matrice génératrice systématique s'écrit

$$G_s = \left[ \ I_k \quad \| \quad P \ \right]. \tag{1.14}$$

Définissons la matrice de contrôle de parité systématique,

$$H_s = \left[ \begin{array}{c|c} -P^\top & I_{n-k} \end{array} \right], \tag{1.15}$$

où la soustraction est effectuée sur le corps $\mathbb{F}_p$ en question. (Donc pour $p = 2$ cette opération n'a pas d'effet car $-0 = 0$ et $-1 = 1$.) On constate que $H_s$ a $(n-k)$ lignes, $n$ colonnes et qu'elle est de rang $n-k$. De plus, nous avons

$$G_s \cdot H_s^\top = -P + P = 0.$$

Donc $H_s$ vérifie l'équation (1.12). $H_s$ est bien une matrice de contrôle de parité du code $\mathcal{C}$. Notez bien que la matrice de contrôle de parité systématique a l'identité à droite et non à gauche comme la matrice génératrice systématique et les tailles des matrices identité ne sont pas les mêmes.

Nous introduisons la famille des codes de Hamming qui jouent un rôle important dans le codage [3] et qui sont définis par leur matrice de contrôle de parité.

**Définition 1.12 (Code de Hamming binaire)** *Soit $m \geq 3$ un entier. Un code de Hamming binaire est un code de longueur $2^m - 1$ et de dimension $2^m - m - 1$. Sa matrice de contrôle de parité contient, en tant que colonnes, tous les $m-$uplets binaires non nuls (il y en a bien $2^m - 1$).*

Nous présentons le code de Hamming pour $m = 3$. Nous choisissons une matrice de contrôle de parité systématique, par exemple,

$$H_s = \left[ \begin{array}{ccccccc} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{array} \right].$$

Notons que, en utilisant les équations (1.15) et (1.14), $H_s$ correspond à la matrice génératrice systématique

$$G_s = \left[ \begin{array}{ccccccc} 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{array} \right].$$

### 1.3.3 Distance minimale et borne de Singleton

La linéarité du code $\mathcal{C}$ nous permet d'exprimer la distance minimale du code de façon plus simple. Nous revenons tout d'abord à sa définition que nous pouvons trouver dans l'équation (1.3).

Il est facile de voir que pour tout $\mathbf{x} \in \mathbb{F}_2^n$ et $\tilde{\mathbf{x}} \in \mathbb{F}_2^n$, on a

$$d_H(\mathbf{x}, \tilde{\mathbf{x}}) = w_H(\mathbf{x} - \tilde{\mathbf{x}}). \tag{1.16}$$

En remarquant que dans $\mathbb{F}_2^n$ l'addition de deux vecteurs est équivalente à leur différence, nous obtenons

$$d_{\min}(\mathcal{C}) = \min_{\substack{\mathbf{c},\tilde{\mathbf{c}} \in \mathcal{C}: \\ \mathbf{c} \neq \tilde{\mathbf{c}}}} d_H(\mathbf{c}, \tilde{\mathbf{c}}) = \min_{\substack{\mathbf{c},\tilde{\mathbf{c}} \in \mathcal{C}: \\ \mathbf{c} \neq \tilde{\mathbf{c}}}} w_H(\mathbf{c} + \tilde{\mathbf{c}}).$$

Or, le code étant linéaire, $\mathbf{c} + \tilde{\mathbf{c}}$ est aussi un mot de code, et nous obtenons finalement,

$$d_{\min}(\mathcal{C}) = \min_{\mathbf{c} \in \mathcal{C}, \mathbf{c} \neq \mathbf{0}} w_H(\mathbf{c}). \tag{1.17}$$

Ainsi, la recherche de la distance minimale d'un code quelconque, en utilisant la méthode exhaustive, demanderait le calcul de $|\mathcal{C}| \cdot (|\mathcal{C}| - 1)$ distances de Hamming alors qu'elle ne demande plus que le calcul de $(|\mathcal{C}| - 1)$ distances de Hamming dans le cas d'un code linéaire.

Il existe une méthode de calcul de la distance minimale d'un code linéaire encore moins complexe, comme décrit dans le résultat suivant.

**Résultat 1.7** *Pour tout code linéaire en bloc, $d_{\min}$ est égale au plus petit nombre de colonnes dépendantes de $H$.*

3. Ce sont notamment les premiers à avoir été publiés. La référence de cet article est la suivante : R. Hamming, *Error-detecting and error-correcting codes*, Bell Systems Technical Journal, 1950.

On obtient par le résultat 1.6 et l'égalité (1.17) :

$$d_{\min}(\mathcal{C}) = \min_{\substack{\mathbf{x}:\ \mathbf{x}\cdot H^\top = 0 \\ w_H(\mathbf{x})>0}} w_H(\mathbf{x}) \tag{1.18}$$

De plus on note que pour tout vecteur-ligne $\mathbf{x}$ tel que $\mathbf{x} \cdot H^\top = 0$ l'ensemble des colonnes de $H$ qui correspondent aux composantes non nulles de $\mathbf{x}$ sont linéairement dépendantes. Ceci prouve le résultat 1.7.

Nous regardons une application de ce résultat pour trouver $d_{\min}$ pour un exemple.

**Exemple 1.5** *On considère le code de Hamming de paramètre $m = 3$ (et donc de longueur 7) donné par sa matrice de contrôle de parité*

$$H = \begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}..$$

*Nous observons que toutes les colonnes sont différentes du vecteur nul. Donc $d_{\min}$ ne peut pas être 1. De plus, nous observons que toute paire de colonnes est linéairement indépendantes –en effet une paire de colonnes dépendantes signifie que ces deux colonnes sont identiques dans $\mathbb{F}_2$– et donc $d_{\min}$ ne peut pas être 2. Par contre les colonnes 1, 5, et 7 sont linéairement dépendantes car leur somme est égale au vecteur nul. Donc par le résultat 1.7, on a la distance minimale $d_{\min} = 3$.*

*Par le même raisonnement on peut montrer que la distance minimale de tout code de Hamming binaire est égale à $d_{\min} = 3$, indépendement de la valeur du paramètre $m$.*

*Le code de Hamming de paramètre $m \geq 3$ est donc un code $\mathcal{C}(2^m - 1, 2^m - m - 1, 3)$.*

Le résultat 1.7 nous permet aussi de trouver la relation fondamentale suivante qui est valable pour tout code linéaire.

**Résultat 1.8 (Borne de Singleton)** *Pour tout code linéaire en bloc $\mathcal{C}(n, k, d_{\min})$, on a*

$$d_{\min} \leq n - k + 1.$$

Chaque colonne de $H$ est de longueur $n-k$. N'importe quelles $n-k+1$ colonnes doivent donc être linéairement dépendantes. Par le résultat 1.7, ceci conclut la preuve.

## 1.4 Modèle des canaux discrets sans mémoire (DMC)

Dans ce chapitre nous introduisons un modèle de canal (c'est-à-dire une vision simplifiée de systèmes de communication) qui décrit la façon dont le canal change ou conserve les valeurs des symboles d'entrée. Ce modèle probabiliste est très général et admet même des entrées et des sorties sur des alphabets plus que binaires. Nous présentons ici un modèle de canal parmi les plus emblématiques dit *Canal Discret sans Mémoire* (*Discrete Memoryless Channel -DMC-*, en anglais).

Un DMC est une boîte (comme sur la figure 1.2) complètement défini par un triplet $(\mathcal{X}, \mathcal{Y}, P_{Y|X}(\cdot|\cdot))$, où

- $\mathcal{X}$ est un alphabet fini contenant toutes les valeurs possibles à l'entrée du DMC ;
- $\mathcal{Y}$ est un alphabet fini contenant toutes les valeurs possibles à la sortie du DMC ;
- $P_{Y|X}(\cdot|\cdot)$ est une loi de probabilité conditionnelle, dite loi de transition, décrivant comment une sortie $Y_t$ est obtenue à partir d'une entrée $x_t$.

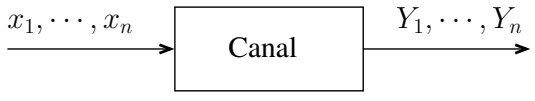



FIGURE 1.2 – Canal discret sans mémoire (DMC)

Cette boîte produit pour chaque séquence de $n$ symboles d'entrée $x_1, \ldots, x_n \in \mathcal{X}^n$ une séquence de $n$ symboles $Y_1, \ldots, Y_n \in \mathcal{Y}^n$ [4] où le symbole de sortie $Y_t$ est obtenu à partir du symbole d'entrée $x_t$ selon la *loi (de probabilité) de transition du DMC* $P_{Y|X}(\cdot|x_t)$, d'où,

$$Y_t \sim P_{Y|X}(\cdot|x_t). \tag{1.19}$$

4. Nous écrivons les symboles de sorties $Y_1, \ldots, Y_n$ en lettres majuscules parce que typiquement ils sont aléatoires même si les symboles d'entrées $x_1, \ldots, x_n$ ne le sont pas.

Donc, $Y_t$ dépend seulement de l'entrée $x_t$ et ni des entrées précédentes $x_1, \ldots, x_{t-1}$, ni des sorties précédentes $Y_1, \ldots, Y_{t-1}$, d'où le terme *sans mémoire*. Notez aussi que la loi de transition $P_{Y|X}(\cdot|\cdot)$ est la même pour tous les $t \in \{1, \ldots, n\}$ et le canal est donc *stationnaire*.

Souvent, la façon la plus facile de décrire un DMC est le *diagramme de canal*. Formellement, un diagramme de canal est un graphe bipartite, où à gauche on met toutes les valeurs possibles d'entrée (les éléments de $\mathcal{X}$) et à droite toutes les valeurs possibles de sortie (les éléments de $\mathcal{Y}$). On connecte ensuite chaque symbole d'entrée $x \in \mathcal{X}$ avec tous les symboles de sorties $y \in \mathcal{Y}$ selon la probabilité de transition $P_{Y|X}(y|x) > 0$. Cette probabilité est indiquée sur l'arête qui connecte $x$ et $y$. L'arête n'est pas dessinée lorsque la probabilité est nulle. Dans ce qui suit, nous verrons trois exemples de DMC et leur diagramme de canal.

**Exemple 1.6 (Canal binaire symétrique (BSC))** *Pour ce canal, les symboles d'entrée $x_1, \ldots, x_n$ et les symboles de sorties $Y_1, \ldots, Y_n$ prennent leurs valeurs dans $\{0, 1\}$. A chaque instant $t \in \{1, \ldots, n\}$, le symbole de sortie $Y_t$ est égal au symbole d'entrée $x_t$ avec une probabilité $1 - p \in (0, 1)$ et il est différent avec une probabilité $p$. La probabilité que le canal introduit une erreur à l'instant $t$ (cette probabilité est égale à $p$) ne dépend ni des bits émis ni des bits précédemment reçus.*

*Le BSC de paramètre $p \in [0, 1]$ est un DMC caractérisé par*

$$\mathcal{X} = \{0, 1\}, \qquad \mathcal{Y} = \{0, 1\}, \qquad P_{Y|X}(y|x) = \begin{cases} p, & y \neq x \\ 1 - p & y = x. \end{cases} \tag{1.20}$$

*La figure 1.3 montre le diagramme de canal du BSC. Notez qu'on peut toujours supposer que $p \leq 1/2$. Si ce*

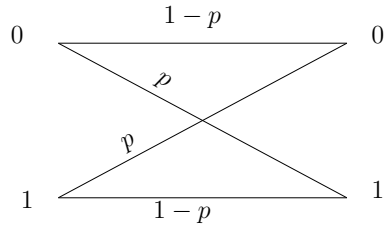



FIGURE 1.3 – Canal binaire symétrique

*n'est pas le cas, il suffit de changer, à la sortie du canal, tous les 0 en 1 et inversement.*

**Exemple 1.7 (Canal binaire à effacement (BEC))** *Pour ce canal, les symboles d'entrée $x_1, \ldots, x_n$ prennent leurs valeurs dans $\{0, 1\}$ et les symboles de sorties $Y_1, \ldots, Y_n$ prennent leurs valeurs dans $\{0, 1, \Delta\}$, où le symbole $\Delta$ représente un effacement. Ce canal n'introduit aucune erreur, mais il efface certains bits. Plus précisément, à chaque instant $t \in \{1, \ldots, n\}$ le symbole de sortie $Y_t$ est égal au symbole d'entrée $x_t$ avec une probabilité $1 - \epsilon$, où $\epsilon \in (0, 1)$, et il est égal au symbole d'effacement $\Delta$ avec une probabilité $\epsilon$. Ce canal est souvent utilisé comme modèle pour les réseaux du type internet. L'effacement modélise alors la perte d'un paquet.*

*Le BEC de paramètre $\epsilon \in [0, 1]$ est un DMC caractérisé par*

$$\mathcal{X} = \{0, 1\}, \qquad \mathcal{Y} = \{0, 1, \Delta\}, \qquad P_{Y|X}(y|x) = \begin{cases} 1 - \epsilon, & y = x \\ \epsilon & y = \Delta \\ 0 & \text{ailleurs.} \end{cases} \tag{1.21}$$

*La figure 1.4 montre le diagramme de canal du BEC.*

Dans ces deux premiers exemples, les DMCs sont à entrée binaire. L'émetteur envoie donc un seul bit par utilisation de canal. Naturellement on peut imaginer que l'émetteur envoie un paquet de plusieurs bits par utilisation du canal (par exemple 8 bits ce qui corresponderait à un octet –*byte*, en anglais–). Dans ce cas, l'alphabet d'entrée est $\mathcal{X} = \{0, 1\}^m$, où $m$ indique le nombre de bits envoyé par utilisation de canal. Un modèle de DMC fréquemment rencontré dans ce contexte est *le canal à effacement de paquet (Packet Erasure Channel)* où l'alphabet de sortie est $\mathcal{Y} = \mathcal{X} \cup \{\Delta\}$ et la loi de transition est celle indiquée dans (1.21). Comme déjà mentionné, les canaux à effacement sont souvent utilisés pour modéliser les réseaux de type internet.

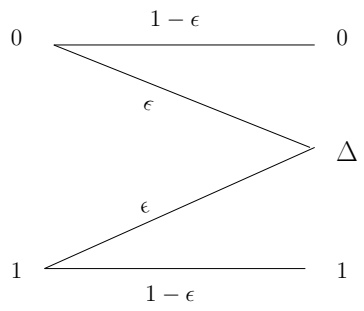



FIGURE 1.4 – Canal binaire à effacement

Les DMCs avec alphabet d'entrée de plus de deux valeurs sont aussi utilisés pour modéliser les communications numériques. A priori pour les modulations dites d'ordre supérieur que nous verrons dans le chapitre 2. L'exemple qui suit décrit un DMC qui modèlise une telle communication.

**Exemple 1.8 (Un canal avec 4 valeurs d'entrées et de sorties)** *Pour ce canal, les symboles d'entrée $x_1, \ldots, x_n$ et les symbols de sorties $Y_1, \ldots, Y_n$ prennent leurs valeurs dans $\{0, 1, 2, 3\}$. A chaque instant $t \in \{1, \ldots, n\}$, le symbole de sortie $Y_t$ est égal au symbole d'entrée $x_t$ avec une probabilité $1 - p \in (0, 1)$ et avec une probabilité $p$ il prend de façon équiprobable une valeur dans l'ensemble $\{x_t - 1, x_t + 1\}$ où les valeurs en dehors de $\{0, 1, 2, 3\}$ sont à ignorér.*

*Plus précisément, le DMC est caractérisé par*

$$\mathcal{X} = \{0, 1, 2, 3\}, \qquad P_{Y|X}(y|x) = \begin{cases} p/2, & |y - x| = 1 \text{ et } x \in \{1, 2\} \\ p & |y - x| = 1 \text{ et } x \in \{0, 3\} \\ 1 - p & y = x \end{cases} \tag{1.22}$$

*ou par le diagramme dans la figure 1.5.*

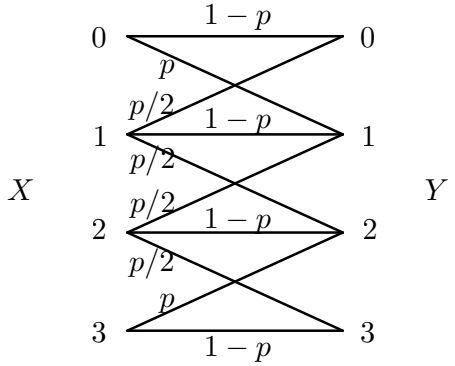



FIGURE 1.5 – Canal avec 4 valeurs d'entrée et de sortie

## 1.5 Décodage

La fonction de décodage $g(\cdot)$ produit un mot de bits détectés $\hat{\mathbf{d}} := (\hat{d}_1, \ldots, \hat{d}_k)$ à partir du mot reçu $\mathbf{y} = (y_1, \ldots, y_n)$ qui, on l'espère, sera identique aux bits d'information $d_1, \ldots, d_k$ transmis. Nous nous limitons à des décodeurs déterministes qui peuvent être décrits par une fonction

$$\begin{aligned} g\colon \mathcal{Y}^n &\to \{0,1\}^k \\ \mathbf{y} &\mapsto (\hat{d}_1, \ldots, \hat{d}_k). \end{aligned}$$

Comme vu précédemment, le décodage se fait typiquement en deux étapes, ou mathématiquement parlant, en enchaînant deux fonctions

$$g := g_2 \circ g_1$$

où $g_1$ et $g_2$ sont décrites ci-dessus.

1. La fonction $g_1$ trouve pour toute observation $\mathbf{y}$ le mot de code $\hat{\mathbf{c}} \in \mathcal{C}$ qui paraît être le plus probable (nous verrons que cela a un sens très précis). Donc $g_1$ prend la forme :

$$\begin{aligned} g_1 \colon \mathcal{Y}^n &\to \mathcal{C} \\ \mathbf{y} &\mapsto \hat{\mathbf{c}} \end{aligned}$$

2. La fonction $g_2$ produit la suite des bits détectées $\hat{d}_1, \ldots, \hat{d}_k$ qui est associée au mot de code $\hat{\mathbf{c}}$ identifié par $g_1$. Donc :

$$\begin{aligned} g_2 \colon \mathcal{C} &\to \{0,1\}^k \\ \hat{\mathbf{c}} &\mapsto (\hat{d}_1, \ldots, \hat{d}_n). \end{aligned}$$

On choisit toujours pour $g_2$ d'inverser la fonction de codage $f$ :

$$g_2 = f^{-1}.$$

Notons que nous avons supposé la fonction de codage $f$ bijective, ce qui implique que cette fonction peut être inversée.

Il reste alors à trouver des bons candidats pour $g_1$. Nous verrons que le choix optimal de $g_1$ dépend du canal et allons trouver ce $g_1$ optimal pour tout DMC. Mais auparavant nous notons que la fonction $g_1$ peut être décrite de manière équivalente par des régions de décision.

### 1.5.1 Régions de décision

Pour une fonction $g_1(\cdot)$ fixée et pour chaque mot de code $\mathbf{c} \in \mathcal{C}$, on définit la *région de décision associée à* $\mathbf{c}$ de la manière suivante

$$\Omega_{\mathbf{c}} := \big\{\mathbf{y} \in \mathcal{Y}^n \colon g_1(\mathbf{y}) = \mathbf{c}\big\}, \qquad \forall \mathbf{c} \in \mathcal{C}.$$

C'est donc l'ensemble de toutes les suites de symboles reçus $\mathbf{y}$ qui amène le récepteur à décider que le mot de code $\mathbf{c}$ a été envoyé.

Notons que les régions de décisions $\{\Omega_{\mathbf{c}}\}_{\mathbf{c}\in\mathcal{C}}$ forment une partition de $\mathcal{Y}^n$ :

$$\Omega_{\mathbf{c}} \ \cap \ \Omega_{\tilde{\mathbf{c}}} = \emptyset, \qquad \forall \mathbf{c}, \tilde{\mathbf{c}} \in \mathcal{C}, \ \mathbf{c} \neq \tilde{\mathbf{c}}$$

et

$$\bigcup_{\mathbf{c}\in\mathcal{C}} \Omega_{\mathbf{c}} = \mathcal{Y}^n.$$

La prochaine sous-section présente un choix des régions de décision qui minimise la probabilité d'erreur.

### 1.5.2 Décodage par maximum de vraisemblance

Nous commençons cette section par un exemple emblématique qui reprend le code à répétition traité auparavant.

**Exemple 1.9** *On suppose un canal BSC avec un paramètre $p \leq 1/2$ et on utilise encore le code à répétition de longueur 3 du tableau 1.1. Dans ce qui suit, nous allons justifier intuitivement le choix du décodeur indiqué dans le tableau 1.2.*

*Supposons que le mot reçu est $\mathbf{y} = (0\ 1\ 0)$. Nous analysons les scénarios $d_1 = 0$ et $d_1 = 1$ :*

- *Si $d_1 = 0$ et donc le mot $\mathbf{c} = (0\ 0\ 0)$ a été envoyé, alors le canal a changé un seul bit à la position 2. La probabilité que cela arrive est $p(1-p)^2$.*
- *Si $d_1 = 1$ et donc le mot $\mathbf{c} = (1\ 1\ 1)$ a été envoyé, alors le canal a changé deux bits aux positions 1 et 3. Ceci arrive avec une probabilité $p^2(1-p)$.*

*Dans le cas pratique où $p$ est proche de 0, $p \ll 1/2$, la probabilité $p(1-p)^2$ est beaucoup plus grande que $p^2(1-p)$. Il est donc beaucoup plus probable que le mot envoyé soit $\mathbf{c} = (0\ 0\ 0)$ au lieu de $\mathbf{c} = (1\ 1\ 1)$. C'est pour cette raison que le décodeur du tableau 1.2 déclare que le mot $(0\ 0\ 0)$ a été envoyé et donc $\hat{d}_1 = 0$ quand le mot reçu est $\mathbf{y} = (0\ 1\ 0)$.*

*Par les mêmes arguments, le décodeur du tableau 1.2 décide en faveur du mot de code $(0\ 0\ 0)$ ou $(1\ 1\ 1)$ selon le nombre de positions égales à 0 ou 1 dans le mot reçu $\mathbf{y}$. Pour le canal BSC, ce critère est en fait celui qui minimise la probabilité d'erreur par mot après décodage. Cette probabilité est définie précisément ci-après par* (1.24).

Soient $(D_1, \ldots, D_k)$ les $k$ bits d'informations. On rappelle que ces bits sont i.i.d. Bernoulli(1/2). Le décodeur fait une erreur quand

$$(D_1, \ldots, D_k) \neq (\hat{D}_1, \ldots, \hat{D}_k),$$

où $(\hat{D}_1, \ldots, \hat{D}_k)$ sont les symboles de bits d'information décidés par le récepteur. Comme nous supposons la fonction de codage $f(\cdot)$ bijective et la deuxième fonction de décodage $g_2 = f^{-1}$, cette condition est équivalente à une erreur dans la première étape du décodage :

$$\hat{\mathbf{C}} \neq \mathbf{C}. \tag{1.23}$$

La probabilité d'erreur est donc :

$$P_e := \Pr\left[\hat{\mathbf{C}} \neq \mathbf{C}\right]. \tag{1.24}$$

On appelle cette probabilité d'erreur aussi la probabilité d'erreur par mot après décodage, pour la distinguer de la probabilité d'erreur par bit, que nous verrons plus tard dans le chapitre.

On veut choisir les régions de décision de façon à minimiser cette probabilité d'erreur, ce qui revient à maximiser la probabilité de succès

$$P_c := \Pr\left[\hat{\mathbf{C}} = \mathbf{C}\right].$$

Nous développons

$$\begin{aligned} P_c &= \sum_{\mathbf{c}\in\mathcal{C}} \Pr\left[\hat{\mathbf{C}} = \mathbf{c} \middle| \mathbf{C} = \mathbf{c}\right] \cdot \Pr[\mathbf{C} = \mathbf{c}] \\ &= \sum_{\mathbf{c}\in\mathcal{C}} \Pr\left[\mathbf{Y} \in \Omega_{\mathbf{c}} \middle| \mathbf{C} = \mathbf{c}\right] \cdot \frac{1}{2^k} \\ &= \sum_{\mathbf{c}\in\mathcal{C}} \sum_{\mathbf{y}\in\Omega_{\mathbf{c}}} \Pr\left[\mathbf{Y} = \mathbf{y} \middle| \mathbf{C} = \mathbf{c}\right] \cdot \frac{1}{2^k}, \end{aligned}$$

où nous avons utilisé que tous les mots de code ont la même probabilité $1/2^k$ ce qui est obtenue par le fait que la fonction de codage $f$ est bijective.

On voit donc que pour maximiser la probabilité de décision correcte $P_c$, il faut choisir les régions de décision $\{\Omega_{\mathbf{c}}\}$ de telle façon que

$$\mathbf{y} \in \Omega_{\mathbf{c}} \quad \text{seulement si} \quad \Pr\left[\mathbf{Y} = \mathbf{y} \middle| \mathbf{C} = \mathbf{c}\right] = \max_{\tilde{\mathbf{c}}\in\mathcal{C}} \Pr\left[\mathbf{Y} = \mathbf{y} \middle| \mathbf{C} = \tilde{\mathbf{c}}\right]. \tag{1.25}$$

Si pour un $\mathbf{y}$ le maximum est atteint par plusieurs $\mathbf{c} \in \mathcal{C}$, alors on a le choix : tous les choix qui satisfont (1.25) donnent lieu à la même $P_c$ et donc à la même probabilité d'erreur.

On résume que la maximisation de $P_c$ revient à la maximisation par rapport à $\tilde{\mathbf{c}}$ de

$$\Pr\left[\mathbf{Y} = \mathbf{y} \middle| \mathbf{C} = \tilde{\mathbf{c}}\right]$$

que l'on nomme vraisemblance. La règle (1.25) est alors nommée "règle du maximum de vraisemblance" et nous avons le résultat suivant.

**Résultat 1.9 (Optimalité de la règle de maximum vraisemblance)** *Si les mots de code sont tous émis avec la même probabilité, minimiser la probabilité d'erreur $P_e$ revient à choisir le mot de code qui maximise la vraisemblance, c'est-à-dire, à choisir les régions de décision de la manière suivante*

$$\mathbf{y} \in \Omega_{\mathbf{c}} \quad \text{seulement si} \quad \Pr\left[\mathbf{Y} = \mathbf{y} \middle| \mathbf{C} = \mathbf{c}\right] = \max_{\tilde{\mathbf{c}}\in\mathcal{C}} \Pr\left[\mathbf{Y} = \mathbf{y} \middle| \mathbf{C} = \tilde{\mathbf{c}}\right].$$

Pour un DMC $(\mathcal{X}, \mathcal{Y}, P_{Y|X})$ et quand le mot de code est directement envoyé sur le canal sans modulation, $\mathbf{X} = \mathbf{C}$, la vraisemblance simplifie en

$$\Pr\left[\mathbf{Y} = \mathbf{y} \middle| \mathbf{C} = \mathbf{c}\right] = \prod_{i=1}^{n} P_{Y|X}\left(y_i|c_i\right). \tag{1.26}$$

**Exemple 1.10** *On considère le BSC de paramètre $p$ illustré dans la figure 1.3 avec $p \in [0, 1/2]$. Donc, $P_{Y|X}(y_i|c_i) = p$ si $y_i \neq c_i$ et $P_{Y|X}(y_i|c_i) = 1 - p$ dans le cas contraire. Nous obtenons de* (1.26) *:*

$$\begin{aligned}\Pr\left[\mathbf{Y} = \mathbf{y} \middle| \mathbf{C} = \mathbf{c}\right] &= \prod_{i=1}^{n} P_{Y|X}\left(y_i|c_i\right) \\ &= p^{d_H(\mathbf{y},\mathbf{c})}(1-p)^{n-d_H(\mathbf{y},\mathbf{c})} \\ &= (1-p)^n \cdot \left(\frac{p}{1-p}\right)^{d_H(\mathbf{y},\mathbf{c})}.\end{aligned}$$

*Si $p \in [0, 1/2)$, alors on a $p/(1-p) < 1$ ce qui nous permet de déduire le résultat suivant.*

**Résultat 1.10** *Pour tout $p \in [0, 1/2)$, le décodeur minimisant la probabilité d'erreur par mot pour un BSC de paramètre $p$ revient à trouver le mot de code $\mathbf{c} \in \mathcal{C}$ qui minimise $d_H(\mathbf{y}, \mathbf{c})$, la distance de Hamming entre le mot reçu $\mathbf{y}$ et $\mathbf{c}$. Cette règle s'appelle aussi* la règle du voisin le plus proche.

La règle de maximum vraisemblance est optimale pour pleins de problèmes de domaines autre que le décodage. En fait, dans tout problème de détection (détection d'un visage dans une image, détection de symboles au chapitre 2, reconnaissance du locuteur, détection d'anomalie, détection de présence, etc) où les éléments à détecter sont équiprobables. Si cette supposition n'est pas satisfaite, alors la règle plus générale dite *maximum a posteriori* est optimale.

Le seul problème avec la règle de maximum de vraisemblance (ou de maximum a posteriori) est sa complexité : en principe la règle nécessite de parcourir tous les mots du code (ou plus généralement tous les éléments à détecter). Dans la section suivante, nous expliquons une implémentation de la détection de maximum de vraisemblance pour les codes en bloc linéaires qui est souvent plus efficace. En pratique, ils existent aussi des algorithmes de décodage sous-optimaux plus rapides qui atteignent des performances très proches des performances optimales.

### 1.5.3 Décodage par syndrome

En sous-section 1.5.2 nous avons vu le décodeur optimal de maximum de vraisemblance. Malheureusement la recherche du mot de code qui maximise la vraisemblance ne peut pas se faire, en pratique, de façon exhaustive. Par exemple, dans les standards de TV numérique non HD, les codes utilisés sont des codes de Reed-Solomon dont le nombre de mots est $256^{144} \simeq 10^{115}$. A titre de comparaison, la partie visible de notre univers contiendrait environ $10^{80}$ atomes!! Il nous faut donc trouver des algorithmes qui effectuent cette recherche de façon la plus efficace possible.

Nous allons maintenant donner une méthode simple qui permet d'implémenter le décodeur de maximum de vraisemblance pour la transmission sur un BSC.

Nous introduisons d'abord la notion de syndrome. Pour ceci, nous pouvons nous aider de la figure 1.6. On voit que si le mot reçu est un mot de code, alors la projection orthogonale sur le code dual est égale à 0. Si le mot reçu n'est pas dans le code, elle n'est pas égale à 0. Cette projection orthogonale du mot reçu est appelée « syndrome ». Pour la calculer, il suffit d'évaluer les produits scalaires entre le mot reçu et les vecteurs d'une base du code dual qui sont les lignes de $H$. Le vecteur évalué sera donné comme combinaison linéaire des vecteurs de la base de $\mathcal{C}^{\perp}$. D'où la définition suivante,

**Définition 1.13 (Syndrome)** *Soit $\mathbf{y}$ un mot de longueur $n$ quelconque. On appelle syndrome la quantité suivante*

$$\mathbf{s} = \mathbf{y} \cdot H^{\top}$$

*où $H$ est une matrice de contrôle de parité de $\mathcal{C}$. Nous notons que $\mathbf{s}$ est un vecteur-ligne à $(n-k)$ colonnes.*

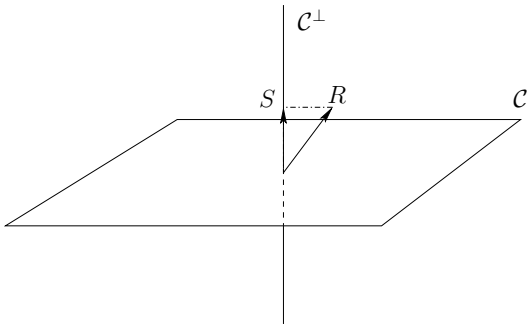



FIGURE 1.6 – Justification du syndrome : schéma de principe

En comparant la définition du syndrome avec le résultat 1.6, on obtient tout de suite :

$$\mathbf{y} \in \mathcal{C} \quad \text{si et seulement si} \quad \mathbf{s} = 0. \tag{1.27}$$

Nous développons maintenant un algorithme de décodage pour la communication sur un BSC grâce à la notion de syndrome. Pour cela, nous considérons que nous avons reçu un mot $\mathbf{y}$. Comme démontré dans le résultat 1.10, le décodage à maximum de vraisemblance pour un BSC revient à trouver le mot de code $\mathbf{c} \in \mathcal{C}$ qui minimise la distance de Hamming $d_H(\mathbf{c}, \mathbf{y})$ avec le mot reçu $\mathbf{y}$. Or, pour un canal BSC

$$\mathbf{y} = \mathbf{c} + \mathbf{e} \tag{1.28}$$

où $\mathbf{c}$ est le mot de code transmis et

$$\mathbf{e} = (e_1 \; e_2 \; \cdots \; e_n) \tag{1.29}$$

est le « mot d'erreur » avec $e_i = 1$ si le BSC a inversé le bit $i$, et $e_i = 0$ autrement. Par la linéarité de la distance de Hamming et la définition du poids de Hamming :

$$d_H(\mathbf{c}, \mathbf{y}) = d_H(\mathbf{c} - \mathbf{c}, \mathbf{y} - \mathbf{c}) = d_H(\mathbf{0}, \mathbf{e}) = w_H(\mathbf{e}),$$

et le décodage de maximum de vraisemblance revient à trouver le mot d'erreur avec poids de Hamming le plus petit parmi tous les mots $\tilde{\mathbf{e}}$ qui satisfont

$$\left(\mathbf{y} + \tilde{\mathbf{e}}\right) \in \mathcal{C}. \tag{1.30}$$

Donc, si $\mathbf{y} \in \mathcal{C}$, alors le décodage de maximum de vraisemblance déclare directement $\hat{\mathbf{c}} = \mathbf{y}$. Autrement, si un mot $\tilde{\mathbf{e}}$ avec poids de Hamming 1 satisfait (1.30) le décodage déclare $\hat{\mathbf{c}} = \mathbf{y} + \tilde{\mathbf{e}}$. Autrement, si un mot $\tilde{\mathbf{e}}$ avec poids de Hamming 2 satisfait (1.30) le décodage déclare $\hat{\mathbf{c}} = \mathbf{y} + \tilde{\mathbf{e}}$. Et ainsi de suite. Si à une certaine étape il y a plusieurs mots d'erreur avec le même poids qui satisfont (1.30), on choisit un mot $\tilde{\mathbf{e}}$ quelconque parmi ces possibilités.

Nous allons rendre la procédure décrite plus efficace en nous servant du syndrome

$$\mathbf{s} = \mathbf{y} \cdot H^\top.$$

Notons que

$$\mathbf{s} \;=\; \underbrace{\mathbf{c} \cdot H^\top}_{=0} + \mathbf{e} \cdot H^\top = \mathbf{e} \cdot H^\top$$

et donc le syndrome ne dépend pas du mot de code $\mathbf{c}$ envoyé mais seulement du mot d'erreur $\mathbf{e}$. Considérons les cas suivants :

— $\mathbf{e} = \mathbf{0}$ : Dans ce cas le syndrome est égal à

$$\mathbf{s} = \mathbf{0}$$

et le mot reçu correspond au mot de code envoyé : $\mathbf{y} = \mathbf{c}$.

— $w_H(\mathbf{e}) = 1$ : Soit $e_i = 1$ pour un index $i \in \{1, \ldots, n\}$. Dans ce cas, le syndrome est égal à

$$\mathbf{s} = \mathbf{h}_i^\top,$$

où $\mathbf{h}_i$ désigne la $i^{\text{ème}}$ colonne de $H$. Si on inverse le bit $y_i$ du mot reçu $\mathbf{y}$ on récupère le mot de code envoyé : $\mathbf{c} = (y_1, \ldots, y_{i-1}, 1 - y_i, y_{i+1}, \ldots, y_n)$.

— $w_H(\mathbf{e}) = 2$ : Soit $e_i = 1$ et $e_j = 1$ pour deux indices différents $i, j \in \{1, \ldots, n\}$. Dans ce cas le syndrome est égal à

$$\mathbf{s} = \mathbf{h}_i^\top + \mathbf{h}_j^\top,$$

et on obtient le mot de code envoyé $\mathbf{c}$ en inversant les deux bits $y_i$ et $y_j$ du mot reçu $\mathbf{y}$.

— $w_H(\mathbf{e}) = 3$ : Soit $e_i = 1, e_j = 1$ et $e_k = 1$ pour trois indices différents $i, j, k \in \{1, \ldots, n\}$. Dans ce cas le syndrome est égal à

$$\mathbf{s} = \mathbf{h}_i^\top + \mathbf{h}_i^\top + \mathbf{h}_k^\top$$

et on obtient le mot de code envoyé $\mathbf{c}$ en inversant les trois bits $y_i$, $y_j$, et $y_k$ du mot reçu $\mathbf{y}$.

— Et ainsi de suite.

Ceci nous amène à **l'algorithme de décodage par syndrome :**

1. Calculer le syndrome $\mathbf{s} = \mathbf{y} \cdot H^\top$.
2. Si $\mathbf{s} = \mathbf{0}$, alors on déclare $\hat{\mathbf{c}} = \mathbf{y}$ et l'algorithme se termine.
3. Vérifier si $\mathbf{s}^\top$ est égal à une colonne de $H$. Si $\mathbf{s}^\top = \mathbf{h}_i$, déclarer $\mathbf{c} = (y_1, \ldots, y_{i-1}, 1 - y_i, y_{i+1}, \ldots, y_n)$ et l'algorithme se termine. S'ils existent plusieurs $i$, en choisir un au hasard.
4. Vérifier si $\mathbf{s}^\top$ est égal à la somme de deux colonnes de $H$. Si $\mathbf{s}^\top = \mathbf{h}_i + \mathbf{h}_j$, déclarer $\mathbf{c} = (y_1, \ldots, y_{i-1}, 1 - y_i, y_{i+1}, \ldots, y_{j-1}, 1 - y_j, y_{j+1}, \ldots, y_n)$ et l'algorithme se termine. S'ils existent plusieurs $i$ et $j$, selon l'implémentation de l'algorithme, en choisir une paire au hasard ou déclarer une erreur de décodage.
5. et ainsi de suite on continue avec la somme de trois, quatre, etc. colonnes de $H$. (Les algorithmes pratiques souvent s'arrêtent après quelques étapes s'ils n'ont pas pu décoder et déclarent une erreur de décodage.)

Dans la sous-section 1.2.2 nous avons vu que tout code $\mathcal{C}$ peut corriger $\lfloor (d_{\min} - 1)/2 \rfloor$ erreurs si le code est utilisé avec le décodage du voisin le plus proche. Comme pour le BSC l'algorithme du décodage par syndrome applique exactement la règle du voisin le plus proche, nous en déduisons que pour tout code binaire $\mathcal{C}$ utilisé sur un canal BSC le décodage par syndrome peut corriger $\lfloor (d_{\min} - 1)/2 \rfloor$ erreurs.

Faisons un exemple.

**Exemple 1.11** *On considère un code de Hamming binaire avec $m = 3$. Rappelons que*

$$H_s = \begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}$$

*est une matrice de contrôle de parité du code.*

*Supposons que* $\mathbf{y} = (0\ 1\ 1\ 0\ 1\ 1\ 1)$. *Calculons le syndrome,*

$$\mathbf{s} = \mathbf{y} \cdot H_s^\top = (1\ 1\ 0) = \mathbf{h}_3^\top.$$

*L'algorithme de décodage par syndrome en déduit qu'il y a une erreur en position* 3*,*

$$\mathbf{e} = (0\ 0\ 1\ 0\ 0\ 0\ 0),$$

*et donc déclare*

$$\mathbf{c} = \mathbf{y} + \mathbf{e} = (0\ 1\ 0\ 0\ 1\ 1\ 1).$$

*Le codeur étant sous forme systématique, ceci correspond à la séquence de bits d'information* $\hat{\mathbf{d}} = (0\ 1\ 0\ 0)$.

## 1.6 Performances

Nous donnons à présent une borne supérieure de la probabilité d'erreur d'un code en bloc linéaire sur un canal binaire symétrique. Nous commençons par calculer la probabilité d'erreur par mot de code.

**Probabilité d'erreur par mot.** Supposons un code linéaire $\mathcal{C}$ de paramètres $(n, k, d_{\min})$. Il y a une erreur « mot » lorsque le mot décodé n'est pas égal au mot émis. Or, on sait qu'un code en bloc de distance minimale $d_{\min}$ corrige toutes les configurations d'erreurs dont le nombre est inférieur ou égal à $t$ (voir résultat 1.2), mais évidemment, il peut en corriger d'autres. Puisque le code est linéaire, on peut supposer que le mot $\mathbf{0}$ est émis. Dans ce cas, le mot reçu correspond aussi au mot d'erreur. On peut borner la probabilité d'erreur par mot de la façon suivante,

$$\boxed{P_{e,\text{mot}} \leq \sum_{i=t+1}^{n} C_n^i p^i (1-p)^{n-i}} \tag{1.31}$$

où $t = \lfloor (d_{\min} - 1)/2 \rfloor$ est la capacité de correction du code.

Dans l'équation (1.31), le terme $C_n^i p^i (1-p)^{n-i}$ représente la probabilité que le mot reçu soit de poids $i$ ($i$ bits égaux à 1, $n-i$ bits égaux à 0 et, bien sûr, $C_n^i$ le nombre de ces mots). Le membre de droite de cette équation n'est donc que la probabilité que le mot reçu soit de poids de Hamming strictement supérieur à $t$.

**Approximation de la probabilité d'erreur par bit d'information.** A partir de l'équation (1.31), on peut en déduire une approximation de la probabilité d'erreur par bit d'information après décodage. Dans l'équation (1.31), chaque terme de la somme correspond à un poids $i$ de l'erreur. Le premier terme,

$$C_n^{t+1} p^{t+1} (1-p)^{n-(t+1)}$$

est souvent dominant. Si un événement de ce type survient, cela veut dire que le mot reçu a un poids de $(t+1)$ et alors le mot décodé (qui n'est pas $\mathbf{0}$ car il y a erreur sur le mot décodé) est le mot de code le plus proche de $\mathbf{0}$ qui est, par définition, de poids $d_{\min}$. Donc, le mot décodé admet $d_{\min}$ bits en erreur sur les $n$ bits qui le composent. Ainsi, on peut approcher la probabilité d'erreur par bit d'information (en supposant que les bits d'information et de parité seront en erreur, après décodage, avec la même probabilité) par

$$P_b \simeq \frac{d_{\min}}{n} C_n^{t+1} p^{t+1} (1-p)^{n-(t+1)}.$$

De plus, si $p \ll 1$, on peut encore simplifier l'expression précédente (en disant que $(1-p) \simeq 1$) et obtenir que

$$P_b \simeq \frac{d_{\min}}{n} C_n^{t+1} p^{t+1}. \tag{1.32}$$

## 1.7 Bilan

Dans ce chapitre, nous vous avons donné un aperçu du codage de canal. Vous avez vu quels pouvaient être les critères pour construire un bon code ainsi que la technique optimale de décodage et une de ses implémentations via le décodage par syndrome. Néanmoins, le domaine du codage est, bien évidemment, beaucoup plus vaste que ce que l'on a découvert dans ce chapitre. En effet, de nombreuses familles de codes linéaires avec leurs avantages et inconvénients respectifs ont été créées au cours des cinquante dernières années : codes cycliques, codes de géométrie algébrique, codes convolutifs, turbo-codes, codes LDPC, codes sur réseaux de points, codes polaires, etc. De plus, leur décodage dépendant des canaux de communication, il y a de nombreux algorithmes de décodage possibles avec de nouveau leurs performances et complexité respectives. Ceci sera étudié en profondeur en deuxième année.

Les deux chapitres suivants vont nous permettre de relier le codage à la problématique des communications. Le chapitre 2 va relier le codage à la modulation alorqs que le chapitre 3 se focalisera sur la théorie de l'information qui donnera les limites fondamentales que l'on ne peut pas dépasser quelques soient les techniques de codage et modulation utilisées.

Nous rappelons ci-dessous les concepts de base et savoir-faire concernant ce chapitre à acquérir durant cette unité d'enseignement.

**Les concepts de base :**

- Paramètres de dimensionnement d'un code : $n$, $k$, distance minimale de Hamming,
- Matrice génératrice, matrice de contrôle de parité,
- Décodage maximum de vraisemblance : application par le décodage par syndrome,
- Notion de capacité de correction et de capacité de détection.

**Les savoir-faire :**

- Déterminer les paramètres d'un code,
- Implémenter le décodage par syndrome (cf. TD et exercices ci-dessous).

## 1.8 Exercices

**Exercice 1.1** On se propose d'étudier des modifications qu'on peut effectuer sur un code correcteur d'erreur en bloc. Les modifications des codes sont souvent utilisées pour adapter le code au format des paquets à l'entrée du codeur imposé par les autres modules de la chaîne de transmission. On se propose d'étudier 4 modifications du code en bloc $C(6,3)$ de matrice génératrice :

$$G = \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 \end{bmatrix}$$

1. *Code étendu* : Afin de construire un code étendu à partir du code $C$, nous rajoutons un bit de parité (de redondance) à tous les mots de code. Ce bit de parité est tel que $c_7 = \sum_{i=1}^{6} c_i$.
   1.1 Donner la matrice $H$ et calculer la distance minimale de $C$.
   1.2 Donner les paramètres $n$ et $k$ du code étendu.
   1.3 Donner la matrice de parité $H_E$ du code étendu en partant de la matrice de parité $H$ de $\mathcal{C}$ (Rappel : tout mot de code $\mathbf{c} \in \mathcal{C}_E$ vérifie $\mathbf{c}.H_E^T = 0$).
   1.4 Écrire $H_E$ sous forme systématique et déduire la matrice génératrice du code étendu.
   1.5 L'extension du code a-t-elle permis d'augmenter la distance minimale ?

2. *Code rallongé* : Afin de construire un code rallongé à partir du code $C$, nous rajoutons un bit aux mots d'information et un bit aux mots de code.
   2.1 Donner les paramètres $n$ et $k$ du code rallongé.
   2.2 Le rallongement du code revient à faire quelle opération sur la matrice de parité $H$ de $C$.
   2.3 Donner la matrice de parité du code rallongé qui donne la meilleure $d_{min}$ . De quel code s'agit-il ?

3. *Code raccourci* : Afin de construire un code raccourcis à partir du code $C$, nous supprimons un bit aux mots d'information et un bit aux mots de code.
   3.1 Donner les paramètres $n$ et $k$ du code raccourcis.
   3.2 Le raccourcissement du code revient à faire quelle opération sur la matrice de parité $H$ de $C$.
   3.3 Donner la matrice de parité du code raccourcis qui donne la meilleure $d_{min}$.

4. *Code expurgé* : Afin de construire un code expurgé à partir du code $C$, nous allons considérer un sous ensemble de mots de code de $C$ que nous allons raccourcir.
   4.1 Nous considérons que les mots d'information ayant le premier bit égal à 0, donner le sous ensemble $S$ de mots de code de $C$ correspondant.
   4.2 Le code étant sous forme systématique et connaissant le premier bit d'information, comment peut-on raccourcir les mots de codes de $S$ ? Donner les paramètres du code expurgé.
   4.3 Proposer une méthode pour décoder le code expurgé en utilisant le décodage de $C$.

**Exercice 1.2** Dans les systèmes pratiques, une concaténation série ou parallèle de deux codes correcteurs d'erreurs est souvent utilisée. On se propose d'étudier la concaténation de deux codes correcteurs d'erreur.

*Concaténation parallèle :* On considère deux codes $C_1(n_1, k, d_{min1})$ et $C_2(n_2, k, d_{min2})$ sous formes systématiques. Le code $C_P$ résultant de la concaténation parallèle de $C_1$et $C_2$ est aussi sous forme systématique, ses mots de code s'écrivent sous la forme suivante :

$$C_P = [k \text{ bits d'information} \mid (n_1 - k) \text{ bits de redondance de } C_1 \mid (n_2 - k) \text{ bits de redondance de } C_2]$$

1. Soient le code $C_1(6,3)$ défini par sa matrice génératrice $G$ et $C_2$ le code de parité $(4,3)$.

$$G_{C_1} = \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 \end{bmatrix}$$

1.1 Calculer les distances minimales $d_{min,1}$ et $d_{min,2}$ respectivement de $C_1$ et $C_2$.
1.2 Quel est le rendement du code $C_P$ ?
1.3 Lister tous les mots de code de $C_P$ et donner sa distance minimale ?

2. Soient $C_1(n_1, k, d_{min,1})$ et $C_2(n_2, k, d_{min,2})$.
2.1 Quel est le rendement du code $C_P$ ?
2.2 Montrer que la distance minimale de $C_P$ vérifie :

$$\max(d_{min,1}, d_{min,2}) < d_{min} \leq \min(d_{min,1} + n_2 - k, d_{min,2} + n_1 - k).$$

*Concaténation série :* On considère deux codes $C_1(n_1, k, d_{min1})$ et $C_2(n_2, n_1, d_{min2})$ mis en série tel que les mots de code de $C_1$ sont les mots d'information de $C_2$.

1. Soient le code $C_1(6, 3)$ défini dans la partie concaténation parallèle et $C_2$ le code de parité $(7, 6)$.
1.1 Donner le rendement du code $C_S$ résultant de la concaténation série de $C_1$ et $C_2$.
1.2 Lister les mots code de $C_S$ et déterminer sa distance minimale.
2. Soient $C_1$ le code de parité $(3, 2)$ et $C_2$ le code $(6, 3)$ défini dans la partie concaténation parallèle.
2.1 Donner le rendement du code $C_S$ résultant de la concaténation série de $C_1$ et $C_2$.
2.2 Lister les mots code de $C_S$ et déterminer sa distance minimale.
3. Soient $C_1(n_1, k, d_{min,1})$ et $C_2(n_2, n_1, d_{min,2})$.
3.1 Donner le rendement de $C_S$ en fonction des rendements de $C_1$ et $C_2$ ?
3.2 Montrer que la distance minimale de $C_S$ vérifie :

$$d_{min} \geq \max(d_{min,1}, d_{min,2}).$$

**Exercice 1.3** *Soit le code systématique $\mathcal{C}$ défini par les équations de parité suivantes :*

$$\begin{aligned} r_1 &= d_2 + d_3 + d_4 \\ r_2 &= d_1 + d_2 + d_3 \\ r_3 &= d_1 + d_2 + d_4 \\ r_4 &= d_1 + d_3 + d_4 \end{aligned}$$

*où $[d_1, d_2, d_3, d_4]$ sont les bits d'informations et $[r_1, r_2, r_3, r_4]$ sont les bits de parité (bits de redondance) d'un mot de code. Un mot de code s'écrit donc $[d_1, d_2, d_3, d_4, r_1, r_2, r_3, r_4]$.*

1. *Trouver la longueur, la dimension, et le rendement de $\mathcal{C}$.*
2. *Donner une matrice génératrice $G$ et une matrice de contrôle de parité $H$ de $\mathcal{C}$ sous forme systématique.*
3. *Quel lien y-a-t-il entre le code $\mathcal{C}$ et le code $\mathcal{C}_{\text{carré}}$ trouvé dans la question 9. du TD 1 ?*
4. *Quelle est la distance minimale $d_{\min}$ de $\mathcal{C}$ ?*
5. *On décide de construire un nouveau code $\mathcal{C}'$ en ne gardant que les mots de code de poids de Hamming 4. Le code $\mathcal{C}'$ est-il un code linéaire ?*

**Exercice 1.4** *On se propose d'utiliser le code systématique $\mathcal{C}$ avec matrice génératrice*

$$G = \begin{bmatrix} 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 \end{bmatrix}$$

*sur un BSC($\epsilon$), où $0 < \epsilon < 1/2$.*

1. *Lister tous les mots du code $\mathcal{C}$.*
2. *Trouver la distance minimale de $\mathcal{C}$.*
3. *Décrire un décodeur optimal en utilisant le tableau suivant.*
4. *Trouver la probabilité d'erreur de votre décodeur optimal.*
5. *On suppose maintenant que le récepteur veut seulement récupérer le premier bit d'information. Quel est le décodeur optimal dans ce cas, et quelle est la probabilité d'erreur de ce nouveau décodeur ?*

| mot reçu $\mathbf{y}$ | mot décodé $\hat{\mathbf{c}}$ |
|---|---|
| 0 0 0 0 0 | |
| 0 0 0 0 1 | |
| 0 0 0 1 0 | |
| 0 0 0 1 1 | |
| 0 0 1 0 0 | |
| 0 0 1 0 1 | |
| 0 0 1 1 0 | |
| 0 0 1 1 1 | |
| 0 1 0 0 0 | |
| 0 1 0 0 1 | |
| 0 1 0 1 0 | |
| 0 1 0 1 1 | |
| 0 1 1 0 0 | |
| 0 1 1 0 1 | |
| 0 1 1 1 0 | |
| 0 1 1 1 1 | |
| 1 0 0 0 0 | |
| 1 0 0 0 1 | |
| 1 0 0 1 0 | |
| 1 0 0 1 1 | |
| 1 0 1 0 0 | |
| 1 0 1 0 1 | |
| 1 0 1 1 0 | |
| 1 0 1 1 1 | |
| 1 1 0 0 0 | |
| 1 1 0 0 1 | |
| 1 1 0 1 0 | |
| 1 1 0 1 1 | |
| 1 1 1 0 0 | |
| 1 1 1 0 1 | |
| 1 1 1 1 0 | |
| 1 1 1 1 1 | |

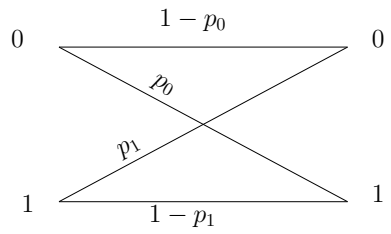



FIGURE 1.7 – Canal binaire non-symétrique

**Exercice 1.5** On considère un canal binaire **non-symétrique** (en anglais *Binary Non-Symmetric Channel -BNSC-*) comme donné sur le diagramme suivant avec $p_0$ la probabilité d'erreur sur le bit '0' et $p_1$ la probabilité d'erreur sur le bit '1'.

Ce canal intervient par exemple en fibre optique où le bit '0' n'émettant pas de lumière, il subit moins de modification que le bit '1' et donc $p_0 < p_1$.

1. Ecrire le maximum de vraisemblance associé au canal binaire non-symétrique quand on émet un mot de code (d'un code linéaire en bloc) $\mathbf{c} = [c_0, \ldots, c_{N-1}]$ et qu'on reçoit un mot $\mathbf{y} = [y_0, \cdots, y_{N-1}]$.
2. Indiquer la règle de décision ? Est-ce la règle du plus proche voisin (au sens de la distance de Hamming).
3. Pour un code à répétition de longueur 3, examiner le décodage du mot reçu $\mathbf{y} = [1, 0, 0]$ avec $p_0 = 0.1$ et $p_1 = 0.4$. Comparer au BSC. Qu'en conclure ?

**Exercice 1.6** On considère un canal BEC. On considère un code linéaire en bloc ($N$, $K$, $d_{\min}$).

1. Ecrire la règle du maximum de vraisemblance pour un mot reçu $\mathbf{y}$ (de longueur $n$).
2. Quelle est la probabilité d'erreur bit $P_b$ ? Comparer au cas BSC.

# Chapitre 2

# Modulations numériques

## 2.1 Introduction

Dans le chapitre 1, vous avez appris à coder des éléments binaires dans le but de les protéger contre les erreurs induites par une transmission sur un canal. Dans ce chapitre, les **objectifs principaux vont être de construire un dispositif permettant de transformer les bits codés, notés $c_i$ (cf. chapitre 1), provenant donc du codeur correcteur d'erreur en un signal attaquant le canal de propagation et vice-versa, et ensuite d'analyser les performances de ces dispositifs**. A titre d'exemple, dans le cadre de communications radio-mobiles (2G, 3G, 4G, 5G, TNT, Wifi), le signal à construire correspond à celui de l'onde électro-magnétique en sortie de l'antenne d'émission. Dans le cadre de communications filaires, on peut penser à l'onde électro-magnétique sortant du modem ADSL ou bien au signal lumineux rentrant dans la fibre optique. De manière moins naturelle, on peut songer à la problématique de la lecture des CD/DVD. Dans ce cas, le sillage sur le disque correspond au signal à construire. Cette liste d'exemples n'est évidemment pas exhaustive.

Ce que nous remarquons, grâce à ces exemples, est le point commun suivant : le signal à transmettre est par essence de nature **analogique** ou de manière synonyme à **temps continu** (pour les CD/DVD, cela serait plutôt une notion d'espace continu, mais le disque tournant, on peut se ramener sans difficulté à une notion temporelle), quand bien même nous faisons des **transmissions numériques**. Ainsi les signaux émis et reçu sont de nature analogique, mais l'information contenue dans ces signaux est de nature numérique ou de manière synonyme à temps discret et à nombre fini de valeurs.

Par conséquent, une grande partie du chapitre sera consacrée au passage de signaux à temps discret à des signaux à temps continu (au niveau de l'émetteur) et vice-versa (au niveau du récepteur). De plus, pour agir convenablement, le récepteur a besoin de savoir comment le signal émis a été perturbé par le canal de propagation. C'est pourquoi le plan de ce chapitre est comme suit :

- en section 2.2, nous introduisons le modèle de canal de propagation considéré dans ce chapitre. Ce modèle est dit « canal gaussien à temps continu ». Des modèles plus réalistes et donc plus compliqués seront étudiés en deuxième année.

- en section 2.3 nous présentons les paramètres d'interêt d'un système de communication et nous étudions un exemple simple d'un système.

- en section 2.4, nous nous plaçons du coté de l'émetteur. Après la présentation de la structure générale de nos signaux émis en sous-section 2.4.1, nous aborderons les deux points importants qui sont la transformation des bits en symboles en sous-section 2.4.2 et le passage du temps discret au temps continu avec la notion de filtre d'émission développé en sous-section 2.4.3. Nous montrerons notamment l'impact de ce filtre sur le spectre du signal émis et donc sur la bande utilisée pour la communication.

- en section 2.5, nous nous plaçons cette fois-ci du coté du récepteur. En premier nous présentons la structure optimale du récepteur en sous-section 2.5.1. Après, nous verrons comment éliminer les interférences entre symboles avec un choix judicieux du filtre d'émission en sous-section 2.5.2. Enfin, en sous-section 2.5.3, nous développerons le détecteur optimal par le biais de la théorie de la détection

optimale présentée au chapitre 1.

- en section 2.6, nous évaluerons théoriquement les performances du système non-codé passant par les boîtiers introduits tout au long de ce chapitre. L'extension au cas codé est également conduite faisant ainsi le lien avec le chapitre 1.

- en section 2.7, nous établissons les liens entre les différents paramètres de dimmensionnent que nous aurons introduit au fur et à mesure de ce chapitre et du chapitre précédent.

Enfin, en section 2.8, nous récapitulons les différentes notions présentées au cours de ce chapitre et les mettons en relation les unes avec les autres.

## 2.2 Canal de propagation

Dans le chapitre précédent nous avons considéré un canal à *temps discret*, c'est-à-dire, un canal qui prenait un symbole en entrée à chaque unité de temps et resortait une observation au niveau du récepteur à la même unité de temps. En réalité, les canaux de transmissions sont à *temps continu* car correspondent à la transmission d'une onde (électromagnétique, sonore, moléculaire, etc) qui admet une valeur à tout instant. Dans ce chapitre, nous allons donc expliquer (entre autre) comment passer d'un canal à temps continu à un canal à temps discret. Nous noterons

— $x(t)$ le signal (à temps continu, dit aussi analogique) émis. Il porte les bits codés $\{c_\ell\}_\ell$ et donc les données $\{d_\ell\}_\ell$.

— et $y(t)$ le signal analogique reçu. Ce signal correspond au signal $x(t)$ modifié par le canal de propagation. Donc $y(t) \neq x(t)$.

Nous nous focalisons sur un des canaux à temps continu les plus simples : le **canal gaussien à temps continu** qui est décrit par la relation suivante

$$y(t) = x(t) + b(t) \tag{2.1}$$

où le bruit $b(t)$ est supposé blanc et gaussien de moyenne nulle et densité spectrale de puissance $N_0/2$. Ce bruit provient au minimum du bruit thermique des composants électroniques du récepteur mais peut également intégrer le bruit de fond.

Ce modèle est notamment valide pour des communications entre stations de bases, des liaisons satellitaires, des réseaux câblés co-axiaux, des communications optiques dans l'espace libre, des communications proches à très hautes fréquences, etc.

## 2.3 Paramètres d'un système de modulation

Le but d'un système de modulation est de transmettre une suite de bits codés ou non-codés par un canal à temps continu, par exemple, le canal décrit dans la sous-section précédente. Dans un tel système on s'intéresse surtout aux quantités suivantes :

- la largeur de bande occupée par le signal émit $x(t)$ ; (la largeur de bande est souvent limitée pour éviter l'interférence entre plusieurs systèmes ou pour des raisons économiques)
- l'énergie consommée par le signal émit $x(t)$ ; (l'énergie consommée détermine la durabilité d'un système avant que ses batteries doivent être rechargées)
- le débit de communication atteint c'est-à-dire le nombre de bits d'informations transmis par unité de temps atteint ;
- la probabilité d'erreur au récepteur. (le débit et la probabilité d'erreur détermine par exemple la fiabilité et la qualité du service)

Dans le reste du chapitre nous allons décrire un émetteur et un récepteur, et analyserons ce système de communication par rapport à sa probabilité d'erreur, sa consommation énergétique et sa largeur de bande utilisée.

Nous commençons par un exemple de système.

**Exemple 2.1** *Considérons une suite de données provenant de la sortie du code correcteur d'erreur* $\{c_\ell\}_{\ell=1,\cdots,N} \in \{0,1\}^N$. *Dans cet exemple, le signal émis* $x(t)$ *est déterminé par le bit codé* $c_\ell$ *pendant l'intervalle* $[\ell T_s, (\ell+1)T_s)$ *avec* $T_s = 1$ *milliseconde : Si* $c_\ell = 0$ *alors* $x(t) = A$ *pendant toute la durée* $[\ell T_s, (\ell+1)T_s)$ *et si* $c_\ell = 1$ *alors* $x(t) = -A$ *pendant toute la durée* $[\ell T_s, (\ell+1)T_s)$.

*Sur la figure 2.1, nous avons tracé le signal $x(t)$ construit selon le principe évoqué ci-dessus pour la suite de données suivantes :* $0, 1, 1, 0, 0, 1, 0, 1, 0, 0$.

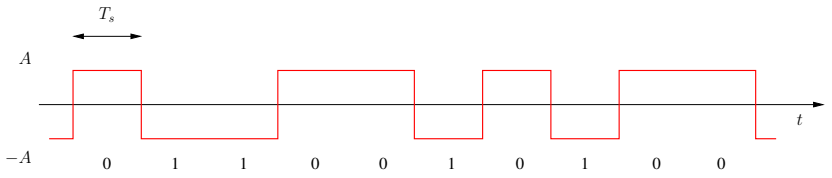



FIGURE 2.1 – $x(t)$ pour l'envoi de $N = 10$ données égales à $0, 1, 1, 0, 0, 1, 0, 1, 0, 0$

*On suppose que le signal est envoyé à travers un canal gaussien, comme décrit dans la sous-section 2.2. Donc* $y(t) = x(t) + b(t)$ *pour un bruit blanc* $b(t)$ *de densité spectrale de puissance égale à* $N_0/2$.

*Pour retrouver un symbole* $c_\ell$*, le récepteur intègre la tranche du signal reçu* $y(t)$ *entre* $\ell T_s$ *et* $(\ell + 1)T_s$*, vérifie si la valeur résultante est positive ou négative. Si elle est positive, le récepteur déclare* $\hat{c}_\ell = 0$ *et sinon il déclare* $\hat{c}_\ell = 1$.

*Nous verrons à la section 2.5 que ce choix du récepteur est effectivement optimal pour l'émetteur et le canal décrits. Nous montrerons aussi que la probabilité d'erreur pour chaque bit codé est égale à* $Q\left(\sqrt{\frac{2A^2}{N_0}}\right)$*, où* $Q(x) = \int_{t=-\infty}^{x} \frac{1}{\sqrt{2\pi}} e^{-\frac{t^2}{2}} \mathrm{d}t$*. L'énergie consommée pour envoyer un bit codé est* $A^2$ *mJ. Nous voyons donc qu'il existe un compromis entre l'énergie consommée et la probabilité obtenue : plus* $A$ *est grand plus la probabilité est grande mais plus on consomme d'énergie. Le débit de transmission est* $1/T_s = 1000$ *bit/sec et la largeur de notre signal* $x(t)$ *infinie car* $x(t)$ *est non-continue. C'est pourquoi le système décrit, même si très simple et avec d'excellentes caracteristiques en terme d'énergie, débit et probabilité d'erreur, n'est jamais utilisé sous cette forme précise en pratique. En fait, on utilisera des systèmes où chaque symbole codé* $c_\ell$ *influence le signal émis* $x(t)$ *sur une durée plus longue (afin de réduire la largeur de bande) et inversement à tout instant le signal* $x(t)$ *est influencé par plusieurs bits codés.*

## 2.4 Description de l'émetteur

La tâche de l'émetteur est de transformer les bits codés $c_1, \ldots, c_n$ au signal envoyé $x(t)$ à temps-continu. Nous allons utiliser une forme bien spécifique pour cette transformation qui nous permettra facilement de déterminer le débit de la communication ainsi que le spectre occupé et l'énergie consommée par le signal $x(t)$.

### 2.4.1 Structure du signal émis

La structure générale de notre signal émis est

$$x(t) = \sum_{\ell=0}^{N-1} s_\ell g(t - \ell T_s) \tag{2.2}$$

avec

- $\{s_\ell\}_\ell$ la suite de **symboles** qui s'expriment en fonction des bits codés $\{c_i\}_i$.
- $g(t)$ le **filtre d'émission**.
- $T_s$ le **temps-symbole**.

Il nous reste donc à définir comment passer des bits codés $\{c_i\}_i$ aux symboles $\{s_\ell\}_\ell$, comment choisir le filtre d'émission $g(t)$ et le temps-symbole $T_s$ et comment ces choix vont influencer le débit de transmission, la largeur de bande, l'énergie consommée et la probabilité d'erreur du système. Cette dernière est étroitement liée au choix du récepteur et nous allons l'étudier après avoir présenté le détecteur optimal dans la section 2.5.3.

Notons que dans l'exemple, on avait choisi :

— les symboles $s_\ell = 2A(1/2 - c_\ell)$ ;

— le filtre d'émission $g(t) = \Pi_{T_s}(t)$ avec $\Pi_T(t)$ une fonction de porte qui vaut 1 entre 0 et $T$ ms et 0 ailleurs ;

— le temps-symbole $T_s = 1$ms.

Ceci nous avait amené à un débit de 1000 bits/sec, une largeur de bande infinie, une énergie de $A^2$ mJ par bit codé, et une probabilité d'erreur par bit codé égale à $Q\left(\sqrt{\frac{2A^2}{N_0}}\right)$.

### 2.4.2 Des bits aux symboles

Le passage des bits aux symboles est spécifié par un ensemble $\mathcal{M} \subseteq \mathbb{R}$ qui contient toutes les valeurs possibles pour un symbole $s_\ell$. L'emplacement de ces valeurs dans $\mathbb{R}$ défini une **constellation** et $M := |\mathcal{M}|$ indique la taille de la constellation. La constellation (à valeurs réelles) la plus classique dispose les symboles de manière régulière sur l'axe réel. Elle est $M$-PAM (pour *Pulse Amplitude Modulation* ou « Modulation D'Amplitude » (MDA-$M$) en français.

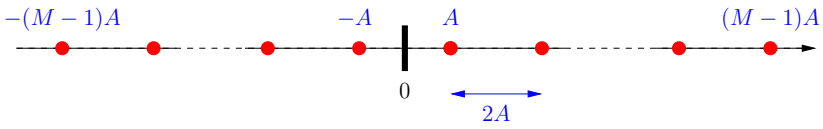



FIGURE 2.2 – Constellation $M$-PAM

Une fois la constellation fixe, il reste à determiner la correspondance des bits codés $\{c_i\}_i$ ou des bits d'information $\{d_i\}_i$ dans le cas non-codé aux symboles $\{s_\ell\}$. Par simplicité, dans la suite nous supposons que la modulation est faite pour des bits d'information. Nous allons expliquer à la fin du chapitre comment les résultats changent pour des bits codés.

Typiquement la modulation est faite *par bloc*. Par exemple, si $\log_2 M$ est un nombre entier, alors des couples de $\log_2 M$ bits sont associés à des symboles différents. Par exemple, pour une 4-PAM avec constellation $\{-3A, -A, A, 3A\}$, 2 bits sont attribués à chaque symbole de la constellation. Cette opération de correspondance entre le couple de bits et les symboles s'appelle l'**étiquetage** (*labelling* ou *mapping*, en anglais). Un exemple pour un tel étiquetage pour la 4-PAM est donné dans le Tableau 2.1. Trouver l'étiquetage

| couple de bits | | symbole associé |
|---|---|---|
| (00) | $\leftrightarrow$ | $-3A$ |
| (01) | $\leftrightarrow$ | $-A$ |
| (11) | $\leftrightarrow$ | $A$ |
| (10) | $\leftrightarrow$ | $3A$ |

TABLE 2.1 – Etiquetage de la 4-PAM

optimal au sens d'une certaine métrique (typiquement la probabilité d'erreur bit) est un problème intéressant et résolu dans de nombreux cas mais ne relève pas du cours de COM105, faute de temps.

Dans un tel étiquetage par bloc chaque symbole représente alors $\log_2(M)$ bits. On définit le *temps-bit* $T_b$ comme le temps-symbole $T_s$ divisé par le nombre de bits associés à chaque symbole :

$$T_b := \frac{T_s}{\log_2(M)}, \tag{2.3}$$

ce qui nous permet de trouver une expression simple pour le *débit binaire* $D_b$ qui représente le nombre moyen de bits envoyés par seconde :

$$D_b := \frac{1}{T_b} = \frac{\log_2(M)}{T_s}. \tag{2.4}$$

Donc, sans modifier la valeur de $T_s$, plus $M$ est grand et plus les débits sont élevés. Nous verrons à la section 2.6 que nous ne pouvons pas jouer totalement à notre guise sur ce paramètre $M$ sans compromettre gravement la qualité de la transmission.

Tout au long du cours, nous allons considérer comme valide l'hypothèse suivante sur les symboles.

**Hypothèse 2.1** *Soit $\{s_\ell\}_{\ell=0,\cdots,N-1}$ la suite des symboles à émettre dont les valeurs sont prises dans $\mathcal{M} = \{a^{(j)}\}_{j=0,\cdots,M-1}$. Les éléments de l'ensemble $\mathcal{M}$ sont de plus rangés dans l'ordre croissant.*

- *la suite aléatoire $\{s_\ell\}_{\ell=0,\cdots,N-1}$ est i.i.d.*

- *chaque symbole $s_\ell$ prend toutes les valeurs de $\mathcal{M}$ avec la même probabilité* $\Pr(s_\ell = a^{(j)}) = 1/M$, $\forall j, \ell$.
- *la constellation est de moyenne nulle.*

Selon ces hypothèses, la moyenne et l'énergie d'un symbole $s_\ell$ ne dépendent pas de $\ell$ et valent respectivement :

$$m_s := \mathbb{E}[s_\ell] = \frac{1}{M}\sum_{j=0}^{M-1} a^{(j)} = 0 \tag{2.5}$$

$$E_s := \mathbb{E}[s_\ell^2] = \frac{1}{M}\sum_{j=0}^{M-1} a^{(j)^2}. \tag{2.6}$$

La $M$-PAM est trivialement de moyenne nulle et donc satisfait notre hypothèse. En utilisant les sommes d'entiers et les sommes d'entiers au carré, nous établissons que l'énergie pour la $M$-PAM vaut

$$E_s = \frac{A^2(M^2-1)}{3}. \tag{2.7}$$

### 2.4.3 Filtre d'émission

D'abord nous listons quelques choix de filtre $g(t)$, et ensuite nous allons découvrir qualitativement les avantages et inconvénients respectifs. On suppose un choix du temps-symbole $T_s$ donné.

- **Exemple 1 : Fonction porte.**

$$g(t) = \frac{1}{\sqrt{T_s}}\Pi_{T_s}(t) = \begin{cases} 1/\sqrt{T_s} & \text{pour } t \in [0, T_s[ \\ 0 & \text{ailleurs} \end{cases}. \tag{2.8}$$

  Dans la suite, on sera interessé par le spectre du filtre, qui en l'occurrence est :

$$|G(f)| = \sqrt{T_s}|\text{sinc}\,(\pi f T_s)\,|$$

  avec $\text{sinc}(\cdot) = \sin(\cdot)/\cdot$ le sinus cardinal. Par conséquent, la largeur de bande du filtre est infinie, et comme on le verra par la suite, la largeur de bande du signal $x(t)$ émis aussi. En pratique, si on ne garde que le lobe principal, la bande est de l'ordre de $2/T_s$. Nous verrons aussi dans la suite que cette valeur représente un certain gâchis de bande ce qui excluera d'utiliser ce type de filtre.

- **Exemple 2 : Fonction sinc.**

$$g(t) = \frac{1}{\sqrt{T_s}}\text{sinc}\left(\pi\frac{t}{T_s}\right)$$

  avec le spectre

$$|G(f)| = \sqrt{T_s}\Pi_{1/T_s}(f - 1/2T_s).$$

  Ce filtre n'est pas implémentable en pratique, même avec des troncatures bien choisies, en raison du phénomène de Gibbs. C'est d'ailleurs la raison pour laquelle construire des filtres analogiques avec des gabarits abrupts s'avèrent délicats et a conduit à de nombreuses solutions sous-optimales dont on peut citer l'exemple des filtres de Butterworth donnés en cours d'ELEC101.

- **Exemple 3 : Fonction en racine de cosinus surélevé.**

$$g(t) = \frac{4\rho}{\pi\sqrt{T_s}}\frac{\cos((1+\rho)\pi t/T_s) + (T_s/(4\rho t))\sin((1-\rho)\pi t/T_s)}{1-(4\rho t/T_s)^2}$$

  et donc

$$|G(f)| = \begin{cases} \sqrt{T_s} & \text{pour } |f| \in [0, (1-\rho)/(2T_s)] \\ \sqrt{\frac{T_s}{2}\left(1+\cos\left(\frac{\pi T_s}{\rho}(|f| - (1-\rho)/(2T_s))\right)\right)} & \text{pour } |f| \in ](1-\rho)/(2T_s), (1+\rho)/(2T_s)] \\ 0 & \text{ailleurs} \end{cases}$$

  avec $\rho$ facteur compris entre 0 et 1. Ce filtre admet la bande $B$ suivante

$$B = \frac{1+\rho}{2T_s}.$$

Ce filtre offre un excellent compromis entre son occupation spectrale et son occupation temporelle. En temps, il a une décroissance bien plus rapide que le sinus cardinal et des lobes secondaires bien plus faibles. Ce compromis temps-fréquence est donc bien meilleur que ceux offerts par les filtres présentés aux exemples précédents.
Le nom du filtre vient du spectre $|G(f)|^2$ qui se compose d'une fonction porte et de morceaux de cosinus surélevé.

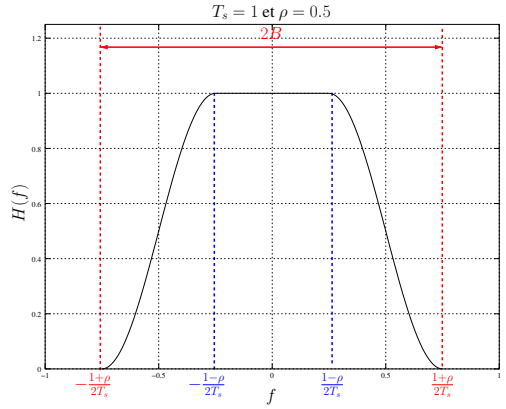



FIGURE 2.3 – spectre $|G(f)|^2$ pour le filtre en racine de cosinus surélevé $g(t)$ avec $\rho = 0.5$

### 2.4.4 Spectre du signal émis

Pour voir l'impact de $g(t)$ sur le spectre de $x(t)$ et donc sur la largeur de bande, il convient d'abord de définir le spectre et la largeur de bande du signal $x(t)$. Comme le signal $x(t)$ est aléatoire en raison de la nature aléatoire de la suite de bits, **le spectre est défini comme étant la densité spectrale de puissance (d.s.p.) du signal.**

En SI101, vous avez montré que la densité spectrale de puissance d'un signal stationnaire (au sens large) correspondait à la transformée de Fourier de la fonction d'autocorrélation. Vérifions que le signal $x(t)$ est stationnaire ou pas au sens large ? Dans la suite de cette sous-section, afin de faciliter les calculs, nous supposerons que $N = +\infty$. En conséquence, en accord avec l'hypothèse 2.1, un calcul élémentaire conduit à la fonction d'autocorrelation

$$r_{xx}(t,\tau) := \mathbb{E}[x(t+\tau)x(t)] \quad = \quad E_s \sum_{\ell=0}^{\infty} g(t+\tau-\ell T_s)g(t-\ell T_s). \tag{2.9}$$

Il est clair que $r_{xx}(t,\tau)$ dépend toujours de $t$. Par conséquent, le signal $x(t)$ n'est pas stationnaire. (En fait, il est cyclostationnaire, c'est-à-dire $r_{xx}(t,\tau) = r_{xx}(t+T_s,\tau)$ pour tout $t$ et $\tau$, comme on peut le vérifier facilement sur (2.9).)

Nous rappelons une définition de la d.s.p. d'un processus aléatoire :

$$S_{xx}(f) := \lim_{T\to+\infty} \frac{1}{T}\mathbb{E}\left[\left|\int_{t=-T/2}^{T/2} x(t)e^{-i2\pi tf}\ dt\right|^2\right]. \tag{2.10}$$

L'idée est de capturer l'énergie moyenne du processus aléatoire $x(t)$ à la fréquence $f$ pendant un certain l'intervalle.

En permutant judicieusement des limites, intégrales et espérances mathématiques on obtient le résultat suivant. Les détails du calcul sont présentés dans l'annexe A.1.

**Résultat 2.1** *Soit $f \mapsto S_{xx}(f)$ la densité spectrale du signal $x(t)$. Nous avons*

$$S_{xx}(f) = \int_{\tau=-\infty}^{\infty} r_{xx}^{(0)}(\tau)e^{-2i\pi f\tau}d\tau = \mathrm{FT}(\tau \mapsto r_{xx}^{(0)}(\tau)) \tag{2.11}$$

*avec*

$$r_{xx}^{(0)}(\tau) = \lim_{T\to\infty} \frac{1}{T}\int_{t=-T/2}^{T/2} r_{xx}(t,\tau)dt. \tag{2.12}$$

En clair, la d.s.p. est la transformée de Fourier de la moyenne de la fonction d'autocorrélation. Ceci est donc une extension de la formule vue en SI101 pour les signaux stationnaires pour lesquels $r_{xx}(t,\tau)$ ne dépend pas de $t$ et $r_{xx}^{(0)}(\tau) = r_{xx}(0,\tau)$.

Etant donné le résultat 2.1, nous sommes maintenant en mesure de déterminer le lien entre la d.s.p. de $x(t)$ et de ces paramètres-constituants qui est reporté dans le résultat 2.2. La preuve est donnée en annexe A.2.

**Résultat 2.2 (Formule de Bennett)** *Soit $x(t)$ le signal défini par l'équation (2.2) avec $N = +\infty$. On considère que les symboles vérifient l'hypothèse 2.1. On a alors*

$$S_{xx}(f) = \frac{E_s}{T_s}|G(f)|^2 \tag{2.13}$$

*avec $f \mapsto G(f)$ la transformée de Fourier de $t \mapsto g(t)$.*

Grâce au résultat 2.2, on remarque que la forme du spectre du signal émis dépend totalement de celle du filtre d'émission $g(t)$. Nous donnons, à la définition 2.1, la définition de la *largeur de bande* pour des signaux aléatoires.

**Définition 2.1** *La largeur de bande $B$ d'un signal aléatoire à valeurs réelles $t \mapsto x(t)$ avec d.s.p. $S_{xx}(f)$ est définie comme la fréquence positive maximale pour laquelle $S_{xx}(f)$ est positive. Donc si pour une fréquence positive $f_{\max}$ la d.s.p. $S_{xx}(f_{\max}) > 0$ et pour toute fréquences $f > f_{\max}$ la d.s.p. $S_{xx}(f) = 0$, alors*

$$B = f_{\max}. \tag{2.14}$$

Etant donné la définition 2.1, nous avons **la largeur de bande de $x(t)$ identique à celle de $g(t)$.**

### 2.4.5 Efficacité spectrale

La notion d'**efficacité spectrale**, notée $\eta$, qui correspond au débit binaire divisé par la largeur de la bande du signal utilisé pour le réaliser et qui s'écrit

$$\eta = \frac{D_b}{B}$$

est un moyen essentiel d'évaluer et de comparer de manière honnête des systèmes utilisant des techniques différentes et des largeurs de bande différentes. Cette efficacité s'exprime en bits/s/Hz. Pour les systèmes actuels, l'efficacité oscille typiquement entre 1 et 10.

### 2.4.6 Consommation énergétique

La puissance $P_x$ du signal $x(t)$ est définie comme

$$P_x := \lim_{T\to\infty} \frac{1}{T} \int_{t=-T/2}^{T/2} \mathbb{E}[|x(t)|^2]\ dt. \tag{2.15}$$

et l'énergie par bit $E_b$ comme

$$E_b := P_x T_b = P_x T_s / \log_2(M). \tag{2.16}$$

Comme $r_{xx}^{(0)}(\tau)$ est la transformée de Fourier inverse de $S_{xx}(f)$, nous obtenons :

$$\int_{f=-\infty}^{\infty} S_{xx}(f)\ df = r_{xx}^{(0)}(0) = \lim_{T\to\infty} \frac{1}{T} \int_{t=-T/2}^{T/2} \mathbb{E}[|x(t)|^2]\ dt = P_x. \tag{2.17}$$

Donc en utilisant l'expression donnée au résultat 2.2, nous obtenons trivialement le résultat suivant.

**Résultat 2.3** *L'énergie consommée pour émettre un bit d'information (dans le cas non codé) s'écrit de la manière suivante*

$$E_b = \frac{E_s \|g(t)\|^2}{\log_2(M)} \tag{2.18}$$

*où $\|g(t)\|^2 := \int g(t)^2 dt$ correspond à la norme au carré du filtre d'émission. Par l'égalité de Parseval, on a aussi $\|g(t)\|^2 = \int |G(f)|^2 df$.*

Dans le reste du cours, pour simplifier, nous supposerons que $\|g(t)\|^2 = 1$ et donc que nous travaillons avec des filtres d'émission normalisés comme fait dans les trois exemples donnés en section 2.4.3.

## 2.5 Description du récepteur

Les opérations à effectuer au récepteur pour retrouver les symboles vont dépendre de la distorsion apportée par le canal de propagation. Comme expliqué auparavant, nous allons nous intéresser uniquement au canal gaussien à temps continu décrit par (2.1).

### 2.5.1 Structure du récepteur optimal

Pour l'émetteur décrit précédemment et le canal gaussien donné par (2.1), un récepteur *optimal* est composé des trois boîtiers suivants :

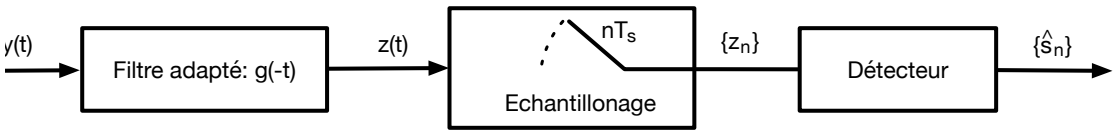



FIGURE 2.4 – Chaîne de réception optimale.

— Le premier boîtier est le *filtre adapté* au filtre d'émission, c'est-à-dire, que le signal reçu $y(t)$ passe par le filtre $g(-t)$ pour obtenir

$$z(t) = (\overleftarrow{g} \star y)(t) = \int g(-\tau)y(t-\tau)d\tau = \int y(\tau)g(\tau - t)d\tau. \tag{2.19}$$

avec $\overleftarrow{g}(t) = g(-t)$ le **filtre adapté** à $g$ et $(f_1 \star f_2)(t)$ la valeur à l'instant $t$ du produit de convolution entre deux fonctions $f_1$ et $f_2$. Pour rappel

$$(f_1 \star f_2)(t) = \int f_1(\tau)f_2(t-\tau)d\tau.$$

— Le deuxième boîtier échantillonne le signal $z(t)$ aux temps multiples de $T_s$ pour obtenir la suite :

$$z_\ell := z(\ell T_s), \qquad \ell = 0, \ldots, N. \tag{2.20}$$

— Le troisième boîtier est un détecteur optimal qui à partir de l'échantillon $z_\ell$ prend une décision sur le symbole envoyé $s_\ell$. On le décrira en détail dans la section 2.5.3.

Notons que l'échantillonneur implémente le passage d'un signal à temps continu à une suite d'échantillons (à valeurs continues). Alors que le détecteur implémente le passage d'échantillons à valeurs continues aux symboles à valeurs discrètes de la constellation utilisée.

Présenté comme cela, on pourrait imaginer appliquer directement un échantillonneur seul et se passer de filtrer le signal reçu. Un des problèmes de cette approche est que la composante du bruit dans chaque échantillon est de variance infinie (car il s'agit d'un bruit blanc). Le deuxième problème est que selon le théorème bien connu d'échantillonnage de Shannon-Nyquist on devrait avoir un débit d'échantillonnage strictement supérieur à $1/T_s$ si on utilise les filtres de mise on forme proposés dans les exemples 2 et 3 (car ils ont une largeur de bande supérieure à $1/(2T_s)$). En fait, on peut contourner le théorème d'échantillonnage de Shannon-Nyquist car le but n'est pas de reconstruire $y(t)$ mais de détecter seulement les symboles $\{s_\ell\}$ présents dans $y(t)$.

Pour prouver que l'architecture décrite au-dessus est optimale, on devrait faire appel à des techniques reposant sur la notion d'espace des signaux à caractère hilbertien. En particulier, l'optimalité des premiers deux boîtiers se base sur le fait que la combinaison du filtre adapté avec l'échantillonnage correspond à une projection du signal reçu $y(t)$ sur une base de l'espace des signaux qui est $\{g(t-\ell T_s)\}_\ell$. Par la nature du bruit blanc, ces échantillons forment ce qu'on appelle une **statistique suffisante** pour la détection des symboles. Ceci est équivalent à dire que la détection optimale des symboles peut se baser sur ces échantillons au lieu du signal complet $y(t)$.

Nous n'allons pas présenter la preuve d'optimalité du récepteur proposé dans ce cours. Par contre, nous allons prouver que le filtre adapté maximise le Rapport Signal-à-Bruit (RSB) (*Signal-to-Noise Ratio (SNR)* en anglais) des échantillons $\{z_\ell\}_\ell$. Ceci n'implique pas l'optimalité du récepteur, mais est juste une propriété intéressante à remarquer. On considère d'abord un unique symbole transmis (en fait le $\ell^{\text{ème}}$), d'où,

$$y(t) = s_\ell g(t - \ell T_s) + b(t)$$

On pose

$$\tilde{z}_\ell := (g_r \star y)(\ell T_s) = \int g_r(\tau) y(\ell T_s - \tau) d\tau, \tag{2.21}$$

ce qui donne

$$\tilde{z}_\ell = \underbrace{s_\ell \int g_r(\tau) g(-\tau) d\tau}_{\text{signal utile}} + \underbrace{\int g_r(\tau) b(\ell T_s - \tau) d\tau}_{\text{bruit}} \tag{2.22}$$

pour un filtre $g_r(t)$ quelconque, et on définit ici le SNR comme l'énergie du signal utile divisée par l'énergie du bruit :

$$\text{SNR} = \frac{2E_s}{N_0} \cdot \frac{\left(\int g_r(\tau) g(-\tau) d\tau\right)^2}{\int g_r(\tau)^2 d\tau}. \tag{2.23}$$

**Résultat 2.4 (Max-SNR et Filtre adapté)** *Parmi tous les filtres $g_r$, le filtre adapté est celui qui maximise le SNR dans* (2.23) *:*

$$\max_{g_r} \frac{\left(\int g_r(\tau) g(-\tau) d\tau\right)^2}{\int g_r(\tau)^2 d\tau} = \left.\frac{\left(\int g_r(\tau) g(-\tau) d\tau\right)^2}{\int g_r(\tau)^2 d\tau}\right|_{g_r(\tau)=g(-\tau)} = \|g(t)\|^2. \tag{2.24}$$

*Preuve :* Par l'inégalité de Cauchy-Schwartz (appliquée, ici, sur des signaux à valeurs réelles ce qui explique l'absence de complexe conjugué), nous avons une borne sur le SNR indépendante de $g_r(t)$ qui vaut

$$\text{SNR} \leq \frac{2E_s}{N_0} \|g(t)\|^2$$

et cette borne est atteinte et donc le SNR est maximisé par rapport à $g_r(t)$ si et seulement si $g_r(t) = g(-t)$ ce qui conclut la preuve.

Pour la petite histoire, le filtre adapté n'a pas été créé pour les communications numériques mais pour le traitement du signal en radar. Il permet en effet de faire ressortir l'écho émis par le radar (noté $g(t)$) caché dans le bruit ambiant.

### 2.5.2 De l'interférence entre symboles : le filtre de Nyquist

Dans ce qui suit, on suppose un filtre d'émission $g(t)$ et l'utilisation du filtre adapté comme décrit dans la section précédente. On définit alors le filtre global réunissant le filtre d'émission et le filtre de réception :

$$h(t) := (\overleftarrow{g} \star g)(t) \tag{2.25}$$

Avant d'aller plus loin, écrivons la suite $\{z_\ell\}_\ell$ en fonction des symboles $\{s_\ell\}_\ell$. Le lien est donné par le résultat 2.5 suivant.

**Résultat 2.5 (Interférence entre symboles)** *Dans le contexte d'un canal gaussien à temps continu et dans le cas de l'application du filtre adapté au récepteur, nous avons*

$$z_\ell = h_0 s_\ell + \underbrace{\sum_{j \neq 0} h_j s_{\ell - j}}_{\textit{Interférence entre symboles}} + w_\ell \tag{2.26}$$

*avec*

- $h_\ell = h(\ell T_s)$ *les échantillons du filtre global, et*
- $w_\ell = w(\ell T_s)$ *avec* $w(t) = \int g(-\tau) b(t - \tau) d\tau$ *le bruit passé à travers le filtre adapté.*

*Notons que la suite $\{w_\ell\}_\ell$ est gaussienne, avec un spectre $S_{ww}(e^{2i\pi f})$ qui vaut*

$$S_{ww}(e^{2i\pi f}) = \frac{N_0}{2} h(e^{2i\pi f}) \tag{2.27}$$

*où $h(e^{2i\pi f}) = \sum_\ell h_\ell e^{-2i\pi f \ell}$ est la Transformée de Fourier à temps discret de la suite $\{h_\ell\}_\ell$.*

*Preuve :* En convoluant le signal reçu $y(t) = \sum_j s_j g(t - jT_s)$ avec le filtre $\overleftarrow{g}$, nous obtenons que

$$z(t) = \sum_j s_j h(t - jT_s) + w(t).$$

En échantillonnant ensuite à la cadence des symboles, nous avons

$$z_\ell = \sum_j s_j h_{\ell - j} + w_\ell.$$

Comme le produit de convolution est commutatif, nous avons bien l'équation (2.26).

La corrélation du bruit $r_{ww}(\ell) = \mathbb{E}[w_{j+\ell} w_j]$ s'exprime en permutant l'espérance mathématique et les intégrales temporelles comme ci-dessous

$$r_{ww}(\ell) = \iint g(-\tau) g(-\tau') \mathbb{E}[b(jT_s + \ell T_s - \tau) b(jT_s - \tau')] d\tau d\tau'.$$

Comme le bruit $b(t)$ est blanc,

$$\begin{aligned} r_{ww}(\ell) &= \frac{N_0}{2} \iint g(-\tau) g(-\tau') \delta(\ell T_s - \tau + \tau') d\tau d\tau' \\ &= \frac{N_0}{2} \int g(-\tau) g(\ell T_s - \tau) d\tau \\ &= \frac{N_0}{2} (\overleftarrow{g} \star g)(\ell T_s) \\ &= \frac{N_0}{2} h_\ell. \end{aligned}$$

Comme le spectre pour les suites aléatoires stationnaires est défini par

$$S_{ww}(e^{2i\pi f}) = \sum_\ell r_{ww}(\ell) e^{-2i\pi f \ell},$$

le résultat est démontré. ■

La présence d'Interférence Entre Symboles (ISI) dans l'équation (2.26) fait qu'un même symbole $s_\ell$ est présent sur plusieurs échantillons $z_{\ell'}$. Le détecteur ayant pour objectif de recouvrir optimalement $s_\ell$ ne peut donc se contenter de travailler échantillon par échantillon mais doit travailler conjointement sur un ensemble d'échantillons dont la taille va certainement dépendre de la longueur du filtre $\{h_\ell\}_\ell$. Cet algorithme existe et s'appelle l'algorithme de Viterbi. Mais dès que le filtre $\{h_\ell\}_\ell$ est trop long ou que la taille de la constellation $M$ est trop grande, cet algorithme devient inopérant en raison de sa complexité algorithmique.

En présence d'un canal gaussien à temps continu, nous allons voir maintenant qu'en choisissant judicieusement le filtre d'émission, on peut éliminer l'ISI et donc développer en sous-section 2.5.3 un détecteur bien plus simple qui ne fera pas appel aux techniques mentionnées au paragraphe précédent.

Lorsque le canal est gaussien à temps continu, l'origine de l'ISI est unique et provient seulement de la présence du filtre d'émission et de son filtre adapté. L'ISI étant vue comme un facteur nuisible (cela complique les récepteurs et en fait augmente la probabilité d'erreur), **il nous paraît opportun de sélectionner un filtre d'émission ne créant pas d'ISI au niveau du signal $z_\ell$.** C'est pourquoi l'objectif de cette sous-section est d'exhiber les conditions sur $g(t)$ et de manière équivalente sur $h(t)$ (qui, nous le rappelons, vaut $(\overleftarrow{g} \star g)(t)$) pour empêcher la présence d'ISI au niveau de $z_\ell$.

En examinant l'équation (2.26), il est évident qu'il n'y aura pas de mélange des symboles au niveau de $z_\ell$ si et seulement si les cœfficients du mélange sont nuls sauf un. Autrement dit, il n'y a pas d'ISI si et seulement si il existe un unique $\ell_0$ pour lequel $h_{\ell_0} \neq 0$ et $h_\ell = 0$, $\forall \ell \neq \ell_0$. Ceci est bien une condition nécessaire et suffisante. En effet, le sens « suffisant » est trivial et le sens « nécessaire » provient du fait que nous voulons l'absence d'ISI quelle que soit la valeur des symboles émis et non pour une combinaison particulière de symboles. Dans la suite du cours, sans perte de généralité, nous prendrons toujours $\ell_0 = 0$ pour simplifier. Cette condition sur le filtre $h(t)$ se traduit par l'introduction du filtre de Nyquist donné en définition 2.2.

**Définition 2.2 (Filtre de Nyquist)** *Un filtre de réponse impulsionnelle $p(t)$ est dit de Nyquist (pour la cadence $1/T_s$) si et seulement si*

$$p_\ell = p(\ell T_s) = \begin{cases} \neq 0 & \text{pour } \ell = 0 \\ 0 & \text{pour } \ell \neq 0 \end{cases} . \tag{2.28}$$

*Par simple application de la formule sommatoire de Poisson ($\sum_\ell P(f - \ell/T_s) = T_s \sum_\ell p_\ell e^{-2i\pi f \ell T_s}$), nous obtenons une équation dans le domaine fréquentiel associée à* (2.28).

*Ainsi, de manière équivalente, un filtre de fonction de transfert $P(f)$, qui est la transformée de Fourier de $p(t)$, est dit de Nyquist si et seulement si*

$$\sum_\ell P\left(f - \frac{\ell}{T_s}\right) = T_s p_0 = \text{constante}, \quad \forall\ f. \tag{2.29}$$

Ainsi le premier résultat de cette sous-section est le suivant.

**Résultat 2.6** *Soit $h(t)$ le filtre englobant le filtre d'émission et le filtre de réception et intervenant dans l'équation* (2.26)*. Pour ne pas créer d'ISI, ce filtre doit être de Nyquist (pour la cadence $1/T_s$).*

En pratique, on choisit le filtre d'émission $g(t)$ qui conduit à un filtre global $h(t) = (\overleftarrow{g} \star g)(t)$ ayant la propriété de Nyquist. On dira alors que $g(t)$ est un filtre de racine en Nyquist, comme défini dans la suite.

**Définition 2.3 (Filtre en racine de Nyquist)** *Un filtre $p(t)$ (à valeurs réelles) est dit en racine de Nyquist si et seulement si le filtre $(\overleftarrow{p} \star p)(t)$ (avec $\overleftarrow{p}(t) = p(-t)$) est un filtre de Nyquist. Autrement dit, le filtre convolué à son filtre adapté est de Nyquist.*

L'expression « racine de Nyquist » provient du fait que le spectre de $(\overleftarrow{p} \star p)(t)$ est égal à $|P(f)|^2$. Donc si $p(t)$ est racine de Nyquist, $(\overleftarrow{p} \star p)(t)$ est Nyquist et $|P(f)|^2$ vérifie la condition de Nyquist fréquentielle. Autrement dit, de tout filtre de Nyquist, on peut créer un filtre en racine de Nyquist dont le spectre sera la racine du module du spectre du filtre de Nyquist initial.

**Résultat 2.7** *Si $h(t)$ est un filtre de Nyquist en accord avec le résultat 2.6 et si $h(t) = (\overleftarrow{g} \star g)(t)$ en accord avec le résultat 2.5, alors $g(t)$ est un filtre en racine de Nyquist.*

Le résultat 2.8 fournit une condition nécessaire (mais pas suffisante) que la bande des filtres de Nyquist ou en racine de Nyquist doit satisfaire. Cette condition n'est pas sans rappeler un commentaire effectué dans la dernière phrase du paragraphe consacré au spectre du signal émis dans la sous-section 2.4.3.

**Résultat 2.8** *La largeur de bande, notée $B$, de tout filtre de Nyquist ou en racine de Nyquist (pour la cadence $1/T_s$) vérifie*

$$B \geq \frac{1}{2T_s}.$$

*Preuve :* Examinons d'abord le cas du filtre de Nyquist. Raisonnons par l'absurde : supposons que $B < 1/(2T_s)$, alors la condition fréquentielle donnée par (2.29) ne peut pas être satisfait puisque le terme sera parfois nul et parfois non nul. En effet, il sera nul quand les fonctions décalées ne se superposent pas ce qui sera le cas puisque $B < 1/(2T_s)$ et comme montré sur la figure 2.5. Donc il faut nécessairement que $B \geq 1/(2T_s)$. Comme le spectre du filtre en racine de Nyquist est la racine du spectre du filtre de Nyquist associé, ils ont même largeur de bande et donc la condition est identique. ■

Evidemment, dans le contexte du canal gaussien à temps continu, cette condition est essentielle car elle nous permet de construire des filtres d'émission ne créant pas d'ISI à la réception. Donnons quelques exemples de filtres en racine de Nyquist et de filtres de Nyquist. Pour cela, revenons sur les filtres d'émission introduits auparavant.

- **Exemple 1 : Fonction porte.** En observant (2.8), on obtient aisément que cette fonction représente un filtre de Nyquist. C'est normal car, par construction, les symboles ne sont pas mélangés : en effet, à chaque instant, et donc a fortiori aux instants $\ell T_s$, un seul symbole est présent. La convolution de la fonction porte avec son filtre adapté conduit à une fonction en triangle entre $-T_s$ et $T_s$. Cette fonction triangle est donc aussi un filtre de Nyquist. Et la fonction porte est ainsi aussi un filtre en racine de Nyquist.

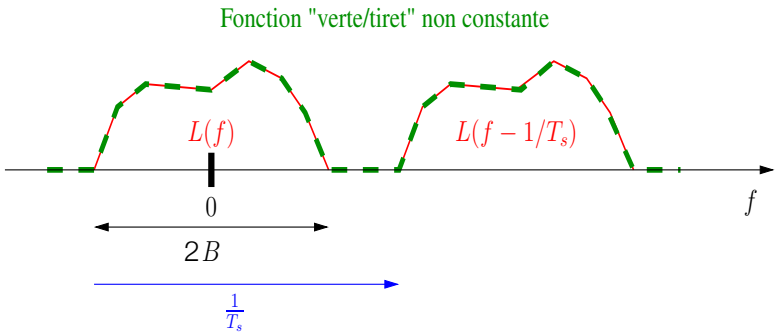



FIGURE 2.5 – Allure du terme de gauche de l'équation (2.29) quand $B < 1/(2T_s)$

- **Exemple 2 : Fonction sinc.** Il est facile de montrer, par l'intermédiaire de la version fréquentielle du critère de Nyquist, que cette fonction de nouveau représente un filtre de Nyquist et un filtre en racine de Nyquist.
- **Exemple 3 : Fonction en racine de cosinus surélevé.** Comme mentionné en sous-section 2.4.3, ce filtre est couramment employé dans les systèmes pratiques en raison de son bon compromis temps-fréquence et aussi parce qu'il est en racine de Nyquist (mais pas de Nyquist). En fait le filtre global résultant de la convolution de ce filtre en racine de cosinus surélevé avec son filtre adapté, qui **se nomme filtre en cosinus surélevé**, admet la réponse impulsionnelle et la fonction de transfert suivantes

$$h(t) = \mathrm{sinc}(\pi t/T_s)\frac{\cos(\pi\rho t/T_s)}{1-(2\rho t/T_s)^2}$$

et

$$H(f) = \begin{cases} T_s & \text{pour } |f| \in [0,(1-\rho)/(2T_s)] \\ T_s/2\left(1+\cos\left(\frac{\pi T_s}{\rho}(|f|-(1-\rho)/(2T_s))\right)\right) & \text{pour } |f| \in ](1-\rho)/(2T_s),(1+\rho)/(2T_s)] \\ 0 & \text{ailleurs} \end{cases}$$

Le terme « cosinus surélevé » provient de l'arc de cosinus translaté verticalement et permettant de

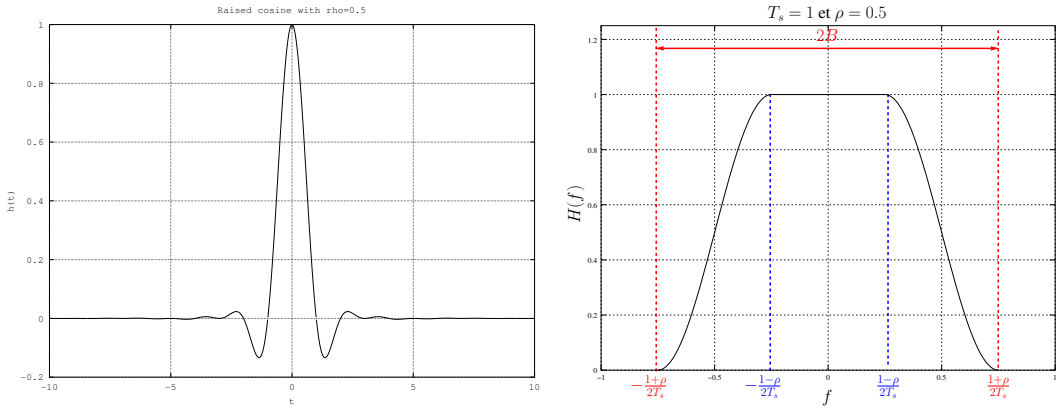



FIGURE 2.6 – $h(t)$ (à gauche) et $H(f)$ (à droite) pour le filtre en cosinus surélevé ($T_s = 1$ et $\rho = 0.5$)

rejoindre les deux parties plates du spectre comme montré sur la figure 2.6. De nouveau la bande vaut

$$B = \frac{1+\rho}{2T_s}$$

ce qui montre que la condition $B \geq 1/(2T_s)$ est respectée. Le facteur d'excès de bande $\rho$ (*roll-off*, en anglais) détermine la bande supplémentaire que l'on s'impose par rapport à la bande minimale requise $1/2T_s$.

Une fois la contrainte de filtre de Nyquist vérifiée par $h(t)$, (2.26) se simplifie de la manière suivante

$$z_\ell = h_0 s_\ell + w_\ell$$

avec, pour rappel, $h_0 = \int g(t)^2 dt$ et donc $h_0 = \|g(t)\|^2$. De plus le bruit $w_j$ devient blanc de variance $h_0 N_0/2$ en accord avec (2.27).

Or, par hypothèse, nous avons fixé $\|g(t)\|^2 = 1$ ce qui implique que $h_0 = 1$. Par conséquent, nous obtenons le modèle suivant pour le **canal gaussien à temps discret**

$$z_\ell = s_\ell + w_\ell \tag{2.30}$$

avec $w_\ell$ un bruit blanc gaussien centré de variance $N_0/2$. L'équation (2.30) justifiera le modèle de canal gaussien (à temps discret) utilisé dans le chapitre 3.

Dans l'exemple-jouet de la sous-section 2.3, nous avions $\|g(t)\|^2 = T_s$ et non $\|g(t)\|^2 = 1$. Pour retrouver le modèle de (2.30), il suffira de normaliser en divisant le signal échantillonné par $T_s$.

### 2.5.3 Détecteur optimal

Dans cette sous-section, nous allons décrire le détecteur optimal du canal gaussien à temps discret, c'est-à-dire, quelles opérations appliquées pour retrouver optimalement les symboles $s_\ell$ à partir de $z_\ell$ comme donné dans (2.30).

Pour cela, nous allons utiliser les résultats généraux sur la théorie de la détection exposés et démontrés à la sous-section 1.5.2 du chapitre 1. Nous rappelons juste que le détecteur qui minimise la probabilité d'erreur suit la règle du maximum de vraisemblance (*Maximum Likelihood (ML)*, en anglais) si les symboles émis sont équiprobables.

Selon l'hypothèse 2.1, les symboles sont indépendants entre eux. De plus comme le bruit est blanc et gaussien, les échantillons $w_\ell$ sont indépendants entre eux. Par conséquent l'échantillon $z_\ell$ ne possède que de l'information sur l'instant $\ell T_s$ et donc sur $s_\ell$. Ainsi nous pouvons travailler, sans perte d'optimalité, symbole par symbole. De plus, toujours selon l'hypothèse 2.1, les symboles sont équiprobables. C'est pourquoi, nous allons appliquer le **détecteur ML sur le signal $z_\ell$ pour retrouver $s_\ell$.** Nous obtenons alors le résultat suivant.

**Résultat 2.9** *Soit $\{s_\ell\}_\ell$ une suite de symbole vérifiant l'hypothèse 2.1 et soit le canal gaussien à temps discret vérifiant* (2.30)*. Alors le détecteur optimal obtient le symbole $\hat{s}_\ell$ de la manière suivante*

$$\hat{s}_\ell = \begin{cases} a^{(0)} & si\ z_\ell \in\ ]-\infty, t^{(0)}] \\ a^{(j)} & si\ z_\ell \in\ ]t^{(j-1)}, t^{(j)}]\ pour\ j \in \{1, \cdots, M-2\} \\ a^{(M-1)} & si\ z_\ell \in\ ]t^{(M-2)}, +\infty[ \end{cases}$$

*avec, pour $j \in \{0, \cdots, M-2\}$, les seuils (*threshold*, en anglais) suivants*

$$t^{(j)} = \frac{a^{(j)} + a^{(j+1)}}{2}.$$

*Preuve :* comme le détecteur ML est optimal, $\hat{s}_\ell$ est défini de la manière suivante

$$\hat{s}_\ell = \arg \max_{s_\ell \in \{a^{(0)}, \cdots, a^{(M-1)}\}} p(z_\ell | s_\ell) \tag{2.31}$$

avec $p(z_\ell|s_\ell)$ la densité de probabilité de $z_\ell$ en supposant $s_\ell$ fixé. Ecrivons donc $p(z_\ell|s_\ell)$. Conditionné à $s_\ell$, l'échantillon $z_\ell$ est gaussien de moyenne $s_\ell$ et de variance $N_0/2$. Par conséquent,

$$p(z_\ell|s_\ell) = \frac{1}{\sqrt{\pi N_0}} e^{-\frac{(z_\ell - s_\ell)^2}{N_0}}. \tag{2.32}$$

A partir de (2.31) et (2.32), nous avons

$$\hat{s}_\ell = \arg \min_{s_\ell \in \{a^{(0)}, \cdots, a^{(M-1)}\}} (z_\ell - s_\ell)^2. \tag{2.33}$$

Par conséquent, $z_\ell$ sera affecté au symbole $a^{(j)}$ le plus proche au sens de la norme quadratique. On peut ainsi définir une région de décision associée au symbole $a^{(j)}$ de la manière suivante

$$\begin{aligned}\Omega^{(j)} &\triangleq \{z \in \mathbb{R} \text{ t.q. } \hat{s} = a^{(j)}\} \\ &= \{z \in \mathbb{R} \text{ t.q. } a^{(j)} \text{ est le plus proche de } z \text{ au sens de la distance quadratique}\}.\end{aligned}$$

Ainsi, la région de décision $\Omega^{(j)}$ est un intervalle de l'axe des réels dont les bornes sont les points à égale distance entre $a^{(j-1)}$ et $a^{(j)}$ et entre $a^{(j)}$ et $a^{(j+1)}$. Evidemment un traitement particulier est à appliquer pour les deux symboles des extrémités ce qui conclut la preuve. ■

Pour la constellation introduite dans le polycopié, à savoir la PAM, il est facile de vérifier que

$$t^{(j)}_{\text{PAM}} = (2j + 2 - M)\, A$$

ce qui donne la figure 2.7 suivante pour la 4-PAM. Ce type de détecteur par région de décisions dont les bornes sont les médiatrices des points des constellations s'appelle le **détecteur à seuil**.

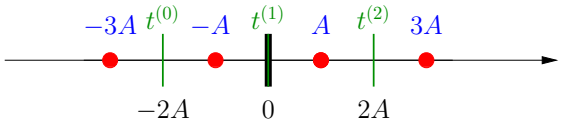



FIGURE 2.7 – Régions de décision pour la 4-PAM

## 2.6 Performances

L'objectif principal de cette section est de déterminer la probabilité d'erreur sur les bits émis. Ceci nous permettra de mesurer l'influence de certaines variables ajustables à notre guise, comme la constellation (cf. la sous-section 2.6.1) et le code correcteur d'erreur (cf. la sous-section 2.6.3).

### 2.6.1 Probabilité d'erreur pour la $M$-PAM (non codée)

Souvent on s'intéresse à la probabilité d'erreur par bit d'information. Pour trouver celle-ci on passe d'habitude par la *probabilité d'erreur par symbole* qui est définie comme suit :

$$P_e := \Pr(\hat{s}_\ell \neq s_\ell). \tag{2.34}$$

On se rappelle que $s_\ell$ est une variable aléatoire qui est uniformément distribuée sur la constellation $\mathcal{M}$ et $\hat{s}_\ell$ est une fonction du signal aléatoire reçu $y(t)$, et donc $\hat{s}_\ell$ est aussi une variable aléatoire. Pour calculer la probabilité d'erreur $P_e$ il faut donc considérer l'aléa introduit par le choix aléatoire du symbole $s_\ell$ et l'aléa introduit par le bruit du canal qui affecte le signal reçu $y(t)$. Souvent il est plus simple de considérer les deux sources d'aléa (le choix du symbole et le bruit du canal) l'une après l'autre en introduisant un conditionnement sur la valeur du symbole choisit $s_\ell$. On calcule alors $P_e$ de la façon suivante :

$$P_e = \sum_{j=0}^{M-1} \Pr(\hat{s}_\ell \neq s_\ell | s_\ell = a^{(j)}) \cdot \Pr(s_\ell = a^{(j)}) = \frac{1}{M} \sum_{j=0}^{M-1} \Pr(\hat{s}_\ell \neq a^{(j)} | s_\ell = a^{(j)}). \tag{2.35}$$

On note que la probabilité $\Pr(\hat{s}_\ell \neq a^{(j)} | s_\ell = a^{(j)})$ porte seulement sur l'aléa du bruit du canal mais pas sur le choix du symbole $s_\ell$ qui est fixé à $a^{(j)}$.

Aux résultats 2.10 et 2.11, nous obtenons les probabilités d'erreur symbole $P_e$ pour la 2-PAM et la $M$-PAM respectivement.

**Résultat 2.10** *Si l'hypothèse 2.1 est vérifiée, la constellation 2-PAM admet les performances suivantes*

$$P_e = Q\left(\sqrt{2\frac{E_b}{N_0}}\right) \tag{2.36}$$

*avec la fonction*

$$x \mapsto Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^{+\infty} e^{-u^2/2} du \tag{2.37}$$

*correspondant à la queue de la loi gaussienne centrée et de variance unité (*Gaussian tail function*, en anglais).*

*Preuve :* Comme $\mathcal{M} = \{-A, A\}$, on obtient que

$$\begin{aligned}
P_e &= \frac{1}{2}\Pr(\hat{s}_\ell = -A | s_\ell = A) + \frac{1}{2}\Pr(\hat{s}_\ell = A | s_\ell = -A) \\
&= \frac{1}{2}\Pr(z_\ell < 0 | s_\ell = A) + \frac{1}{2}\Pr(z_\ell \geq 0 | s_\ell = -A) \text{ (car seuil à zéro pour le détecteur optimal)} \\
&= \frac{1}{2}\Pr(A + w_\ell < 0) + \frac{1}{2}\Pr(-A + w_\ell \geq 0) \text{ (car } z_\ell = s_\ell + w_\ell) \\
&= \frac{1}{2}\Pr(w_\ell < -A) + \frac{1}{2}\Pr(w_\ell \geq A) \\
&= \Pr(w_\ell \geq A) \text{ (par symétrie autour de zéro de la loi gaussienne)}
\end{aligned}$$

Il reste à calculer la probabilité qu'une variable gaussienne centrée et de variance $N_0/2$ soit supérieure à un seuil $A$. Ceci peut se faire par l'intermédiaire de la fonction de répartition (*Cumulative Density Function (cdf)*, en anglais) et correspond à l'intégration entre $A$ et $+\infty$ de la densité de probabilité (*Probability Density Function (pdf)*, en anglais). Ainsi

$$\begin{aligned}
P_e &= \frac{1}{\sqrt{\pi N_0}} \int_A^{+\infty} e^{-u^2/N_0} du \\
&= Q\left(\frac{\sqrt{2}A}{\sqrt{N_0}}\right) \text{ (par changement de variable } u' = u\sqrt{2/N_0})
\end{aligned}$$

Il convient maintenant d'exprimer $A$ en fonction de $E_b$. Comme $E_s = A^2$ et $M = 2$, (2.18) conduit à $E_b = A^2$ ce qui prouve le résultat. ■

**Résultat 2.11** *Si l'hypothèse 2.1 est vérifiée, la constellation $M$-PAM admet les performances suivantes*

$$P_e = 2\left(1 - \frac{1}{M}\right) Q\left(\sqrt{\frac{6\log_2(M)}{M^2 - 1}\frac{E_b}{N_0}}\right). \tag{2.38}$$

*Preuve :* Les probabilités d'erreur individuelles $\Pr(\hat{s}_\ell \neq a^{(j)} | s_\ell = a^{(j)})$ se décomposent en deux catégories : les symboles aux deux bords de la constellation $M$-PAM ($a^{(0)}$ et $a^{(M-1)}$) ont par symétrie la même probabilité d'erreur notée $P_e^{externe}$ qui en revanche est différente des autres symboles de la constellation. Les autres symboles ont eux aussi par symétrie une probabilité d'erreur commune, notée $P_e^{interne}$. On a simplement que

$$P_e = \frac{1}{M}(M-2)P_e^{interne} + \frac{2}{M}P_e^{externe}. \tag{2.39}$$

Pour les symboles aux extrémités de la constellation, on a

$$P_e^{externe} = \Pr\{w_\ell > A\} = \int_A^{\infty} \frac{1}{\sqrt{\pi N_0}} \exp\left(-\frac{t^2}{N_0}\right) dt = Q\left(\sqrt{\frac{2A^2}{N_0}}\right). \tag{2.40}$$

Pour les autres symboles de la constellation, il y a erreur si $z_\ell$ passe à gauche ou à droite de la médiatrice séparant le symbole transmis avec ses symboles adjacents. Ceci est possible si le bruit est inférieur à $-A$ ou supérieur à $A$. Ainsi on a

$$P_e^{interne} = \Pr\{w_\ell > A \text{ ou } w_\ell < -A\} = 2\Pr\{w_\ell > A\} = 2Q\left(\sqrt{\frac{2A^2}{N_0}}\right). \tag{2.41}$$

En réunissant (2.39), (2.40) et (2.41), on obtient

$$P_e = 2\left(1 - \frac{1}{M}\right) Q\left(\sqrt{\frac{2A^2}{N_0}}\right).$$

Il suffit juste maintenant de relier $A$ et $E_b$. Pour cela, on calcule d'abord $E_s$. Etant donné (2.6), on a

$$E_s = \frac{A^2}{M}\sum_{j=1}^{M}(2j - M - 1)^2 = \frac{A^2}{3}(M^2 - 1) \tag{2.42}$$

L'énergie par bit reçu vaut naturellement :

$$E_b = \frac{E_s}{\log_2(M)} = \frac{A^2(M^2-1)}{3\log_2(M)}$$

ce qui permet de conclure la preuve. ■

Les deux résultats précédents sont souvent présentés dans la littérature par le biais de la fonction erfc(·). Par simple changement de variable, on sait que $Q(x) = (1/2)\text{erfc}(x/\sqrt{2})$ permettant ainsi de passer d'un formalisme à l'autre aisément.

La fonction $Q(\cdot)$ n'admet pas d'expression analytique autre que celle donnée en (2.37). En revanche, de nombreuses approximations relativement précises existent. Dans la suite du cours, nous utiliserons le fait que

$$Q(x) \approx \frac{1}{2}e^{-x^2/2}. \tag{2.43}$$

Par conséquent, étant donné (2.38), la probabilité d'erreur admet une **décroissance exponentielle** en fonction du SNR et donc une décroissance extrêmement rapide. C'est d'ailleurs la raison pour laquelle les figures affichant la probabilité d'erreur, notamment la figure 2.8, utilisent une échelle logarithmique.

### 2.6.2 Probabilité d'erreur par bit

Il nous faut un lien entre la probabilité d'erreur au niveau bit (qui est notre but ultime) et la probabilité d'erreur $P_e$ au niveau symbole (qui est plus facilement calculable analytiquement). Pour cela, revenons à la notion d'étiquetage (cf. la sous-section 2.4.2). On rappelle qu'à chaque symbole $s_\ell$ on peut associer $m = \log_2(M)$ bits $d_1, \ldots, d_m$. De même, à chaque symbole décodé $\hat{s}_\ell$ on peut associer les $m$ bits $\hat{d}_1, \ldots, \hat{d}_m$. La *probabilité d'erreur par bit* $P_b$ est censée mesurer la probabilité qu'un bit précis est erroné et est définie de la façon suivante :

$$P_b := \frac{1}{m}\sum_{\ell=1}^{m}\Pr(\hat{d}_\ell \neq d_\ell). \tag{2.44}$$

Comme une erreur sur un symbole peut conduire à une erreur sur *plusieurs* bits, il est souvent difficile de calculer la valeur exacte de $P_b$. Sauf pour le cas trivial où $m = 1$ et donc $P_e = P_b$, on se contente d'approximer $P_b$ par une fonction de $P_e$. On supposera

- que les erreurs prédominantes ne sont produites que par des symboles adjacents, et
- que l'étiquetage permet d'avoir seulement un bit différent entre deux symboles adjacents.

Alors, seulement 1 bit sur $m = \log_2(M)$ est en erreur quand un symbole est en erreur (car en erreur avec un symbole adjacent) ce qui induit la relation suivante

$$P_b \approx \frac{1}{\log_2(M)}P_e. \tag{2.45}$$

Noter que la condition sur l'étiquetage n'est pas restrictive. En effet, on peut toujours ordonner une série de vecteurs contenant $\log_2(M)$ bits en modifiant seulement un bit à la fois. Un tel étiquetage est dit de Gray.

Pour une $M$-PAM, on obtient donc l'approximation donnée dans le résultat suivant.

**Résultat 2.12 (Probabilité d'erreur par bit pour une $M$-PAM)** *Pour une $M$-PAM :*

$$P_b \approx \frac{2}{\log_2(M)}\left(1-\frac{1}{M}\right) Q\left(\sqrt{\frac{6\log_2(M)}{M^2-1}\frac{E_b}{N_0}}\right). \tag{2.46}$$

Une fois l'expression de probabilité d'erreur obtenue, nous souhaitons nous en servir pour comparer les constellations entre elles. Grâce au résultat 2.12, nous remarquons que

$$P_b \approx \beta_{\mathcal{M}} Q\left(\sqrt{\gamma_{\mathcal{M}}\frac{E_b}{N_0}}\right) \tag{2.47}$$

avec

$$\beta_{\mathcal{M}} := \frac{2}{\log_2(M)}\left(1-\frac{1}{M}\right) \tag{2.48}$$

$$\gamma_{\mathcal{M}} := \frac{6\log_2(M)}{M^2-1} \tag{2.49}$$

deux paramètres dépendant uniquement de la constellation $\mathcal{M}$ choisie. Cette forme s'avère en fait valable pour de très nombreuses constellations non traitées en cours. Ainsi les performances de la constellation $\mathcal{M}$ sont parfaitement caractérisées par la seule connaissance de $\beta_{\mathcal{M}}$ et $\gamma_{\mathcal{M}}$. Comme $\beta_{\mathcal{M}}$ se trouve à l'extérieur de la fonction $Q(\cdot)$ et varie assez peu d'une constellation à l'autre, son impact sur les performances est en pratique négligeable. On dira donc que $P_b \approx Q(\sqrt{\gamma_{\mathcal{M}} E_b/N_0})$ ce qui implique que seul le paramètre $\gamma_{\mathcal{M}}$, appelé **gain de modulation**, différencie vraiment les performances des constellations entre elles. Pour les $M$-PAM, il nous suffit donc d'analyser

$$\gamma_{M-\text{PAM}} = \frac{6\log_2(M)}{M^2-1}. \tag{2.50}$$

Evidemment, le gain de modulation diminue en fonction de $M$ et comme $Q(\cdot)$ est une fonction décroissante, **la probabilité d'erreur augmente quand la taille de la constellation augmente** comme on le constate à la figure 2.8. En revanche, augmenter la taille de la constellation permet d'augmenter le débit (puisqu'il est proportionnel à $\log_2(M)$ en raison de (2.4)). **Un compromis est donc nécessaire entre performances et débit.** Pour comparer les courbes de la figure 2.8, on peut aussi procéder de la manière suivante : examiner

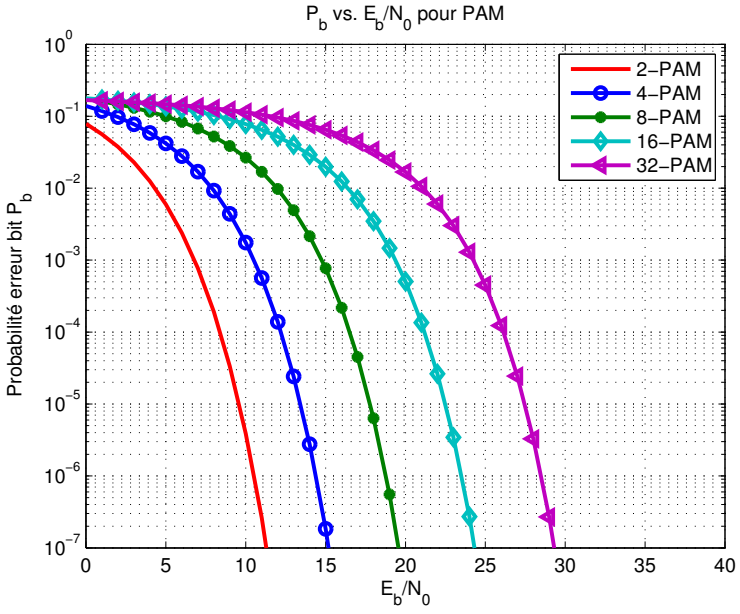



FIGURE 2.8 – $P_b$ en fonction de $E_b/N_0$ (en dB) pour différentes $M$-PAM

et comparer les $E_b/N_0$ nécessaires pour atteindre une certaine probabilité d'erreur cible, notée $P_b^{\text{target}}$, avec

différentes constellations. Ainsi, si nous considérons deux constellations $\mathcal{M}_1$ et $\mathcal{M}_2$, nous avons

$$P_b^{\text{target}} \approx Q\left(\sqrt{\gamma_{\mathcal{M}_1}\left.\frac{E_b}{N_0}\right|_1}\right) = Q\left(\sqrt{\gamma_{\mathcal{M}_2}\left.\frac{E_b}{N_0}\right|_2}\right) \Rightarrow \gamma_{\mathcal{M}_1}\left.\frac{E_b}{N_0}\right|_1 = \gamma_{\mathcal{M}_2}\left.\frac{E_b}{N_0}\right|_2$$

avec $E_b/N_0|_u$ le $E_b/N_0$ de la constellation $\mathcal{M}_u$. En passant au décibel, nous obtenons

$$\left.\frac{E_b}{N_0}\right|_{1,\text{dB}} = \left.\frac{E_b}{N_0}\right|_{2,\text{dB}} + \underbrace{10\log_{10}\left(\frac{\gamma_{\mathcal{M}_2}}{\gamma_{\mathcal{M}_1}}\right)}_{\Theta_{|\text{dB}}=\text{gain } (>0) \text{ ou perte } (<0) \text{ de } \mathcal{M}_2 \text{ par rapport à } \mathcal{M}_1}$$

Le terme $\Theta_{|\text{dB}}$ représente l'écart en $E_b/N_0$, exprimé en dB, qu'il existe entre les deux courbes de performances des constellations comparées. Ceci veut aussi dire que les différentes courbes de la figure 2.8 sont identiques à un décalage en $E_b/N_0$ près les unes par rapport aux autres. Ce terme $\Theta_{|\text{dB}}$ est le moyen le plus commode pour comparer des schémas de modulation (et de codage aussi comme nous le verrons ci-après) entre eux.

### 2.6.3 Extension au cas avec codage correcteur d'erreur du chapitre 1

Nous sommes maintenant en mesure de calculer les performances globales d'un système intégrant un code correcteur d'erreur (cf. la section concernant les performances d'un code au chapitre 1) et une modulation.

Le canal gaussien à temps discret pouvant se modéliser par une probabilité de confondre un bit '0' avec un bit '1' et vice-versa avec une probabilité $p$ donnée par la probabilité d'erreur bit du canal gaussien pour la constellation $\mathcal{M}$ utilisée, le lien entre les bits émis et les bits décodés est bien un canal binaire symétrique de probabilité $p$ obéissant à (2.47). De plus si le SNR est suffisamment grand, la probabilité $p$ donnée par (2.47) devient petit devant 1.

Nous savons qu'un code correcteur binaire linéaire en bloc admet la probabilité d'erreur bit suivante sur un canal binaire symétrique de probabilité d'erreur $p$, si $p \ll 1$ (cf. (1.32)).

$$P_{b|\text{avec codage}} = \delta_t p^{t+1} \tag{2.51}$$

avec, pour rappel, $\delta_t = d_{\min}C_n^{t+1}/n$, $t = \lfloor (d_{\min}-1)/2 \rfloor$ la capacité de correction et $n$ la longueur du mot de code. Bref nous avons

$$P_{b|\text{avec codage}} = \delta_t \beta_{\mathcal{M}}^{t+1} Q^{t+1}\left(\sqrt{R\gamma_{\mathcal{M}}\frac{E_b}{N_0}}\right).$$

Attention, nous avons incorporé une multiplication par le rendement du code correcteur d'erreur $R$ pour prendre en compte le fait que $E_b$ représente l'énergie consommée pour émettre un bit utile. Pour le cas codé, la définition de l'énergie par bit doit être modifié à

$$E_b := \frac{P_x T_b}{R} = \frac{E_s}{R\log_2(M)}, \tag{2.52}$$

où $R$ indique le rendement du code et donc $R\log_2(M)$ le nombre de bits d'information transmis par symbole.

En utilisant l'approximation de la fonction $Q(\cdot)$ de (2.43), nous allons obtenir une expression encore plus parlante de $P_{b|\text{avec codage}}$. En effet, nous pouvons alors écrire que

$$P_{b|\text{avec codage}} = \tilde{\beta}_{\mathcal{M},\text{FEC}} e^{-\frac{\gamma_{\text{FEC}}\cdot\gamma_{\text{M}}}{2}\frac{E_b}{N_0}} \tag{2.53}$$

avec

$$\gamma_{\text{FEC}} = R(t+1) \tag{2.54}$$

et $\tilde{\beta}_{\mathcal{M},\text{FEC}} = \delta_t \beta_{\mathcal{M}}^{t+1}/2^{t+1}$. L'acronyme FEC désigne le code correcteur d'erreur (*Forward Error Correcting coding -FEC-*, en anglais) et le terme $\gamma_{\text{FEC}}$ est appelé **gain de codage** car il correspond au gain en SNR obtenu grâce au codage correcteur d'erreur. Ce gain de codage s'additionne en dB au gain de modulation.

Examinons maintenant quelques gains de codage de codes correcteur d'erreur dont certains ont été vus au chapitre 1.

- **Exemple 1 : code à répétition.** Nous considérons un code à répétition $(n, 1)$. Par conséquent les deux mots de code n'ont aucun bit commun et $d_{\min} = n$ ce qui est le maximum. Le rendement en revanche vaut $1/n$ ce qui est le minimum. Ainsi le gain de codage est égal à $(1/n)(\lfloor (n-1)/2 \rfloor + 1)$ qui est bien approché, quand $n$ est suffisamment grand, par

$$\gamma_{\text{code à répétition}} \approx \frac{1}{2} \quad (= -3 \text{ dB}).$$

  Comme ce gain de codage est inférieur à 1 et donc négatif en dB, **il correspond à une perte !** Donc si on raisonne à énergie par bit utile constante, le codage à répétition est à éviter absolument car il aggrave les performances et évidemment il diminue le débit utile. Espérons que d'autres codes soient plus performants.

- **Exemple 2 : code de Hamming** $(7, 4, 3)$**.** La distance minimale vaut 3 et donc la capacité de correction 1. Le rendement vaut $4/7$ ce qui implique que

$$\gamma_{\text{code de Hamming (7,4,3)}} \approx \frac{8}{7} \quad (= +0.58 \text{ dB}).$$

  **Ouf !** Cela correspond à un gain en performances par rapport au cas non codé. En revanche, le débit est encore diminué d'environ 40%. Il serait en fait intéressant de quantifier l'apport ou non du codage pour des systèmes fonctionnant à même énergie bit utile et à même débit utile. Ceci est fait dans l'exemple 3 suivant.

- **Exemple 3 : code de Reed-Müller** $(128, 64, 16)$**.** Le rendement vaut $1/2$ et la capacité de correction vaut 7. Ainsi le gain de codage vaut

$$\gamma_{\text{code de Reed-Müller (128,64,16)}} \approx 4 \quad (= +6 \text{ dB}).$$

  Nous voulons savoir si un système codé fonctionnant au même débit utile qu'un système non codé peut être plus performant. Pour cela, considérons une transmission d'un bit utile par utilisation de canal. Pour le système sans codage, une 2-PAM sera donc mise en œuvre. Pour le système avec codage dont le code est celui de Reed-Müller, une 4-PAM sera nécessaire. Il faut donc comparer les gains de chaque système

$$\begin{cases} \gamma_{\text{sans codage}} &= \gamma_{2-\text{PAM}} = 2 \\ \gamma_{\text{avec codage}} &= \gamma_{\text{code de Reed-Müller}} \times \gamma_{4-\text{PAM}} = 3.2 \end{cases}.$$

  Par conséquent, le système avec codage permet de gagner 2.05 dB en SNR pour un début utile identique et une consommation énergétique par bit utile identique. **Un codage bien choisi permet donc d'améliorer les systèmes de communication toute chose égale par ailleurs.**

Sur la figure 2.9, nous avons tracé $P_b$ pour différents codes de Hamming en fonction de $E_b/N_0$ quand une constellation 2-PAM est considérée. Pour le code $(7, 4, 3)$, on peut vérifier que le gain de codage annoncé est bien celui constaté.

## 2.7 Lien entre les paramètres de dimensionnement d'un système

Nous avons jusuq'ici introduit un certain nombre de paramètres dont il est important de comprendre leur nature et leurs relations.

Certains paramètres sont **exogènes** puisque le concepteur du système n'a pas de marge de manœuvre sur eux car ils sont contraints par ailleurs, notamment l'application ou la compatibilité électro-magnétique :

- Occupation spectrale via la bande $B$,
- Consommation via l'énergie/puissance par bit $E_b$ ou par symbole $E_s$,
- Qualité de transmission via la probabilité d'erreur bit $P_b$,
- Rapidité de transmission via le débit binaire utile $D_b$.

D'autres paramètres sont **endogènes** puisque l'ingénieur concevant le système a un certain degré de liberté sur leur dimensionnement.

- Filtre d'émission $g(t)$, d'où le temps-symbole $T_s$,

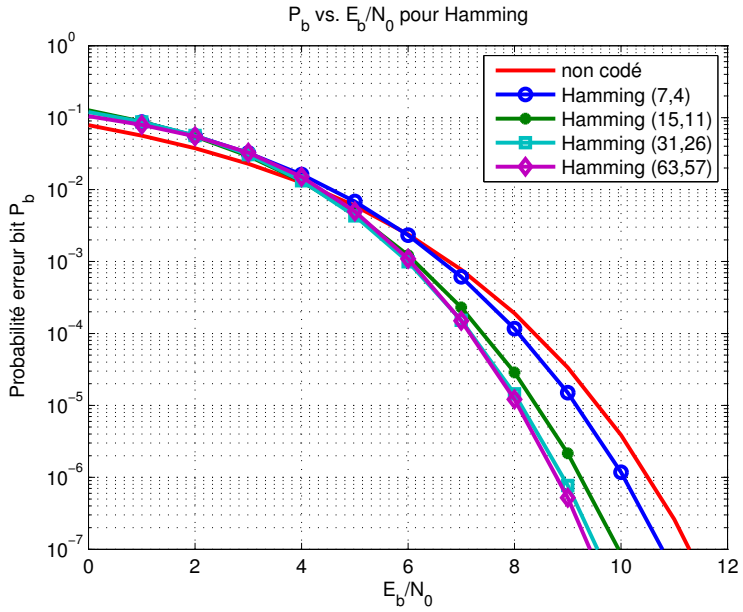



FIGURE 2.9 – $P_b$ pour différents codes de Hamming en 2-PAM en fonction de $E_b/N_0$ (en dB)

- Constellation $\mathcal{M}$, d'où le nombre de symboles $M$,
- Code correcteur d'erreur, d'où la capacité de correction $t$ et et le rendement $R$.

Sur la figure 2.10, nous avons récapitulé le lien entre ces paramètres (trait plein = croissant, trait en tiret = décroissant).

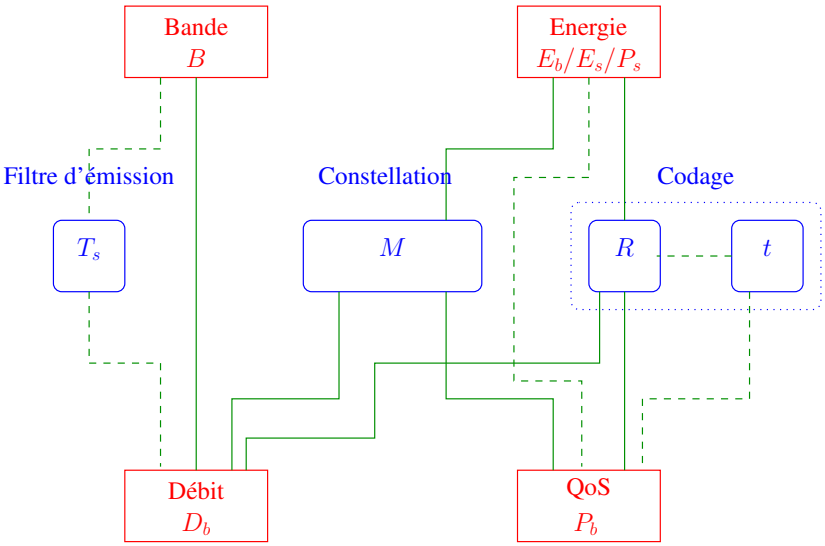



FIGURE 2.10 – Lien entre les paramètres de dimensionnement d'un système

## 2.8 Bilan

Nous récapitulons sur la figure 2.11 les éléments et paramètres intervenant dans un émetteur et récepteur standard si le canal est gaussien. Pour retrouver les liens entre les paramètres, consultez la figure 2.10.

Nous rappelons ci-dessous les concepts de base et savoir-faire concernant ce chapitre à acquérir durant cette unité d'enseignement.

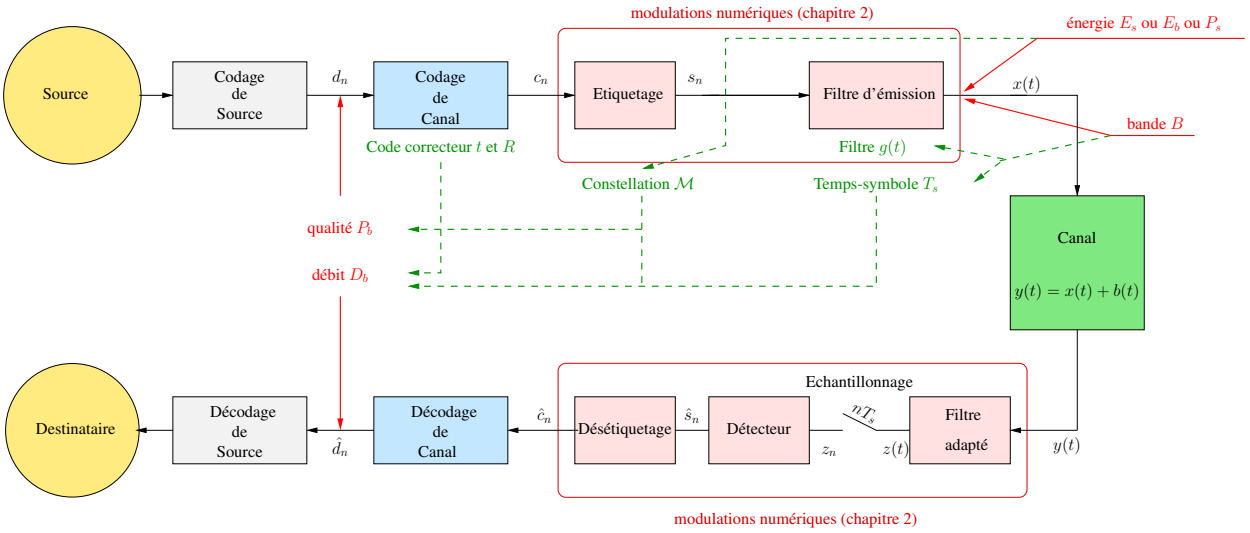



FIGURE 2.11 – Schéma d'un émetteur (TX) et d'un récepteur (RX)

**Les concepts de base :**

- Les éléments présents dans un émetteur/récepteur,
- Le filtre adapté,
- Le filtre de Nyquist,
- Les relations entre les paramètres de dimensionnement.

**Les savoir-faire :**

- Vérifier les propriétés des filtres,
- Calculer des probabilités d'erreur bit pour de nouvelles modulations,
- Comparer des techniques et systèmes entre eux,
- Concevoir un système via ces paramètres pour un cahier des charges donné (cf. TD et exercices ci-dessous).

## 2.9 Exercices

**Exercice 2.1** On désire transmettre, sans IES à la réception, un message numérique à 7200bits/s sur une largeur de bande de 2000Hz par le biais d'une modulation $M$-PAM.

1. Quelle taille minimale de modulation $M$ doit-on choisir ?
2. En déduire alors le facteur d'excès de bande $\rho$ requis si un filtre en racine de cosinus surélevé est mis en œuvre à l'émission ?

**Exercice 2.2** Considérons une transmission utilisant $M$ vecteurs possibles, chacun de taille $N$. Le $m^{\text{ème}}$ vecteur de taille $N$ sera noté $\mathbf{x}_m = [x_{m,0}, \cdots, x_{m,N-1}]^{\mathrm{T}}$ avec $(.)^{\mathrm{T}}$ l'opérateur de transposition. Ces vecteurs correspondent à des mots de codes correcteur d'erreur modulés en 2-PAM.

On notera, dans la suite, $P(\mathbf{x}_m \to \mathbf{x}_{m'})$, la probabilité d'erreur paire définie comme étant la probabilité de confondre le mot $\mathbf{x}_m$ avec le mot $\mathbf{x}_{m'}$ lorsqu'aucun autre mot n'est possible dans le système de transmission.

On considère que le vecteur reçu $\mathbf{z}$ s'écrit de la manière suivante

$$\mathbf{z} = \mathbf{x} + \mathbf{w}$$

avec

— $\mathbf{x}$ un mot d'information appartenant à l'ensemble des $M$ mots possibles $\{\mathbf{x}_m\}_{m=0,\cdots M-1}$. On considère que les mots sont équiprobables.

— $\mathbf{w}$ un bruit gaussien i.i.d. de moyenne nulle et de variance $N_0/2$.

On met en place le récepteur du maximum de vraisemblance (basé sur l'observation $\mathbf{z}$).

1. Calculer $P(\mathbf{x}_m \to \mathbf{x}_{m'})$ en fonction de $N_0$ et $\|\mathbf{x}_m - \mathbf{x}_{m'}\|$ la distance euclidienne entre $\mathbf{x}_m$ et $\mathbf{x}_{m'}$.
2. Quelle est la probabilité d'erreur globale, notée $P_e$. En ne retenant que le terme dominant (utiliser le fait que $Q(a) \ll Q(b)$ si $a > b > 0$), écrire une formule approchée de $P_e$. Quelle est la distance qui rentre en jeu ?
3. En se rappelant que les $x_{m,n}$ sont modulés par une 2-PAM, écrire $P_e$ en fonction de la distance minimale de Hamming du code correcteur d'erreur.
4. Quel est alors le gain de codage ? Pourquoi y a-t-il une différence avec le gain de codage donné en section 2.6.3.

**Exercice 2.3** Dans de nombreux systèmes pratiques (câble optique sous-marin, système cellulaire du futur), l'information peut être relayée afin d'arriver avec une fiabilité accrue au récepteur ou afin d'augmenter la portée du système comme dessiné sur la figure 2.12.

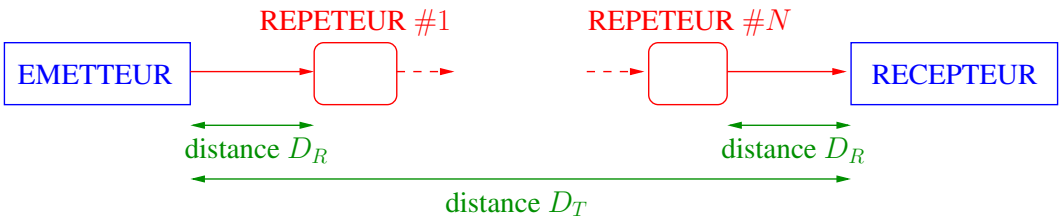



FIGURE 2.12 – Système de communication avec $N$ répéteurs

— La distance entre l'émetteur et le récepteur est fixe et est notée $D_T$.
— Chaque répéteur reçoit un signal analogique et le décode pour retrouver l'information du symbole émis par le répéteur précédent. S'il n'arrive pas à décoder le signal, alors le répéteur ne retransmet rien et le système est en erreur. Cette technique est dite *Decode-and-Forward (DF)*. De plus le lien entre les répéteurs $\#n$ et $\#(n+1)$ est de longueur $D_R$ (variant suivant le nombre de répéteurs) et modélisé par le canal suivant à l'instant $k$

$$z_k^{(n+1)} = s_k^{(n)} + w_k^{(n)}$$

avec $\{s_k^{(n)}\}_n$ des symbole émis **2-PAM** par le répéteur $n$ ($s_k^{(n)}$ est égal à $s_k$ le symbole émis par l'émetteur, s'il n'y a pas d'erreur durant les liens précédents), $r_n^{(k+1)}$ le signal reçu au répéteur $\#(n+1)$ et $w_k^{(n)}$ le bruit blanc gaussien de moyenne nulle et de variance $N_0/2$.
— Chaque lien entre deux répéteurs admet la même probabilité d'erreur symbole, notée $P_e^{(i)}$.
— La probabilité d'erreur symbole globale entre l'émetteur et le récepteur sera notée $P_e$.

1. Montrer que
$$P_e = 1 - (1 - P_e^{(i)})^{N+1}.$$
2. Sur chaque lien (de longueur $d$), on suppose que l'atténuation est la suivante
$$P_{\text{reçue}} = \frac{P_{\text{émise}}}{d^2}$$
avec $P_{\text{émise}}$ la puissance émise et $P_{\text{reçue}}$ la puissance reçue.
En déduire la probabilité d'erreur symbole $P_e^{(i)}$ sur chaque lien en fonction de l'énergie symbole $E_s$ consommée au répéteur et de la distance $D_R$.
3. En déduire la probabilité d'erreur symbole globale $P_e$ en fonction de $E_s$, $N$, $D_T$ et $N_0$.
4. Trouver alors la valeur optimale de $N$ minimisant la probabilité d'erreur. Qu'en conclure ?
5. Maintenant, nous nous fixons une probabilité d'erreur symbole globale cible $P_e^{(0)}$. Caractériser analytiquement le nombre minimal de répéteurs, noté $N_{\min}$, à mettre en place.
6. Application numérique : $E_s/N_0 = 100$dB, $D_T = 1000$km, $P_e^{(0)} = 10^{-3}$. Indication : soit $f(x) = 1 - (1 - Q(\sqrt{2}.10^{-1}(x+1)))^{x+1}$, alors $f(27) > 10^{-3}$ et $f(28) < 10^{-3}$.

**Exercice 2.4** Considérons que nous avons 2 voies de communications indépendantes. Par exemple,

— ces deux voies peuvent être deux fibres optiques dans la même gaine
— ces deux voies peuvent être deux antennes d'émission et de réception qui n'interfèrent pas.
— ces deux voies peuvent être deux routes permettant d'aller d'un point à un autre.
— ces deux voies peuvent correspondre à des fréquences différentes.

Bref, on a le schéma suivant

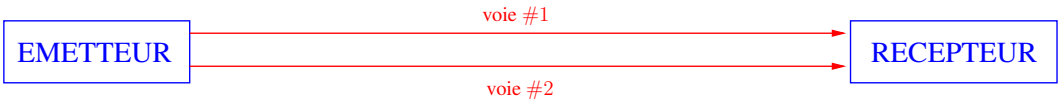



FIGURE 2.13 – Système de communication avec 2 voies parallèles

qui se traduit par l'équation suivante

$$z_k^{(n)} = h^{(n)} s_k^{(n)} + w_k^{(n)}, \quad n = \{1, 2\}$$

avec
— $k$ l'indice du temps et $n$ l'indice de la voie.
— $h^{(n)}$ l'atténuation du canal sur la voie $n$.
— $z_k^{(n)}$ le signal reçu à temps discret.
— $s_k^{(n)}$ les symboles émis appartenant à une modulation $M^{(n)}$-PAM (la taille peut être différente sur les deux voies).
— $w_k^{(n)}$ deux bruits blancs $(/k)$ et indépendants $(/n)$ gaussien de moyenne nulle et de variance $N_0/2$.

1. *Approche 1 : émission de symboles indépendants sur les deux voies au même instant.*
   1.1 Décrire le récepteur Maximum de Vraisemblance (ML) pour une voie.
   1.2 En forçant chacune des voies à offrir une probabilité d'erreur symbole cible $P_e^{(0)}$, donner le débit obtenu par le système pour une énergie $E_s$ par symbole émis et par voie.
2. *Approche 2 : émission du même symbole sur les deux voies au même instant.*
   2.1 Montrer que le récepteur Maximum de Vraisemblance (ML) revient à effectuer une détection à seuil sur le signal suivant
   $$z_k = \frac{h^{(1)} z_k^{(1)} + h^{(2)} z_k^{(2)}}{|h^{(1)}|^2 + |h^{(2)}|^2}.$$
   2.2 Ecrire $z_k$ en fonction de $s_k$ et d'un bruit équivalent dont on calculera la variance. En déduire la probabilité d'erreur symbole.
   2.3 Quel est le débit atteignable offrant une probabilité d'erreur cible $P_e^{(0)}$ pour une énergie $E_s$ par symbole émis et par voie.
3. En supposant $h^{(1)} = h^{(2)} = 1$, quelle est la meilleure approche pour maximiser le débit tout en assurant la probabilité d'erreur symbole cible $P_e^{(0)}$.

**Exercice 2.5** Comme à l'exercice 2.4, on suppose transmettre sur deux voies. Nous utilisons les mêmes notations. On s'autorise juste à utiliser des énergies par symbole différentes sur les deux voies. Donc on note par $E_s^{(n)}$ l'énergie symbole utilisée sur la voie $n$.

1. Quelle est la capacité du système à deux voies en fonction de $N_0$, $E_s^{(1)}$, $E_s^{(2)}$, $h^{(1)}$ et $h^{(2)}$.
2. On force $E_s^{(1)} = E_s^{(2)} = E_s$ et on veut savoir quel est le meilleur canal. Montrer que le canal $(h^{(1)}, h^{[2)})$ normalisé en énergie moyenne par voie, c'est-à-dire, $|h^{(1)}|^2 + |h^{[2)}|^2 = 2$ qui optimise la capacité de la question 1. est
   $$h^{(1)} = h^{[2)} = 1.$$
3. On s'autorise de nouveau à avoir $E_s^{(1)} \neq E_s^{(2)}$. En revanche, on force une contrainte d'énergie globale
   $$E_s^{(1)} + E_s^{(2)} = 2E_s.$$
   Trouver les meilleurs énergies par voie en fonction de $N_0$, $E_s$, $h^{(1)}$ et $h^{(2)}$.

# Chapitre 3

# Théorie de l'information

## 3.1 Introduction

Dans le chapitre 1, vous avez découvert les codes correcteur d'erreur avec notamment le principe de construction de ces codes ainsi que la règle de décodage optimal, dite règle du maximum de vraisemblance (ML). Vous avez vu que ces codes décodés optimalement permettaient de corriger ou de détecter un certain nombre d'erreurs introduites lors d'une communication sur un canal (typiquement le canal binaire symétrique). Dans ce chapitre, nous allons changer de point de vue puisque nous allons nous intéresser aux performances ultimes (c'est-à-dire, qu'un code ne pourra dépasser) d'un système de communication. Dans le chapitre 1, vous avez vu qu'un code correcteur de rendement $R = k/n$ où $k$ est le nombre de bits d'information envoyés pendant $n$ utilisations de canal pouvait améliorer la probabilité d'erreur du système si sa distance minimale de Hamming était augmentée par rapport au cas non codé (cf. section 1.6). Par performances ultimes, nous entendons : quels peuvent être les gains espérés en probabilité d'erreur par l'utilisation de codes de rendement $R$ strictement positif. Vous verrez que la réponse à cette question se formule en fait de manière radicalement différente puisque nous allons montrer que **la probabilité d'erreur peut être rendue arbitrairement faible pour peu que $R$ ne dépasse pas un certain seuil que nous appellerons capacité du canal et ceci sous l'hypothèse que $k$ et $n$ tendent vers l'infini**.

Pour arriver à ce résultat fondamental, ce chapitre sera organisé de la manière suivante :

- en section 3.2, nous introduisons les notions d'**entropie** et d'**information mutuelle** parce qu'elles vont jouer un rôle essentiel dans la définition de la capacité.

- en section 3.3, nous rentrons dans le vif du sujet en donnant la définition de la capacité d'un canal discret sans mémoire. Nous y énonçons le théorème principal de ce chapitre démontré par **C.E Shannon en 1948** et qui dit en substance que pour tout rendement $R > 0$ inférieur à la *capacité* $C$, il est possible de communiquer avec une probabilité d'erreur aussi faible que l'on souhaite pour peu que l'on choisisse une taille de mot de code $n$ suffisamment grande. En revanche, pour tout rendement $R$ supérieur à la capacité $C$, cela n'est pas possible.

- en section 3.4, nous montrons quelques expressions de la capacité pour les canaux discrets sans mémoire les plus communs comme le canal binaire symétrique, le canal à effacement et même le canal gaussien.

En section 3.5, nous faisons un bilan de ce chapitre en rappelant les éléments essentiels à retenir et les outils à maîtriser. Dans ce polycopié, par souci de simplicité et faute de temps, la plupart des preuves associées à ce chapitre sont omises. Un cours complet sur la théorie de l'information est donné en deuxième année dans la filière ACCQ ainsi qu'en troisième année dans le master de l'Institut Polytechnique de Paris dénommé MICAS. Tant dans ce chapitre qu'en deuxième année, le système se ramène à une communication dite point-à-point (typiquement entre un mobile et une station de base, ou entre un modem ADSL et le DSLAM du central téléphonique). Afin d'améliorer les réseaux, un système ne doit plus se concevoir comme une juxtaposition de communications point-à-point mais d'emblée comme une communication multipoint-à-multipoint ce qui implique de développer la théorie de l'information pour des flux (et non un flux) d'information. Ceci s'appelle la théorie de l'information pour les réseaux (*Network Information Theory*) et fait l'objet de recherche intense actuellement et elle est partiellement enseignée en troisième année.

Dans la suite du chapitre, sauf indication contraire, tous les variables aléatoires sont discrètes sur un alphabet fini.

## 3.2 Entropies et Information mutuelle

### 3.2.1 Entropie : mesure d'incertitude

L'entropie va permettre de capter le degré d'incertitude contenu dans une variable aléatoire. Dans ce polycopié on s'en servira pour caracteriser les taux de transmission atteignables d'un système de communication. Mais l'entropie peut être utilisée pour décrire les limites fondamentales de plein d'autres systèmes, comme les systèmes de compression ou la génération de clés secrètes.

Avant de donner une définition mathématique précise de l'entropie, nous allons discuter de quelques propriétés que l'entropie doit intuitivement satisfaire. Par exemple, une variable aléatoire $X$ prenant toujours la même valeur, disons 42, n'a pas d'incertitude puisqu'elle est égale à coup sûr à son unique valeur, ici 42. Une telle variable aléatoire sera dite déterministe. Dans ce cas, l'entropie de $X$ doit être égale à 0. A contrario, toute variable aléatoire qui n'est pas déterministe et donc de réalisation incertaine, doit présenter une entropie strictement plus grande que 0.

D'autres propriétés souhaitables ressortent si on compare l'entropie de différentes variables aléatoires. Par exemple, si on considère deux variables aléatoires binaires $X$ et $Y$ prenant leurs valeurs dans $\{0,1\}$ pour lesquelles $X$ admet une loi de probabilité quelconque alors que $Y$ admet une loi uniforme (c'est-à-dire que $Y$ est égale à 0 avec une probabilité 1/2 et égale à 1 avec une probabilité 1/2). Intuitivement $Y$ est plus incertaine que $X$ dans le sens où il est plus difficile de deviner la valeur de la réalisation de $Y$, et donc l'entropie de $Y$ doit être plus grande que l'entropie de $X$. Plus généralement, on s'attend à ce qu'une variable uniforme ait une plus grande entropie parmi toutes les variables aléatoires ayant le même alphabet.

Nous verrons que la définition 3.1 de l'entropie admet les propriétés décrites ci-dessus, ainsi que d'autres propriétés très intuitives que l'on peut exiger d'une mesure d'incertitude.

**Définition 3.1 (Entropie)** *Soit $X$ une variable aléatoire sur un alphabet fini $\mathcal{X}$ avec la loi de probabilité $P_X(\cdot)$. L'entropie de cette variable aléatoire est*

$$H(X) := -\sum_{x\in\mathcal{X}} P_X(x)\log_2(P_X(x)) \quad [\text{bits}], \tag{3.1}$$

*avec, par convention, $0\cdot\log_2(0) = 0$.*

Notez que l'entropie d'une variable aléatoire $X$ ne dépend pas des valeurs que cette variable aléatoire prend, c'est-à-dire des éléments de l'alphabet $\mathcal{X}$, mais uniquement des probabilités $P_X(\cdot)$ que ces valeurs prennent. En particulier, une variable qui prend les valeurs 0 et 1 avec probabilité 1/2 chacune a la même entropie qu'une variable aléatoire qui prend les valeurs $-100$ et 345 avec probabilité 1/2 chacune.

**Exemple 3.1 (Entropie d'une variable uniforme)** *Soit $X$ une variable aléatoire de loi uniforme sur tout l'alphabet $\mathcal{X}$, c'est-à-dire,*

$$P_X(x) = \frac{1}{|\mathcal{X}|}, \quad \forall x\in\mathcal{X}, \tag{3.2}$$

*où $|\mathcal{X}|$ désigne le cardinal (le nombre d'éléments) de l'alphabet $\mathcal{X}$.*
*L'entropie de $X$ vaut*

$$\begin{aligned} H(X) &= -\sum_{x\in\mathcal{X}} P_X(x)\log_2(P_X(x)) = -\sum_{x\in\mathcal{X}} \frac{1}{|\mathcal{X}|}\log_2\left(\frac{1}{|\mathcal{X}|}\right) = |\mathcal{X}|\cdot\frac{1}{|\mathcal{X}|}\log_2(|\mathcal{X}|) \\ &= \log_2(|\mathcal{X}|). \end{aligned} \tag{3.3}$$

**Exemple 3.2 (Entropie d'une variable déterministe)** *Soit $X$ une variable déterministe qui prend toujours la valeur $x_0\in\mathcal{X}$, c'est-à-dire,*

$$P_X(x) = \begin{cases} 1, & \text{si } x = x_0 \\ 0, & \text{ailleurs} \end{cases}. \tag{3.4}$$

*L'entropie de $X$ vaut*

$$H(X) = -\sum_{x\in\mathcal{X}} P_X(x)\log_2(P_X(x)) = -1\log_2(1) - \sum_{x\in\mathcal{X}\setminus x_0} 0\cdot\log_2(0) = 0, \tag{3.5}$$

*où la dernière somme est égale à* 0 *parce que, par convention, nous avons* $0 \cdot \log_2(0) = 0$.

Les deux exemples 3.1 et 3.2 traitent en fait des deux cas extrêmes de l'entropie. En effet, nous avons le résultat suivant.

**Résultat 3.1 (Valeurs extrêmes de l'entropie)** *Pour toute variable aléatoire* $X$ *sur un alphabet fini* $\mathcal{X}$*, l'entropie de* $X$ *satisfait*

$$0 \leq H(X) \leq \log_2(|\mathcal{X}|). \tag{3.6}$$

*En outre,*

— $H(X) = 0$ *si et seulement si* $X$ *est déterministe*

— $H(X) = \log_2(|\mathcal{X}|)$ *si et seulement si* $X$ *est uniforme sur tout* $\mathcal{X}$.

La définition 3.1 de l'entropie a donc bien les propriétés intuitives souhaitées décrites dans les deux premiers paragraphes de cette sous-section. Continuons avec un troisième exemple, moins extrême, pour nous convaincre une dernière fois du lien entre entropie et degré d'incertitude.

**Exemple 3.3** *Soient* $X$ *et* $Y$ *deux variables aléatoires binaires qui indiquent s'il y a du soleil en Martinique (*$X$*) et à Paris (*$Y$*). Ainsi* $X$ *et* $Y$ *ont le même alphabet,* $\mathcal{X} = \mathcal{Y} = \{0, 1\}$ *où* 0 *représente de la pluie et* 1 *du soleil. De plus*

— $X = 0$ *(pluie en Martinique) admet une probabilité de* $1/4$ *et* $X = 1$ *(soleil en Martinique) une probabilité de* $3/4$.

— $Y = 0$ *(pluie à Paris) admet une probabilité de* $5/8$ *et* $Y = 1$ *(soleil à Paris) avec probabilité de* $3/8$.

*L'entropie de* $X$ *(temps à Martinique) se calcule de la manière suivante*

$$\begin{aligned} H(X) &= -\sum_{x \in \mathcal{X}} P_X(x) \log_2(P_X(x)) \\ &= -P_X(0) \log_2(P_X(0)) - P_X(1) \log_2(P_X(1)) \\ &= -(1/4) \log_2(1/4) - (3/4) \log_2(3/4) \\ &\approx 0.8113, \end{aligned}$$

*ce qui est bien entre* 0 *et* $\log_2(2) = 1$ *comme indiqué par le résultat 3.1.*

*L'entropie de* $Y$ *(temps à Paris) se calcule de la manière suivante*

$$\begin{aligned} H(Y) &= -\sum_{y \in \mathcal{Y}} P_Y(y) \log_2(P_Y(y)) \\ &= -P_Y(0) \log_2(P_Y(0)) - P_Y(1) \log_2(P_Y(1)) \\ &= -(5/8) \log_2(5/8) - (3/8) \log_2(3/8) \\ &\approx 0.9544. \end{aligned}$$

*On voit donc que le temps à Paris a une entropie plus grande que le temps en Martinique, ce qui correspond bien à notre intuition que le temps à Paris est plus incertain que celui en Martinique.*

Dans ce chapitre, souvent on parlera d'entropie pour des variables aléatoires binaires. C'est pourquoi la définition suivante nous sera fort utile car nous permettra de rendre nos futures expressions analytiques plus compactes.

**Définition 3.2 (Fonction d'entropie binaire)** *La* fonction d'entropie binaire *est définie comme suit*

$$H_{\mathrm{b}}(p) := -p \log_2(p) - (1-p) \log_2(1-p), \tag{3.7}$$

*où le seul argument* $p$ *doit être un nombre réel dans l'intervalle* $[0, 1]$.

*Notez que* $H_{\mathrm{b}}(p)$ *est égale à l'entropie d'une variable aléatoire binaire* $X$*,*

$$H_{\mathrm{b}}(p) = H(X), \tag{3.8}$$

*si* $X$ *est* Bernoulli*(*$p$*), c'est-à-dire, si* $X$ *prend la valeur* 1 *avec une probabilité* $p$ *et la valeur* 0 *avec une probabilité* $(1-p)$.

Nous avons tracé $H_{\mathrm{b}}(.)$ sur la figure 3.1. Notez que cette fonction $H_{\mathrm{b}}(.)$ est strictement concave [1] et symétrique par rapport à la valeur $1/2$. Sa valeur minimale est atteinte en $p = 0$ et $p = 1$ où elle est égale à 0. Sa valeur maximale est atteinte en $p = 1/2$ où elle est égale à 1. La fonction est strictement croissante entre 0 et $1/2$ et strictement décroissante entre $1/2$ et 1.

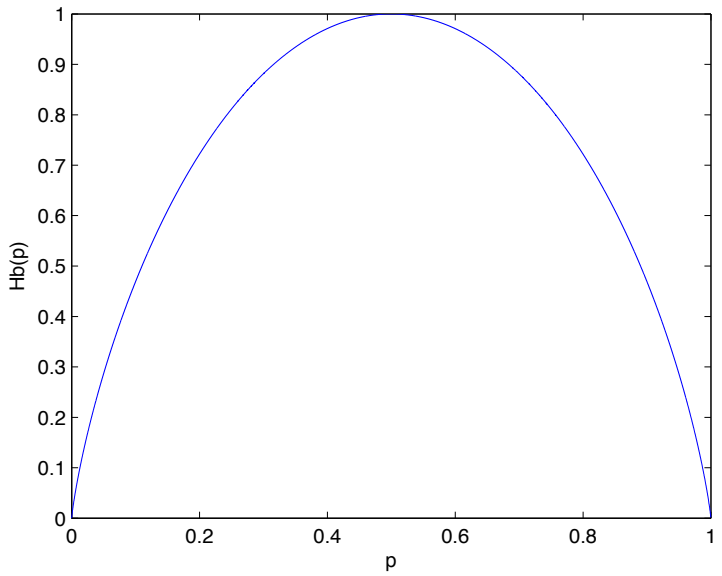



FIGURE 3.1 – Fonction d'entropie binaire $H_{\mathrm{b}}(.)$

### 3.2.2 Entropie conjointe

L'entropie conjointe représente une généralisation de l'entropie à une *paire* de variables aléatoires. Notons d'abord que toute paire $(X, Y)$ de deux variables aléatoires sur des alphabets finis $\mathcal{X}$ et $\mathcal{Y}$ peut être vue comme une variable aléatoire avec un alphabet fini plus grand égal au produit cartesien $\mathcal{X} \times \mathcal{Y}$. Donc, la définition 3.1 s'applique aussi à ce cas et on obtient la définition suivante pour l'entropie d'une paire de variables aléatoires, dite aussi *entropie conjointe*.

**Définition 3.3 (Entropie conjointe)** *Soient $X$ et $Y$ deux variables aléatoires sur des alphabets discrets $\mathcal{X}$ et $\mathcal{Y}$ avec comme loi de probabilité conjointe $P_{XY}(x, y)$.* L'entropie conjointe *de cette paire de variables aléatoires est définie comme*

$$H(X, Y) := -\sum_{x \in \mathcal{X}, y \in \mathcal{Y}} P_{XY}(x, y) \log_2(P_{XY}(x, y)). \tag{3.9}$$

Notez que l'entropie conjointe est symétrique par rapport à ces deux arguments, c'est-à-dire, qu'on a toujours $H(X, Y) = H(Y, X)$. De plus, l'entropie jointe ne mesure pas les incertitudes des deux variables aléatoires individuellement, mais l'incertitude qu'il existe sur l'ensemble des deux. En particulier, elle prend en compte les liens qui existent entre $X$ et $Y$, comme on peut le voir sur les deux exemples 3.4 et 3.5 suivants.

**Exemple 3.4 (Variables indépendantes)** *Soient $X$ et $Y$ deux variables aléatoires indépendantes sur des*

1. strictement concave signifie que pour tout $\lambda, p, q \in [0, 1]$,

$$H_{\mathrm{b}}(\lambda p + (1 - \lambda)q) \geq \lambda H_{\mathrm{b}}(p) + (1 - \lambda) H_{\mathrm{b}}(q),$$

avec égalité si et seulement si $\lambda = 0$, $\lambda = 1$, ou $p = q$.

*alphabets finis $\mathcal{X}$ et $\mathcal{Y}$. Ceci implique que $P_{XY}(x,y) = P_X(x)P_Y(y)$ pour tous $x \in \mathcal{X}$ et $y \in \mathcal{Y}$. Donc :*

$$\begin{aligned} H(X,Y) &= -\sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} P_{XY}(x,y)\log_2(P_{XY}(x,y)) \\ &= -\sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} P_X(x)P_Y(y)\log_2(P_X(x)\cdot P_Y(y)) \\ &= -\sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} P_X(x)P_Y(y)\log_2(P_X(x)) - \sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} P_X(x)P_Y(y)\log_2(P_Y(y)) \\ &= -\sum_{x\in\mathcal{X}}\Big(\sum_{y\in\mathcal{Y}} P_Y(y)\Big)P_X(x)\log_2(P_X(x)) - \sum_{y\in\mathcal{Y}}\Big(\sum_{x\in\mathcal{X}} P_X(x)\Big)P_Y(y)\log_2(P_Y(y)) \\ &= -\sum_{x\in\mathcal{X}} P_X(x)\log_2(P_X(x)) - \sum_{y\in\mathcal{Y}} P_Y(y)\log_2(P_Y(y)) \\ &= H(X) + H(Y), \end{aligned} \tag{3.10}$$

*où dans la troisième égalité nous avons utilisé que pour tout nombre réel $a, b > 0$, $\log_2(ab) = \log_2(a) + \log_2(b)$ et dans la cinquième égalité nous avons utilisé que $\sum_{y\in\mathcal{Y}} P_Y(y) = 1$ et $\sum_{x\in\mathcal{X}} P_X(x) = 1$.*

**Exemple 3.5 (Variables identiques)** *Soit $X = Y$, c'est-à-dire, $\mathcal{X} = \mathcal{Y}$ et pour tout $x \in \mathcal{X}$, $P_{XY}(x,y) = P_X(x)$ si $x = y$ et $P_{XY}(x,y) = 0$ si $y \neq x$. Donc, nous avons*

$$H(X,Y) = -\sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} P_{XY}(x,y)\log_2(P_{XY}(x,y)) = -\sum_{x\in\mathcal{X}} P_X(x)\log_2 P_X(x) = H(X). \tag{3.11}$$

En fait, les deux exemples 3.4 et 3.5 traitent des deux cas extrêmes de l'entropie conjointe. C'est pourquoi, nous pouvons énoncer le résultat suivant.

**Résultat 3.2 (Valeurs extrêmes de l'entropie conjointe)** *Pour toute paire de variables aléatoires $X$ et $Y$ sur des alphabets finis $\mathcal{X}$ et $\mathcal{Y}$, l'entropie conjointe $H(X,Y)$ satisfait*

$$\max\{H(X), H(Y)\} \leq H(X,Y) \leq H(X) + H(Y). \tag{3.12}$$

*En outre,*
- *$H(X,Y) = H(X)$ si et seulement si $Y = g(X)$ pour une fonction déterministe $g$ quelconque.*
- *$H(X,Y) = H(X) + H(Y)$ si et seulement si $X$ et $Y$ sont indépendants.*

**Exemple 3.6** *Soit $X$ une variable aléatoire déterminant le lieu où se trouve Scarlett Johansson en ce moment : $X = 1$ veut dire qu'elle est à Paris et $X = 0$ qu'elle est à Hollywood. Soit $Y$ une variable aléatoire déterminant la météo du lieu dans lequel se trouve Scarlett Johansson en ce moment : $Y = 1$ veut dire qu'elle a du soleil et $Y = 0$ veut dire qu'il pleut. Les deux variables aléatoires $X$ et $Y$ ont la loi de probabilité conjointe suivante*

$$P_{XY}(0,0) = 0, \quad P_{XY}(1,0) = 1/3, \quad P_{XY} = (1,1) = 1/3, \quad P_{XY}(0,1) = 1/3.$$

*L'entropie conjointe de $X$ et $Y$ est donc*

$$\begin{aligned} H(X,Y) &= -\sum_{x\in\mathcal{X},y\in\mathcal{Y}} P_{XY}(x,y)\log_2(P_{XY}(x,y)) \\ &= -0\log_2(0) - (1/3)\log_2(1/3) - (1/3)\log_2(1/3) - (1/3)\log_2(1/3) \\ &= \log_2(3) \approx 1.585. \end{aligned}$$

Notons aussi que, grâce à la formule de loi marginale, on obtient

$$\begin{cases} P_X(0) &= \sum_{y\in\mathcal{Y}} P_{XY}(x,y) = P_{XY}(0,0) + P_{XY}(0,1) = 0 + 1/3 = 1/3 \\ P_X(1) &= 1 - P_X(0) = 2/3 \end{cases}$$

Donc, l'entropie de $X$ vaut

$$\begin{aligned} H(X) &= -\sum_{x\in\mathcal{X}} P_X(x)\log_2(P_X(x)) \\ &= -(1/3)\log_2(1/3) - (2/3)\log_2(2/3) \\ &\approx 0.925. \end{aligned}$$

De la même façon, on obtient que $H(Y) \approx 0.925$. Ainsi, sur cet exemple, nous retrouvons bien le résultat 3.2 puisque

$$H(X) = H(Y) = 0.925 < H(X,Y) = 1.585 < H(X) + H(Y) = 1.85,$$

### 3.2.3 Entropie conditionnelle

Grâce à l'*entropie conditionnelle* d'une variable aléatoire $X$ par rapport à une autre variable aléatoire $Y$, on va capturer l'incertitude qu'on a sur $X$, une fois qu'on a observé $Y$.

Notez que cette entropie conditionnelle représente une moyenne, parce que l'incertitude qu'on a sur $X$ peut varier selon les différentes réalisations de $Y$. Nous définissons donc d'abord l'entropie conditionnelle de $X$ sachant que $Y$ est égale à une réalisation $y \in \mathcal{Y}$. Soit $P_{X|Y}(x|y)$ la loi de probabilité conditionnelle de $X$ si $Y = y$.

$$H(X|Y=y) := -\sum_{x\in\mathcal{X}} P_{X|Y}(x|y)\log_2(P_{X|Y}(x|y)). \tag{3.13}$$

L'entropie conditionnelle de $X$ sachant $Y$ est la moyenne de toutes les entropies conditionnelles $H(X|Y=y)$ moyennées avec la loi de probabilité $P_Y(\cdot)$.

**Définition 3.4 (Entropie conditionnelle)** *Soient $X$ et $Y$ deux variables aléatoires sur des alphabets finis $\mathcal{X}$ et $\mathcal{Y}$ avec une loi de probabilité conjointe $P_{XY}(x,y)$.* L'entropie conditionnelle de $X$ sachant $Y$ *est*

$$\begin{aligned} H(X|Y) &:= \sum_{y\in\mathcal{Y}} P_Y(y)H(X|Y=y) && (3.14)\\ &= -\sum_{y\in\mathcal{Y}} P_Y(y)\sum_{x\in\mathcal{X}} P_{X|Y}(x|y)\log_2(P_{X|Y}(x|y)) && (3.15)\\ &= -\sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} P_{XY}(x,y)\log_2(P_{X|Y}(x|y)). && (3.16)\end{aligned}$$

*Les deux dernières égalités ont été obtenues grâce à la note de bas de page*[2].

**Exemple 3.7 (Variables indépendantes)** *Soient $X$ et $Y$ deux variables aléatoires indépendantes sur des alphabets $\mathcal{X}$ et $\mathcal{Y}$. Ceci implique que $P_{XY}(x,y) = P_X(x)P_Y(y)$ et $P_{X|Y}(x|y) = P_X(x)$ pour tous $x \in \mathcal{X}$ et $y \in \mathcal{Y}$. Donc*

$$\begin{aligned} H(X|Y) &= -\sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} P_{XY}(x,y)\log_2(P_{X|Y}(x|y))\\ &= -\sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} P_{XY}(x,y)\log_2(P_X(x))\\ &= -\sum_{x\in\mathcal{X}}\Big(\sum_{y\in\mathcal{Y}} P_Y(y)\Big)P_X(x)\log_2(P_X(x))\\ &= -\sum_{x\in\mathcal{X}} P_X(x)\log_2(P_X(x))\\ &= H(X). && (3.18)\end{aligned}$$

**Résultat 3.3 (Valeurs extrêmes de l'entropie conditionnelle)** *Pour toute paire de variables aléatoires $X$ et $Y$ sur des alphabets $\mathcal{X}$ et $\mathcal{Y}$, l'entropie conditionnelle $H(X|Y)$ satisfait*

$$0 \le H(X|Y) \le H(X). \tag{3.19}$$

*En outre,*

2. Selon la règle de Bayes,

$$P_{X|Y}(x|y) = \frac{P_{XY}(x,y)}{P_Y(y)}.$$

Grâce à la formule de la loi marginale, on a également

$$P_{X|Y}(x|y) = \frac{P_{XY}(x,y)}{\sum_{x\in\mathcal{X}} P_{XY}(x,y)}. \tag{3.17}$$

— $H(X|Y) = 0$ *si et seulement si* $X = f(Y)$ *pour une fonction déterministe* $f$ *quelconque. (En effet, une fois qu'on a observé* $Y$*, on connaît parfaitement* $X$ *et donc on n'a plus d'incertitude.)*
— $H(X|Y) = H(X)$ *si et seulement si* $X$ *et* $Y$ *sont indépendants. (En effet,* $Y$ *ne révèle rien sur les valeurs de* $X$ *et donc l'entropie de* $X$ *reste inchangée en observant* $Y$*.)*

Regardons maintenant l'exemple 3.6 sous l'œil de l'entropie conditionnelle et non de l'entropie conjointe.

**Exemple 3.8** *Soient* $X$ *et* $Y$ *définies comme dans l'exemple 3.6. Nous calculons l'entropie de* $X$ *sachant* $Y$*, c'est-à-dire l'incertitude qui existe sur le lieu où se trouve Scarlett Johansson une fois qu'on sait si elle a du soleil ou de la pluie. Nous calculons d'abord l'entropie sur le lieu* $X$ *en sachant qu'elle a du soleil (*$Y = 0$*). Nous utilisons d'abord l'équation* (3.17) *pour obtenir les probabilités conditionnelles :*

$$\begin{cases} P_{X|Y}(0|0) &= \frac{P_{XY}(0,0)}{P_{XY}(0,0)+P_{XY}(1,0)} = \frac{0}{1/3} = 0 \\ P_{X|Y}(1|0) &= 1 - P_{X|Y}(0|0) = 1 \end{cases} \tag{3.20}$$

*Par l'équation* (3.13)*, nous obtenons donc*

$$H(X|Y=0) = -\sum_{x\in\mathcal{X}} P_{X|Y}(x|0)\log_2(P_{X|Y}(x|0)) = -0\cdot\log_2(0) - 1\cdot\log_2(1) = 0. \tag{3.21}$$

*De manière analogue, on obtient que*

$$\begin{cases} P_{X|Y}(0|1) &= \frac{P_{XY}(0,1)}{P_{XY}(0,1)+P_{XY}(1,1)} = \frac{1/3}{2/3} = 1/2 \\ P_{X|Y}(1|1) &= 1 - P_{X|Y}(0|1) = 1/2. \end{cases} \tag{3.22}$$

*et donc*

$$H(X|Y=1) = -\sum_{x\in\mathcal{X}} P_{X|Y}(x|1)\log_2(P_{X|Y}(x|1)) = -(1/2)\log_2(1/2) - (1/2)\log_2(1/2) = 1. \tag{3.23}$$

*Donc, l'entropie conditionnelle est égale à*

$$H(X|Y) = P_Y(0)H(X|Y=0) + P_Y(1)H(X|Y=1) = 1/3\cdot 0 + 2/3\cdot 1 = 2/3. \tag{3.24}$$

### 3.2.4 La règle de chainage

Si nous combinons les résultats de l'exemple 3.8 avec ceux de l'exemple 3.6, nous remarquons que

$$H(X,Y) = H(Y) + H(X|Y) = H(X) + H(Y|X). \tag{3.25}$$

Cette équation est *toujours valable* pour n'importe quel $X$ et $Y$. L'équation est connue sous le nom de *règle de chaînage* (*chain rule*) et est une des équations les plus utilisées en théorie de l'information. Elle signifie que l'incertitude sur une paire de variables aléatoires $(X,Y)$ est égale à l'incertitude qu'on a sur $X$ plus l'incertitude qu'on a sur $Y$ une fois qu'on connaît $X$ ou vice-versa. Autrement dit, le nombre de bits qu'il faut pour décrire la paire $X,Y$ est égal au nombre de bits qui sont nécessaires pour décrire $X$ plus le nombre de bits nécessaires pour décrire $Y$ une fois que $X$ est connue. Notez également que comme $H(X)$ peut être différent de $H(Y)$, cela implique que généralement $H(X|Y)$ est différent de $H(Y|X)$ !

**Résultat 3.4 (Règle de chaînage)** *Pour toute paire de variables aléatoires* $X$ *et* $Y$ *sur des alphabets* $\mathcal{X}$ *et* $\mathcal{Y}$ *:*

$$H(X,Y) = H(X) + H(Y|X) = H(Y) + H(X|Y). \tag{3.26}$$

La preuve de ce résultat découle simplement des définitions et du fait que $\log_2(a\cdot b) = \log_2(a) + \log_2(b)$ pour toutes valeurs $a, b \geq 0$. Plus précisémment :

$$\begin{aligned} H(X,Y) &= -\sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} P_{XY}(x,y)\log_2(P_{X,Y}(x,y)) \\ &= -\sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} P_{XY}(x,y)\log_2(P_X(x)\cdot P_{Y|X}(y|x)) \\ &= -\sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} P_{XY}(x,y)\log_2(P_X(x)) - \sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} P_{XY}(x,y)\log_2(P_{Y|X}(y|x)) \\ &= -\sum_{x\in\mathcal{X}} P_X(x)\log_2(P_X(x)) - \sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} P_{XY}(x,y)\log_2(P_{Y|X}(y|x)) \\ &= H(X) + H(Y|X). \end{aligned} \tag{3.27}$$

De façon similaire on peut démontrer aussi que $H(X,Y) = H(Y) + H(X|Y)$.

La règle de chainage s'étend aussi au cas avec plus de 2 variables aléatoires et à l'entropie conditionelle. Donc, pour trois variables aléatoires $X, Y, Z$ sur des alphabets finis $\mathcal{X}$, $\mathcal{Y}$, et $\mathcal{Z}$ :

$$H(X,Y,Z) = H(X) + H(Y|X) + H(Z|Y,X), \tag{3.28}$$

où l'entropie conjointe de trois variables aléatoires $(X,Y,Z)$ est définie de façon analogue à l'entropie conjointe pour deux variables aléatoires mais en utilisant la loi du triplet $P_{XYZ}$ au lieu de $P_{XY}$ :

$$H(X,Y,Z) := -\sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}}\sum_{z\in\mathcal{Z}} P_{XYZ}(x,y,z)\log_2(P_{X,Y,Z}(x,y,z)), \tag{3.29}$$

et où l'entropie conditionelle sur deux variables aléatoire est obtenue à partir de la définition d'entropie conditionelle en remplaçant la variable conditionnée simple par une paire :

$$H(Z|X,Y) := -\sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}}\sum_{z\in\mathcal{Z}} P_{XYZ}(x,y,z)\log_2(P_{Z|X,Y}(z|x,y)). \tag{3.30}$$

Des fois, il peut aussi s'avérer utile de calculer l'entropie conditionelle $H(Z|X,Y)$ en prenant l'espérance sur $Y$ :

$$H(Z|X,Y) = \sum_{y\in\mathcal{Y}} P_Y(y)H(Z|X,Y=y), \tag{3.31}$$

où $H(Z|X,Y=y)$ est l'entropie conditionelle de $Z$ sachant $X$ quand on fixe $Y=y$ :

$$H(Z|X,Y=y) := -\sum_{x\in\mathcal{X}}\sum_{z\in\mathcal{Z}} P_{XZ|Y}(x,z|y)\log_2(P_{Z|X,Y}(z|x,y)). \tag{3.32}$$

### 3.2.5 Information mutuelle

Grâce à l'*information mutuelle*, on veut quantifier l'information commune entre $X$ et $Y$, autrement dit l'information obtenue sur une variable aléatoire $X$ en révélant une deuxième variable $Y$. Intuitivement on s'attend à ce que cette information soit nulle si $X$ et $Y$ sont indépendants et qu'elle soit maximale si $X = Y$ dans le cas où $Y$ révèle tout sur $X$. Comme le mot l'indique, l'information est mutuelle ce qui veut dire que l'information que $X$ révèle sur $Y$ est la même que $Y$ révèle sur $X$.

Techniquement parlant, l'information mutuelle est définie comme la réduction en incertitude (entropie) sur $X$ qu'on obtient en révélant $Y$.

**Définition 3.5 (Information mutuelle)** *Soient $X$ et $Y$ deux variables aléatoires sur des alphabets finis $\mathcal{X}$ et $\mathcal{Y}$ avec une loi de probabilité conjointe $P_{XY}(x,y)$.* L'information mutuelle de $X$ et $Y$ *est*

$$\begin{aligned} I(X;Y) &:= H(X) + H(Y) - H(X,Y) && (3.33)\\ &= H(X) - H(X|Y) = H(Y) - H(Y|X). && (3.34)\end{aligned}$$

Nous voyons donc que l'information mutuelle est bien symétrique,

$$I(X;Y) = I(Y;X). \tag{3.35}$$

**Exemple 3.9 (Variables identiques)** *Soit $X$ une variable aléatoire sur un alphabet fini $\mathcal{X}$. L'information mutuelle entre $X$ et elle-même est égale à*

$$I(X;X) = H(X) - \underbrace{H(X|X)}_{=0} = H(X), \tag{3.36}$$

*où nous avons utilisé le fait que $X$ est une fonction de $X$, et donc par le résultat 3.4 que $H(X|X) = 0$.*

**Exemple 3.10 (Variables indépendantes)** *Soient $X$ et $Y$ deux variables indépendantes sur des alphabets finis $\mathcal{X}$ et $\mathcal{Y}$. Leur information mutuelle est*

$$I(X;Y) = H(X) - \underbrace{H(X|Y)}_{=H(X)} = H(X) - H(X) = 0, \tag{3.37}$$

*où nous avons utilisé que $X$ et $Y$ sont indépendants, et donc par le résultat 3.4 que $H(X|Y) = H(X)$.*

**Résultat 3.5 (Valeurs extrêmes de l'information mutuelle)** *Soient $X$ et $Y$ deux variables aléatoires sur des alphabets finis $\mathcal{X}$ et $\mathcal{Y}$. L'information mutuelle entre les deux satisfait*

$$0 \leq I(X;Y) \leq \min\{H(X), H(Y)\}.$$

*En outre,*
- *$I(X;Y) = 0$ si et seulement si $X$ et $Y$ sont indépendants.*
- *$I(X;Y) = H(X)$ si et seulement si $X = f(Y)$ pour une fonction déterministe $f$ quelconque. Idem, $I(X;Y) = H(Y)$ si et seulement si $Y = g(X)$ pour une fonction déterministe $g$ quelconque.*

## 3.3 Définition et théorème de la capacité pour le DMC

Dans cette section, nous allons exhiber la limite fondamentale concernant le nombre de bits d'information qu'on peut transmettre sur un DMC de façon *fiable*, c'est-à-dire, avec une probabilité d'erreur arbitrairement petite.

### 3.3.1 Probabilté d'erreur et définition d'atteignabilité

On considère le même problème de communication comme celui du chapitre 1 (cf. figure 1.1). Mais, contrairement au chapitre 1 consacré au codage, en théorie de l'information les fonctions de codage $f^{(n)}$ et de décodage $g^{(n)}$ sont inclues dans la définition d'un code. De plus on a l'habitude de mettre un exposant $^{(n)}$ indiquant la taille du bloc à laquelle les deux fonctions s'appliquent. Dans ce chapitre, un code sera donc spécifié par le quadruplet $(n, k, f^{(n)}, g^{(n)})$.

Le rendement (en théorie de l'information, on parle plutôt de « taux » -*Rate* en anglais-) $R$ d'un code est égal à

$$R = \frac{k}{n} \qquad \left[\frac{\text{bits d'information}}{\text{utilisation de canal}}\right]. \tag{3.38}$$

On dit qu'il y a une erreur sur la communication si le récepteur se trompe sur au moins un bit d'information :

$$(D_1, \ldots, D_k) \neq (\hat{D}_1, \ldots, \hat{D}_k). \tag{3.39}$$

En théorie de l'information, on considère donc typiquement une *probabilité d'erreur en bloc* et non pas une probabilité d'erreur par bit (ou un taux d'erreur binaire -*Bit Error Rate (BER)*-, en anglais). Par conséquent dans ce chapitre, la probabilité d'erreur est définie comme

$$P_{\text{e}}^{(n)} := \Pr\big\{(D_1, \ldots, D_k) \neq (\hat{D}_1, \ldots, \hat{D}_k)\big\}.$$

Il n'est pas difficile de montrer que la probabilité d'erreur en bloc $P_{\text{e}}^{(n)}$ ne peut pas être inférieure à la probabilité d'erreur par bit. Ce qui implique si on trouve une borne supérieure à $P_{\text{e}}^{(n)}$, elle pourra s'appliquer au BER.

Idéalement, nous aimerions trouver des codes qui atteignent une probabilité d'erreur égale à 0. Malheureusement, ceci est impossible pour presque tous les DMC. A la place, on souhaite seulement trouver un ensemble de codes tel que la probabilité d'erreur peut être rendue aussi faible que l'on souhaite si on choisit le bon code de cet ensemble [3]. Un exemple très simple qui atteint ce but sur le BSC est l'ensemble des codes à répétition (que vous avez vu dans le chapitre 1) avec $k$ fixé et $n$ (multiple de $k$) qui tend vers l'infini. Le problème avec cet ensemble de codes est que pour chaque $n$ le taux est égal à $k/n$ et donc si $n$ tend vers l'infini, le taux $k/n$ tend vers 0. Même si la taille du bloc $n$ tend vers infini, on aimerait bien garder le taux $k/n$ fixe et positif. Plus précisément, on fixe un taux $R > 0$ et on propose un ensemble de codes où $k$ croit avec $n$ comme $k = \lfloor Rn \rfloor$. Le vrai taux du code $k/n$ est alors un peu inférieur à $R$ mais quand $n$ augmente, le taux $k/n$ s'approche asymptotiquement de $R$ et donc l'erreur induite est complètement négligeable. Ceci est résumé dans la définition suivante.

**Définition 3.6** *Un taux $R > 0$ est dit atteignable sur un DMC $(\mathcal{X}, \mathcal{Y}, P_{Y|X})$ s'il existe une suite de codes $\{\big(n, k = \lfloor nR \rfloor, f^{(n)}, g^{(n)}\big)\}_{n=1}^{\infty}$ telle que*

$$P_{\text{e}}^{(n)} \to 0 \qquad \text{lorsque } n \to \infty. \tag{3.40}$$

3. D'un point de vue pratique, rendre aussi faible que possible la probabilité d'erreur n'est pas restrictif car on ne sait pas distinguer en pratique un dispositif offrant une probabilité d'erreur nulle à celui offrant une probabilité d'erreur de $10^{-42}$.

Démontrer pour un DMC quelconque qu'un certain taux $R > 0$ est atteignable est difficile en général et va au-delà du but de ce cours. Comme indiqué dans le prochain exemple, cette tâche par contre devient simple si on se focalise sur un DMC sans bruit.

**Exemple 3.11** *On considère le DMC $(\mathcal{X}, \mathcal{Y}, P_{Y|X})$ décrit par*

$$\mathcal{X} = \mathcal{Y} = \{0, 1, 2\} \qquad \text{et} \qquad P_{Y|X}(y|x) = \mathbb{1}\{x = y\}, \tag{3.41}$$

*et représenté par le diagramme de la figure 3.2 ci-dessous.*

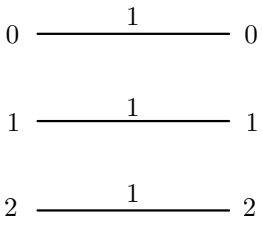



FIGURE 3.2 – Canal sans bruit à 3 entrées.

*Comme le canal est sans bruit, le meilleur choix pour la fonction de décodage est d'inverser $f^{(n)}$, donc $g^{(n)} = (f^{(n)})^{-1}$. Il suffit alors d'attribuer à chaque suite de bits d'information $d_1, \ldots, d_k$ une différente suite de symboles d'entrée $x_1, \ldots, x_n$ pour obtenir une probabilité d'erreur $P_e^{(n)} = 0$. (On note que pour ce cas spécial une probabilité d'erreur égale à 0 est possible. Mais comme déjà mentionné ce n'est pas le cas en général.)*

*Considèrons d'abord une seule utilisation de canal $n = 1$, où l'émetteur n'a que trois valeurs à sa disposition. Il ne peut donc coder qu'un seul bit d'information par mot de code envoyé. Donc $k = 1$ et le taux de transmissions est égale à $k/n = 1$. Un choix naturel pour la fonction de codage par exemple est $f^{(1)}(d_1) = d_1$ pour tout $d_1 \in \{0, 1\}$, mais d'autres choix conduisent au même résultat.*

*Passons maintenant à $n = 2$ où l'émetteur a $3^2 = 9$ valeurs à sa disposition. Il peut donc coder $k = 3$ bits d'information et le taux de transmission est $k/n = 3/2 = 1.5$. Le tableau suivant indique un choix possible pour la fonction de codage :*

| *Donnée $d_1, d_2, d_3$* | $f^{(2)}(d_1, d_2, d_3) = (x_1, x_2)$ |
|---|---|
| 000 | 00 |
| 001 | 01 |
| 010 | 02 |
| 100 | 10 |
| 110 | 11 |
| 011 | 12 |
| 101 | 20 |
| 111 | 21 |

*De la même façon, pour toute valeur de $n \geq 3$ on peut coder $k$ bits avec une probabilité d'erreur $P_e^{(n)} = 0$ si $2^k \leq 3^n$, donc si*

$$\frac{k}{n} \leq \log_2(3). \tag{3.42}$$

*Pour tout taux $R < \log_2(3)$ et pour tout $n$ suffisamment grand, il est donc possible de trouver des fonctions de codage $f^{(n)}$ et décodage $g^{(n)} = (f^{(n)})^{-1}$ tel que la probabilité d'erreur $P_e^{(n)} = 0$. Selon la définition 3.6 tout taux $R < \log_2(3)$ est donc atteignable sur ce DMC. On peut se convaincre, au moins intuitivement, que tout taux $R > \log_2(3)$ n'est pas atteignable.*

### 3.3.2 Capacité et théorème de Shannon

Pour le DMC déterministe à 3 entrées/sorties vu dans l'exemple 3.11, nous pouvions établir par des moyens simples que $\log_2(3)$ réprésente une limite fondamentale pour ce canal dans le sens que tout taux

inférieur à $\log_2(3)$ est atteignable, mais tout taux supérieur à ce seuil ne l'est pas. Ce seuil est généralement nommé la *capacité du canal*. Dans son papier fondamental de 1948, qui à crée le domaine de la théorie de l'information, Shannon a trouvé une formule pour calculer la capacité de tout DMC.

**Définition 3.7** *La capacité $C$ d'un DMC caractérisé par le triple $(\mathcal{X}, \mathcal{Y}, P_{Y|X})$ est*

$$C = \max_{P_X} I(X;Y), \tag{3.43}$$

*où la maximisation se fait sur toutes les lois de probabilité de $X$ et où $Y \sim P_{Y|X}(\cdot|X)$. Donc, dans cette expression, la paire $(X,Y)$ suit la loi de probabilité $P_{XY}(x,y) = P_X(x)P_{Y|X}(y|x)$.*

**Résultat 3.6 (Théorème de Shannon de la capacité)** *Pour un DMC caractérisé par $(\mathcal{X}, \mathcal{Y}, P_{Y|X})$ :*

1. *Tous les débits $0 < R < C$ sont atteignables.*
2. *Aucun débit $R > C$ n'est atteignable,*

Notez que le théorème ne dit rien de ce qui se passe pour les taux $R = C$. Le comportement pour $R = C$ dépend du canal et reste inconnu pour la majorité des canaux à ce jour ! Ce résultat est d'une puissance incroyable et finalement assez contre-intuitif. En effet, on peut rendre fiable n'importe quel lien de transmission à condition d'en payer le prix en débit.

On pourrait permettre à l'émetteur ou au récepteur de fonctionner de manière aléatoire. Il n'est pas difficile de prouver qu'un tel comportement aléatoire n'améliore pas la probabilité d'erreur du système et donc n'augmente pas le taux maximal atteignable sur un DMC. Le théorème de Shannon de la capacité s'applique donc aussi à des systèmes de communication où les émetteurs ou les récepteurs fonctionnent de façon aléatoire.

## 3.4 Expressions analytiques de la capacité de quelques canaux

Dans cette section, nous allons exprimer analytiquement quelques capacités pour quelques canaux. Sachez qu'on ne connaît pas encore de formules analytiques de la capacité pour de très nombreux canaux. A titre d'exemple, la capacité d'un canal composé d'une fibre optique est encore inconnue à ce jour (notamment en raison des effets non-linéaires qui s'y produisent).

Pour calculer la capacité d'un DMC, on part généralement de (3.43). On calcule ainsi d'abord l'information mutuelle $I(X;Y)$ pour une loi d'entrée $P_X(\cdot)$ quelconque, et ensuite on maximise cette information mutuelle obtenue sur $P_X(\cdot)$. Pour calculer l'information mutuelle $I(X;Y)$, on a deux possibilités

- soit on la calcule via
$$I(X;Y) = H(X) - H(X|Y), \tag{3.44}$$
- soit on la calcule via
$$I(X;Y) = H(Y) - H(Y|X). \tag{3.45}$$

Souvent la deuxième méthode est la plus facile, mais ce n'est pas toujours le cas. Nous allons voir que pour le BEC c'est plus facile de calculer l'information mutuelle $I(X;Y)$ en utilisant la première expression.

### 3.4.1 Cas du canal sans bruit

On commence par la classe de DMC la plus simple, les DMCs sans bruit. Soit $m$ un nombre entier positif et

$$\mathcal{X} = \mathcal{Y} = \{1, \ldots, m\} \quad \text{et} \quad P_{Y|X} = \mathbb{1}\{x = y\}. \tag{3.46}$$

Le canal de l'exemple 3.11 est un cas spécial qui correspond à $m = 3$. Le diagramme de canal pour $m = 5$ est donné par la figure 3.3.

Pour calculer la capacité du canal, on note que si on connait l'entrée $X$ alors on connait aussi la sortie du canal $Y$. Donc $Y$ est une fonction déterministe de $X$ et

$$H(Y|X) = 0. \tag{3.47}$$

De plus, on sait que $H(Y) \leq \log_2(m)$ car la loi uniforme maximise l'entropie. On obtient donc

$$C = \max_{P_X} I(X;Y) = \max_{P_X}(H(Y) - H(Y|X)) = \max_{P_X} H(Y) \leq \log_2(m). \tag{3.48}$$

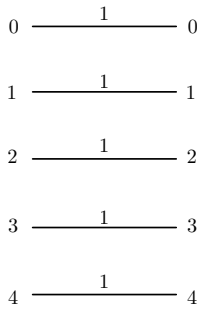



FIGURE 3.3 – Canal sans bruit à 5 entrées.

On note qu'en choisissant pour $P_X$ une loi uniforme sur tout $\mathcal{X}$, la sortie du canal $Y$ suit une loi uniforme aussi et $H(Y) = \log_2(m)$. Par conséquent, l'inégalité en (3.48) devient une égalité et

$$C = \log_2(m). \tag{3.49}$$

En particulier pour $m = 1$ la capacité est nulle. Ceci n'est pas étonnant car les alphabets d'entrée et de sortie $\mathcal{X}$ et $\mathcal{Y}$ ne contiennent qu'une seule valeur et il n'est donc pas possible de communiquer de l'information.

### 3.4.2 Cas du canal binaire symétrique

Nous considérons un canal binaire symétrique introduit dans l'exemple 1.6. Nous commençons par calculer l'information mutuelle $I(X;Y)$ pour une loi $P_X(\cdot)$ quelconque en utilisant (3.45). Il faut donc calculer $H(Y)$ et $H(Y|X)$. Nous commençons par $H(Y|X)$. Nous calculons d'abord $H(Y|X=0)$ et $H(Y|X=1)$ :

$$\begin{aligned} H(Y|X=0) &= -P_{Y|X}(0|0)\log_2(P_{Y|X}(0|0)) - P_{Y|X}(1|0)\log_2(P_{Y|X}(1|0)) \\ &= -(1-p)\log_2(1-p) - p\log_2 p \\ &= H_{\mathrm{b}}(p) \end{aligned} \tag{3.50}$$

et

$$\begin{aligned} H(Y|X=1) &= -P_{Y|X}(0|0)\log_2(P_{Y|X}(0|1)) - P_{Y|X}(1|1)\log_2(P_{Y|X}(1|1)) \\ &= -p\log_2 p - (1-p)\log_2(1-p) \\ &= H_{\mathrm{b}}(p). \end{aligned} \tag{3.51}$$

Donc, indépendamment du choix de la loi d'entrée $P_X(\cdot)$,

$$H(Y|X) = \sum_{x\in\{0,1\}} P_X(x) H_{\mathrm{b}}(p) = H_{\mathrm{b}}(p), \tag{3.52}$$

parce que $\sum_{x\in\{0,1\}} P_X(x) = 1$.

Pour simplifier la suite des calculs, nous définissons $q \triangleq P_X(1)$ et donc $P_X(0) = 1-q$, où $q \in [0,1]$. Par la formule de la loi marginale et par la règle de Bayes :

$$\begin{aligned} P_Y(1) &= P_{XY}(0,1) + P_{XY}(1,1) && (3.53)\\ &= P_X(0)P_{Y|X}(1|0) + P_X(1)P_{Y|X}(1|1) && (3.54)\\ &= (1-q)p + q(1-p), && (3.55) \end{aligned}$$

et

$$P_Y(0) = 1 - P_Y(1) = 1 - \big((1-q)p + q(1-p)\big) = qp + (1-p)(1-q). \tag{3.56}$$

Cela fait que l'entropie de $Y$ est égale à

$$H(Y) = H_{\mathrm{b}}\big((1-q)p + q(1-p)\big). \tag{3.57}$$

Donc, pour $P_X(1) = q \in [0,1]$ quelconque, l'information mutuelle est

$$I(X;Y) = H(Y) - H(Y|X) = H_{\mathrm{b}}\big((1-q)p + (1-p)q\big) - H_{\mathrm{b}}(p). \tag{3.58}$$

La capacité du canal BSC vaut donc

$$\begin{aligned} C_{\mathrm{BSC}(p)} &= \max_{q\in[0,1]} \big[H_{\mathrm{b}}\big((1-q)p+(1-p)q\big) - H_{\mathrm{b}}(p)\big] && (3.59)\\ &= \left(\max_{q\in[0,1]} H_{\mathrm{b}}\big((1-q)p+(1-p)q\big)\right) - H_{\mathrm{b}}(p) && (3.60)\\ &= 1 - H_{\mathrm{b}}(p), && (3.61) \end{aligned}$$

où la deuxième égalité est obtenue parce que $H_{\mathrm{b}}(p)$ ne dépend pas de $q$, et la troisième égalité est obtenue parce que $H_{\mathrm{b}}(t) \leq 1$ avec égalité si $t = 1/2$ et parce que pour $q = 1/2$,

$$(1-q)p + q(1-p) = (1/2)(1-p) + (1/2)p = 1/2. \tag{3.62}$$

**Résultat 3.7** *La capacité d'un BSC(p) est égale à*

$$C_{\mathrm{BSC}(p)} = 1 - H_{\mathrm{b}}(p). \tag{3.63}$$

Nous analysons ci-après quelques cas spéciaux de ce canal. Notons que si $p = 0$ alors la capacité est égale à 1 bit,

$$C_{\mathrm{BSC}(0)} = 1.$$

Ceci est conforme à notre intuition : si $p = 0$ alors la sortie du canal est toujours égale à son entrée $Y = X$ et on se retrouve avec un canal sans bruit par lequel on peut communiquer un bit d'information par utilisation de canal sans jamais faire d'erreur.

Idem, si $p = 1$, alors la capacité est égale à 1 bit,

$$C_{\mathrm{BSC}(1)} = 1.$$

Dans ce cas, on a à nouveau un canal sans bruit, mais le canal inverse tous les bits d'entrée. En inversant simplement tous les bits à la sortie, le récepteur peut récupérer tous les bits d'entrée sans jamais faire d'erreur, comme pour le cas $p = 0$.

Par contre, si $p = 1/2$, chaque sortie de canal est 0 ou 1 avec probabilité $1/2$ indépendamment de la valeur à l'entrée du canal. Comme le récepteur peut toujours construire une suite de Bernoulli(1/2) bits indépendants de l'entrée du canal sans regarder les sorties du canal, ce canal est complètement opaque. Sa capacité est donc nécessairement nulle,

$$C_{\mathrm{BSC}(1/2)} = 0.$$

### 3.4.3 Cas du canal binaire à effacement

Nous considérons le canal binaire à effacement présenté dans l'exemple 1.7. Nous calculons l'information mutuelle $I(X;Y)$ pour une loi d'entrée $P_X(\cdot)$ quelconque en utilisant (3.44). Nous calculons donc les entropies $H(X)$ et $H(X|Y)$, et nous commençons par le premier terme. Pour simplifier la notation dans la suite, nous définissons $q \triangleq P_X(1) = 1 - P_X(0)$. Alors, comme $X$ est une variable Bernoulli($q$),

$$H(X) = H_{\mathrm{b}}(q). \tag{3.64}$$

Nous regardons maintenant le deuxième terme, $H(X|Y)$, en calculant séparément $H(X|Y=0)$, $H(X|Y=1)$, et $H(X|Y=\Delta)$. Notez que si $Y = 0$ alors $X = 0$ avec une probabilité 1 (indépendamment de $P_X(x)$), car le canal n'inverse jamais un bit d'entrée mais peut seulement l'effacer. Donc, $P_{X|Y}(0|0) = 1$ et $P_{X|Y}(1|0) = 0$ et

$$H(X|Y=0) = -0\log_2(0) - 1\log_2(1) = 0. \tag{3.65}$$

Idem, si $Y = 1$, alors $X = 1$ avec une probabilité 1, et donc $P_{X|Y}(1|1) = 1 = 1 - P_{X|Y}(0|1)$ et

$$H(X|Y=1) = 0. \tag{3.66}$$

Donc, l'expression de $H(X|Y)$ se simplifie comme suit

$$H(X|Y) = \sum_{y\in\{0,1,\Delta\}} P_Y(y)H(X|Y=y) = P_Y(\Delta)H(X|Y=\Delta). \tag{3.67}$$

Par la formule de la loi marginale et la règle de Bayes, on a

$$P_Y(\Delta) = \sum_{x\in\{0,1\}} P_{XY}(x,\Delta) = \sum_{x\in\{0,1\}} P_X(x)P_{Y|X}(\Delta|x) = \underbrace{\Big(\sum_{x\in\{0,1\}} P_X(x)\Big)}_{=1}\epsilon = \epsilon. \tag{3.68}$$

Nous calculons maintenant $P_{X|Y}(1|\Delta)$ en fonction du paramètre $q = P_X(x)$. En appliquant deux fois la règle de Bayes et en utilisant (3.68),

$$P_{X|Y}(1|\Delta) = \frac{P_{X,Y}(1,\Delta)}{P_Y(\Delta)} = \frac{P_X(1)P_{Y|X}(\Delta|1)}{P_Y(\Delta)} = \frac{q\epsilon}{\epsilon} = q. \tag{3.69}$$

Donc, si la sortie $Y$ est égale à $\Delta$, alors l'entrée $X$ est Bernoulli($q$) et

$$H(X|Y=\Delta) = H_{\mathrm{b}}(q). \tag{3.70}$$

(3.67) et (3.70) conduisent à

$$H(X|Y) = \epsilon H_{\mathrm{b}}(q). \tag{3.71}$$

Avec (3.64) nous obtenons donc pour la capacité du canal BEC

$$C_{\mathrm{BEC}(\epsilon)} = \max_{q\in[0,1]} \big[H_{\mathrm{b}}(q) - \epsilon H_{\mathrm{b}}(q)\big] = (1-\epsilon)\cdot \max_{q\in[0,1]} H_{\mathrm{b}}(q) = 1-\epsilon, \tag{3.72}$$

où la dernière égalité est obtenue parce que $H_{\mathrm{b}}(p) \leq 1$ avec égalité si $q = 1/2$.

**Résultat 3.8** *La capacité d'un BEC($\epsilon$) est égale à*

$$C_{\mathrm{BEC}(\epsilon)} = 1-\epsilon. \tag{3.73}$$

Durant le déroulement dess calculs nous avons vu que

- Si on observe un 0 ou un 1 à la sortie, alors il n'y a pas de doute sur la valeur d'entrée du canal et donc $H(X|Y=0) = H(X|Y=1) = 0$.
- Si on observe un effacement $\Delta$ à la sortie, alors on n'a rien appris sur l'entrée et donc $H(X|Y=\Delta) = H(X)$.

De plus par la stricte concavité de la fonction $H_{\mathrm{b}}(\cdot)$ et parce que $H_{\mathrm{b}}(1/2) = 1 > 1/2$, il n'est pas difficile de montrer que

$$\epsilon < H_{\mathrm{b}}(\epsilon) \qquad \text{pour tout } \epsilon \in (0, 1/2]. \tag{3.74}$$

ce qui implique, pour $\epsilon \in (0,1/2]$, que

$$\underbrace{1-\epsilon}_{C_{\mathrm{BEC}(\epsilon)}} > \underbrace{1-H_{\mathrm{b}}(\epsilon)}_{C_{\mathrm{BSC}(\epsilon)}}, \tag{3.75}$$

ce qui veut dire que les inversions de bits sont plus durs à gérer que les effacements.

### 3.4.4 Cas du canal gaussien

Tous les résultats antérieurs à cette sous-section ont été obtenus pour des DMCs uniquement. Le résultat 3.6, c'est-à-dire, le théorème de Shannon, s'applique aussi aux canaux sans mémoire *avec alphabets infinis* pour peu qu'on modifie la définition de l'information mutuelle $I(X;Y)$ en conséquence. Nous n'entrerons pas dans plus de détails (cf. le cours de deuxième année), mais présenterons néanmoins des expressions analytiques de la capacité pour deux *canaux gaussien sans mémoire avec contrainte à l'entrée.*

Le canal gaussien est caractérisé par la relation suivante entre l'entrée $X$ et la sortie $Y$

$$Y = X + W,$$

où $W$ est une variable gaussienne réelle de moyenne nulle et variance $N_0/2$. La loi de transition du canal est donc spécifiée par la densité conditionnelle (qui remplace la loi de probabilité $P_{Y|X}(\cdot|\cdot)$ du cas de l'alphabet fini)

$$p_{Y|X}(y|x) = \frac{1}{\sqrt{\pi N_0}} e^{-\frac{(y-x)^2}{N_0}}.$$

Donc, l'alphabet de sortie est nécessairement l'ensemble des nombres réels :

$$\mathcal{Y} = \mathbb{R}.$$

L'alphabet d'entrée dépend des contraintes imposées. Dans un premier cas, nous supposons que seulement deux valeurs sont permises à l'entrée :

$$\mathcal{X} = \{-A, A\}. \tag{3.76}$$

Vous avez vu dans le chapitre 2 que ce cas a un grand intérêt pratique puisqu'il correspond à la « 2-PAM ». La capacité du canal gaussien avec des entrées binaires vérifiant (3.76) vaut

$$C_{\text{2-PAM}} = 1 - \frac{1}{\sqrt{\pi N_0}} \int_{-\infty}^{\infty} e^{-\frac{u^2}{N_0}} \log_2 \left(1 + e^{\frac{4A(A+u)}{N_0}}\right) \mathrm{d}\, u. \tag{3.77}$$

Si on ajoute plus de symboles à l'alphabet d'entrée, la capacité évidemment augmente. En effet l'émetteur peut toujours choisir de ne pas utiliser certains symboles d'entrée que l'on vient d'ajouter et alors on revient au cas précédent. Dans le cas limite où l'alphabet d'entrée contient tous les nombres réels,

$$\mathcal{X} = \mathbb{R} \tag{3.78}$$

la capacité du canal gaussien devient infinie. Mais ce cas n'a pas d'intérêt pratique puisque pour atteindre cette capacité, l'énergie des symboles d'entrées , $\mathbb{E}[X^2]$, doit nécessairement tendre vers infini et qu'un système réel à énergie illimité n'est pas faisable techniquement. De plus, on veut aussi limiter l'énergie du signal émis pour des raisons de sûreté et pour augmenter la durée de vie de la batterie à l'émetteur.

Ainsi, le canal gaussien avec alphabet d'entrée donné par (3.78) est plus intéressant avec une *contrainte d'énergie moyenne à l'entrée*

$$\mathbb{E}[X^2] \leq E_s \tag{3.79}$$

pour une énergie $E_s > 0$ par symbole émis. La capacité de ce canal est

$$C_{\text{Gaussien}} = \frac{1}{2} \log_2 \left(1 + \frac{2E_s}{N_0}\right). \tag{3.80}$$

Notons que si on se limite aux symboles d'entrée $-\sqrt{E_s}$ et $\sqrt{E_s}$, alors l'entrée du canal satisfait toujours la contrainte d'énergie donnée par (3.79). Donc si $A = \sqrt{E_s}$, la capacité du canal gaussien à contrainte d'énergie moyenne, $C_{\text{Gaussien}}$, n'est jamais inférieure à celle du canal gaussien à entrée $2-$PAM, $C_{\text{2-PAM}}$. Numériquement on peut vérifier que pour $A = \sqrt{E_s}$

$$C_{\text{2-PAM}} < C_{\text{Gaussien}}$$

mais que $C_{\text{2-PAM}}$ est une bonne approximation de $C_{\text{Gaussien}}$ quand $E_s/N_0$ est très petit.

On définit le rapport signal-sur-bruit (*Signal-to-Noise Ratio - SNR -*) par

$$\text{SNR} := \frac{2E_s}{N_0}, \tag{3.81}$$

qui est bien le ratio entre la variance du signal utile $X$ et la variance du bruit $N_0/2$. Alors, on obtient la formule célèbre de Shannon pour le canal gaussien [4]

$$C_{\text{Gaussien}} = \frac{1}{2} \log_2 (1 + \text{SNR}). \tag{3.82}$$

La capacité donnée par (3.82) du canal gaussien avec contrainte d'énergie moyenne nécessite d'utiliser un nombre illimité de symboles d'entrée. Techniquement cela n'est pas réalisable, néanmoins nous avons vu au chapitre 2 comment s'en rapprocher.

4. Peut-être avez-vous rencontré cette formule sans le facteur 1/2 devant le terme logarithmique. Dans ce cas, vous avez probablement étudié le canal gaussien *à valeurs complexes* avec un bruit circulaire. Ce canal peut être vu comme deux canaux gaussiens indépendants à valeurs réelles, et sa capacité est donc le double de la capacité du canal gaussien à valeurs réelles.

## 3.5 Bilan

Le théorème de Shannon montre qu'il *existe* des codes qui atteignent la capacité. Néanmoins, la preuve du théoème n'est pas constructive et donc ne permet pas d'en déduire des codes implémentables. C'est seulement dans les années 1990 avec la découverte des Turbo-codes à Télécom Bretagne et des codes LDPC au MIT que des codes implémentables tant au niveau du codeur que du décodeur ont permis de s'approcher très fortement de la capacité du canal gaussien, soit 40 ans après la découverte du théorème de Shannon ! En 2009, les codes polaires ont été découverts à l'université de Bilkent (Turquie) permettant d'atteindre la capacité d'une grande classe de DMC.

Toutes ces constructions de codes et de décodeurs associés sont enseignés à l'école dans le cadre du cycle Master. Concernant la théorie de l'information, en deuxième année, on approfondit les bases de théorie de l'information pour les communications point-à-point, c'est-à-dire, d'un seul émetteur vers un seul récepteur et pour la compression des données. En troisième année, on traite le problème des communications et de la compression de données dans un réseau, domaine qui actuellement est très actif en recherche ainsi que dans l'utilisation directe des résultats de théorie de l'information pour les réseaux dans les systèmes pratiques (tels que la 5G pour la collaboration entre stations de base au niveau des communications et le stockage distribué (*cloud/distributed storage*, en anglais) au niveau de la compression des données.

Nous rappelons ci-dessous les concepts de base et savoir-faire concernant ce chapitre à acquérir durant cette unité d'enseignement.

**Les concepts de base :**

- L'entropie $H(X)$,
- L'entropie conditionnelle $H(X|Y)$,
- L'information mutuelle $I(X;Y)$,
- Canaux discrets sans mémoire (DMC),
- La capacité d'un DMC dont celles des canaux BSC et BEC.

**Les savoir-faire :**

- Connaître les propriétés de $H(X)$, $H(X|Y)$, $H(X,Y)$, et $I(X;Y)$ et savoir calculer ces quantités,
- Connaître la définition de la capacité de canal et savoir calculer la capacité de canaux simples (cf. TD et exercices ci-dessous)..

## 3.6 Exercices

**Exercice 3.1** Soit $U, V, X$ des variables aléatoires sur des alphabets finis $\mathcal{U}$, $\mathcal{V}$, et $\mathcal{X}$. Soit $g$ une fonction $g\colon \mathcal{X} \to \mathcal{X}'$, où $\mathcal{X}'$ est aussi un alphabet fini.

1. Montrer que $H(X) \geq H(g(X))$. À quelle condition cette inégalité est-elle une égalité ? Ces résultats correspondent-ils à votre intuition ?

   *Indication :* Exprimer $I(X; g(X))$ de deux façons différentes et analyser les termes.

2. Soit $W = U + V$. En utilisant la question 1., montrer que $H(W) \leq H(U) + H(V)$. À quelle condition cette inégalité est-elle une égalité ? Donner un exemple de $U$ et $V$ telles que $H(W) = H(U) + H(V)$.

**Exercice 3.2** Soit $X$ une variable aléatoire qui prend ses valeurs dans l'alphabet $\mathcal{X} := \{1, \ldots, m\}$ suivant une loi de probabilité $P_X(i) = p_i$, pour $i = 1, \ldots, m$. On suppose que

$$p_1 > p_2 \geq p_3 \geq \ldots \geq p_m.$$

1. On veut détecter $X$, c'est-à-dire choisir une valeur $\hat{x} \in \{1, \ldots, m\}$ de façon à minimiser la probabilité d'erreur

$$p_{\mathrm{e}}(\hat{x}) := \Pr(X \neq \hat{x}).$$

Quel choix de $\hat{x}$ minimise $p_{\mathrm{e}}(\hat{x})$ ? Quelle est la probabilité d'erreur minimale

$$p_{\mathrm{e}}^{\star} := \min_{\hat{x}\in\{1,\ldots,m\}} p_{\mathrm{e}}(\hat{x}). \tag{3.83}$$

2. Montrer que pour chaque $X$, il existe une autre variable aléatoire $\tilde{X}$ avec seulement $m-1$ valeurs et tel que :
$$H(X) = H_{\mathrm{b}}(p_{\mathrm{e}}^{\star}) + p_{\mathrm{e}}^{\star} \cdot H(\tilde{X}). \tag{3.84}$$
*Indication : choisir $\tilde{X}$ sur l'alphabet $\mathcal{X}\backslash\{\hat{x}^{\star}\}$ avec les mêmes probabilités normalisées que $X$, où $\hat{x}^{\star}$ est la valeur de $\hat{x}$ qui minimise* (3.83).
3. A partir de l'égalité (3.84), prouver que
$$H(X) \leq H_{\mathrm{b}}(p_{\mathrm{e}}^{\star}) + p_{\mathrm{e}}^{\star} \log_2(m-1).$$

Cette dernière inégalité implique que l'entropie $H(X) \to 0$ quand la probabilité d'erreur minimale $p_{\mathrm{e}}^{\star} \to 0$. Une version plus générale de cette inégalité est connue comme *inégalité de Fano* et est fondamentale pour prouver que tout taux $R > C$ n'est pas atteignable.

**Exercice 3.3** *On considère le canal de la figure 3.4, où l'entrée du canal $X$ est binaire et prend des valeurs dans $\{0,1\}$. Le bruit du canal $Z$ est additif et prend ses valeurs équiprobable dans un alphabet fini $\mathcal{Z}$. La sortie du canal est $Y = X + Z$.*

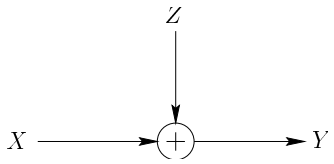
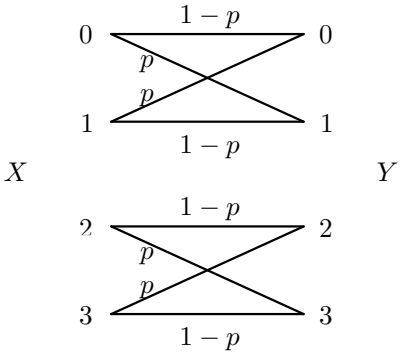



FIGURE 3.4 – Schéma du canal étudié

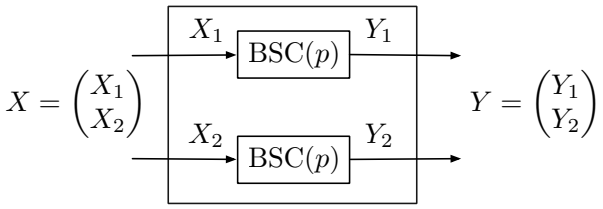
1. *Faire un diagramme du canal et trouver sa capacité si $\mathcal{Z} = \{0,2\}$, c'est-à-dire si $Z$ égale à 0 ou 2 avec probabilité 1/2.*
2. *Trouver la capacité du canal si $\mathcal{Z} = \{0,1,2\}$, c'est-à-dire si $Z$ égale à 0, 1, ou 2 avec probabilité 1/3. Pour cela :*
   *2.1 Faire un diagramme du canal.*
   *2.2 Trouver $I(X;Y)$ si $X$ suit une loi de Bernoulli-$q$ pour $q \in [0,1]$. Exprimer le résultat simplement grâce à la fonction d'entropie binaire $H_{\mathrm{b}}(\cdot)$.*
   *2.3 Calculer la capacité du canal en maximisant l'expression en 2.2 par rapport au paramètre $q \in [0,1]$.*

**Exercice 3.4** Soit $\mathcal{X} = \mathcal{Y} = \{0, 1, 2, 3\}$ et $p \in (0, 1/2)$. On suppose le DMC $(\mathcal{X}, \mathcal{Y}, P_{Y|X})$ décrit par le diagramme de canal suivant :




1. Prouver que la capacité $C$ de ce DMC satisfait $C \geq 2 - H_{\mathrm{b}}(p)$.
2. Trouver la capacité de ce DMC.

**Exercice 3.5** Soit $\mathcal{X} = \mathcal{Y} = \{0, 1\} \times \{0, 1\}$ et $p \in (0, 1/2)$. On suppose le DMC $(\mathcal{X}, \mathcal{Y}, P_{Y|X})$ composé de deux BSC($p$) parallèles comme indiqué dans la figure suivante.




1. Prouver que pour toute loi d'entrée de $(X_1, X_2)$ :

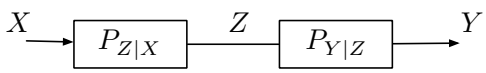
$$I(X_1, X_2; Y_1, Y_2) = H(Y_1, Y_2) - 2H_{\mathrm{b}}(p).$$

2. Prouver que pour toute loi d'entrée de $(X_1, X_2)$ :

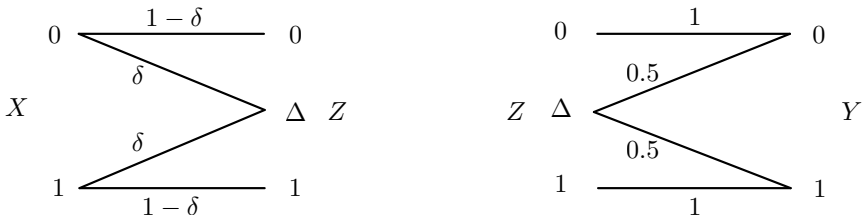
$$I(X_1, X_2; Y_1, Y_2) \leq 2(1 - H_{\mathrm{b}}(p)).$$

3. Trouver la capacité du DMC $P_{Y|X}$.

**Exercice 3.6** On s'intéresse à un canal composé qui enchaîne un BEC ($\mathcal{X} := \{0, 1\},\ \mathcal{Z} := \{0, 1, \Delta\},\ P_{Z|X}$) et un « BEC inversé »($\mathcal{Z},\ \mathcal{Y} := \{0, 1\},\ P_{Y|Z}$) comme suit :




Les diagrammes du BEC et du « BEC inversé » sont présentés dans la figure suivante :




1. Donner le diagramme du DMC composé $P_{Y|X}$.
2. Trouver les capacités du DMC composé $P_{Y|X}$, et des deux DMC $P_{Z|X}$ et $P_{Y|Z}$. Parmi ces trois, quelle capacité est la plus grande ?

3. En utilisant la définition d'un *taux atteignable* (définition 3.6) et le théorème de Shannon de la capacité (théorème 3.6), prouver que la capacité du DMC composé $P_{Y|X}$ ne peut pas dépasser la capacité d'un des deux DMC $P_{Z|X}$ et $P_{Y|Z}$.

   *Indication : le codeur peut décider d'appliquer un traitement aléatoire sur ces symboles codés avant de les envoyer sur un canal. De la même façon, un récepteur peut décider d'appliquer un traitement aléatoire sur ces symboles reçus avant de passer au décodage.*
4. Laquelle des deux capacités est la plus grande : la capacité d'un BSC($\epsilon$) ou la capacité d'un BEC($\epsilon$), où on suppose $0 < \epsilon \leq 1/2$ ?

   *Indication : la question 3. peut servir pour répondre à la question 4.*

# Annexe A

# Quelques preuves relatives au chapitre 2

## A.1 Preuve du résultat 2.1

On a

$$\begin{aligned} S_{xx}(f) &= \lim_{T\to+\infty} \frac{1}{T}\mathbb{E}\left[\left|\int_{t=-T/2}^{T/2} x(t)e^{-i2\pi tf}\ dt\right|^2\right] \\ S_{xx}(f) &= \lim_{T\to+\infty} \frac{1}{T}\int_{t=-T/2}^{T/2}\int_{t'=-T/2}^{T/2} e^{i2\pi(t-t')f}\,\mathbb{E}\left[x(t)x(t')\right]dtdt'. \end{aligned}$$

Faisons un changement de variable $(t,t')$ en $(t,\tau=t'-t)$, ce qui conduit à

$$\begin{aligned} S_{xx}(f) &= \lim_{T\to+\infty}\int_{\tau=-T}^{T} e^{-i2\pi\tau f}\left[\frac{1}{T}\int_{t=-T/2-\tau}^{T/2-\tau}\mathbb{E}\left[x(t)x(t+\tau)\right]dt\right]d\tau. \\ &= \int_{\tau=-\infty}^{\infty} e^{-i2\pi\tau f}\lim_{T\to+\infty}\left[\frac{1}{T}\int_{t=-T/2-\tau}^{T/2-\tau} r_{xx}(t,\tau)dt\right]d\tau. \end{aligned}$$

Comme

$$\lim_{T\to+\infty}\frac{1}{T}\int_{t=-T/2-\tau}^{T/2-\tau} r_{xx}(t,\tau)dt = \lim_{T\to+\infty}\frac{1}{T}\int_{t=-T/2}^{T/2} r_{xx}(t,\tau)dt,\ \forall\tau,$$

on obtient le résultat final.

## A.2 Preuve du résultat 2.2

Soit $x(t)$ le signal défini par l'équation (2.2) avec $N=+\infty$. On a aisément que

$$r_{xx}(t,\tau) = E_s\sum_{\ell} g(t+\tau-\ell T_s)g(t-\ell T_s)$$

Il est facilement de montrer, par simple changement d'indice dans les sommes, que la fonction $t\mapsto r_{xx}(t,\tau)$ est périodique de période $T_s$. Par conséquent, on a

$$r_{xx}^{(0)}(\tau) = \lim_{T\to\infty}\frac{1}{T}\int_{-T/2}^{T/2} r_{xx}(t,\tau)dt = \frac{1}{T_s}\int_{-T_s/2}^{T_s/2} r_{xx}(t,\tau)dt.$$

Donc

$$r_{xx}^{(0)}(\tau) = \frac{E_s}{T_s}\sum_{\ell}\int_{-T_s/2}^{T_s/2} g(t+\tau-\ell T_s)g(t-\ell T_s)dt.$$

En réalisant le changement de variable dans les intégrales $t'=t-nT_s$ et dans la première somme $n''=n'-n$, nous obtenons

$$r_{xx}^{(0)}(\tau) = \frac{E_s}{T_s}\sum_{\ell}\int_{-T_s/2-\ell T_s}^{T_s/2-\ell T_s} g(t'+\tau)g(t')dt'$$

d'où

$$r_{xx}^{(0)}(\tau) = \frac{E_s}{T_s} \int_{-\infty}^{+\infty} g(t'+\tau)g(t')dt'.$$

Par l'égalité de Parseval, nous avons

$$r_{xx}^{(0)}(\tau) = \frac{E_s}{T_s} \int_{-\infty}^{+\infty} |G(f)|^2 e^{2i\pi f\tau} df$$

Par transformée de Fourier inverse, on obtient le résultat voulu.