Codes convolutifs et codes concaténés associés

Charly Poulliat

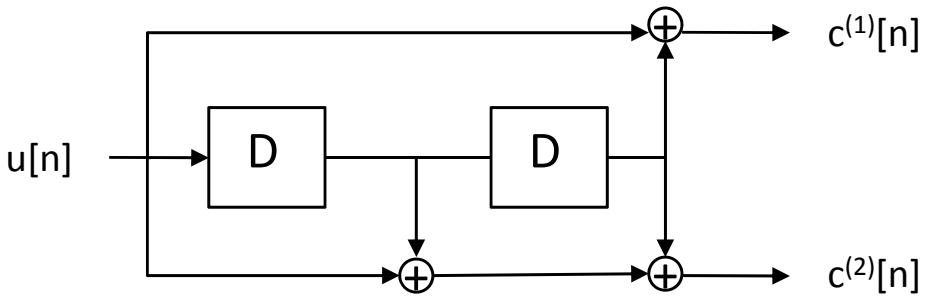
20 octobre 2011

Plan

1. Un exemple : le code $(5, 7)_8$
2. Représentation générale
3. Représentation en treillis
4. Décodage MAP
5. Turbo-codes parallèles
6. Turbo-codes série
7. Analyse EXIT charts
8. Systèmes à décodage itératif avec concaténation en série

Introduction : le code $(5, 7)_8$

Représentation par registre à décalage



- Représentation vectorielle : $g^{(1)} = [101] = [5]_8$ et $g^{(2)} = [111] = [7]_8$
- Représentation polynomiale (opérateur du retard D) : $g^{(1)}(D) = 1 + D^2$ et $g^{(2)}(D) = 1 + D + D^2$
- Rendement : $R = k/n = 1/2$ (k entrées, n sorties).
- *Mémoire* $\nu = 2$, *longueur de contrainte* $l_c = \nu + 1$.

Introduction : le code $(5,7)_8$

Notations

- Relations entrées-sorties en temps : soit $u[n]$ la séquence d'entrée, on a alors comme sorties

$$c^{(i)}[n] = u \circledast g^{(i)}[n],\ \forall i \in \{1,2\}$$

où $\circledast$ représente le produit de convolution sur $GF(2)$.

- Représentation polynomiale : $c^{(i)}(D) = u(D)g^{(i)}(D),\ \forall i \in \{1,2\}$
- Représentation vectorielle polynomiale :

$$\underline{c}(D) = [c^{(1)}(D), c^{(2)}(D)] = u(D)[g^{(1)}(D), g^{(2)}(D)] = u(D)G(D)$$

où $G(D)$ matrice génératrice polynômiale de taille $k \times n$.

Introduction : le code $(5, 7)_8$

Terminaison de codage

- Troncature : pas de terminaison particulière du codage. Pour K bits en entrée, on a émis $N = K/R$ bits codés.
- Fermeture de treillis : On ajoute aux K bits d'information ν bits, dits de fermeture, pour rejoindre l'état 'nul' du registre. Cela entraine une baisse du rendement R. Ici, $R = K/2 * (K + 2)$.
- Fermeture circulaire de treillis : tail-biting

Codes convolutifs

Notations

- Rendement : k entrées et n sorties d'où le rendement $R = k/n$.
- Représentation vectorielle polynomiale : soient
  $\underline{u}(D) = [u^{(1)}(D), u^{(2)}(D), \ldots, u^{(k)}(D)]$,
  $\underline{c}(D) = [c^{(1)}(D), c^{(2)}(D), \ldots, c^{(n)}(D)]$

$$\underline{c}(D) = \sum_{i=1}^{k} u^{(i)}(D)\underline{g}_i(D)$$

  où $\underline{g}_i(D) = [g_i^{(1)}(D), g_i^{(2)}(D), \ldots g_i^{(n)}(D)]$ et $g_i^{(j)}(D)$ représente le transfert entre l'entrée i et la sortie j.

$$\underline{c}(D) = \underline{u}(D)G(D)$$

  où $G(D)$ matrice génératrice polynômiale de taille $k \times n$.
- Matrice de parité : $\underline{c}(D)H^T(D)$ et donc $G(D)H^T(D) = \mathbf{0}$

Codes convolutifs

Notations

- Code récursif : certaines relations entrées sorties peuvent être définies par un code récursif

$$g_i^{(j)}(D) = \frac{b_0 + b_1 D + b_2 D^2 + \ldots + b_m D^m}{1 + a_1 D + a_2 D^2 + \ldots + a_m D^m}$$

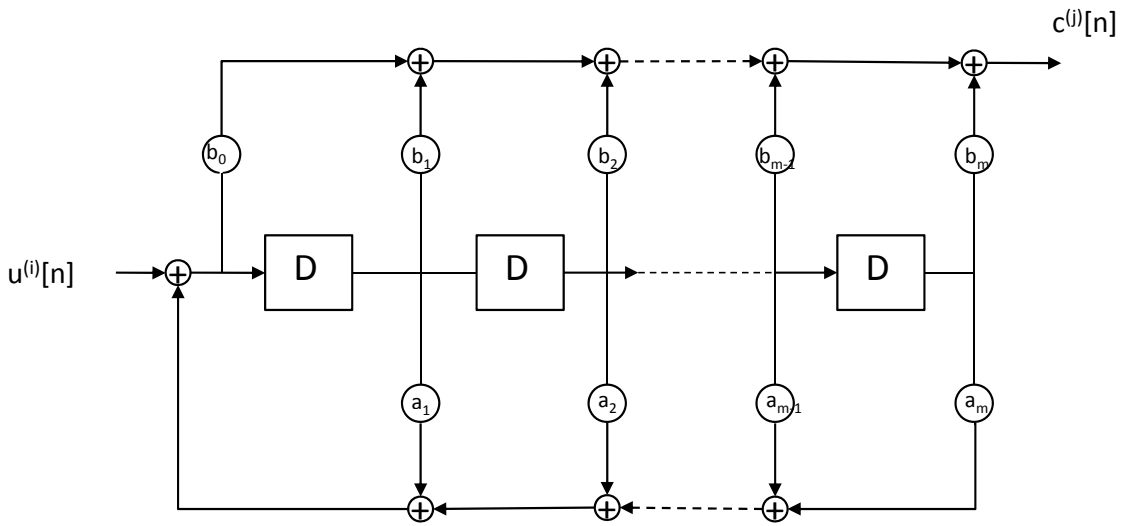


FIGURE: Implémentation d'un transfert récursif par réalisation d'un filtre récursif à réponse impulsionnelle infinie de Type I.

Codes convolutifs
Notations

- Exemple : code récursif systématique $(1, 5/7)_8$ de matrice génératrice

$$G(D) = [1 \; \frac{1 + D^2}{1 + D + D^2}]$$

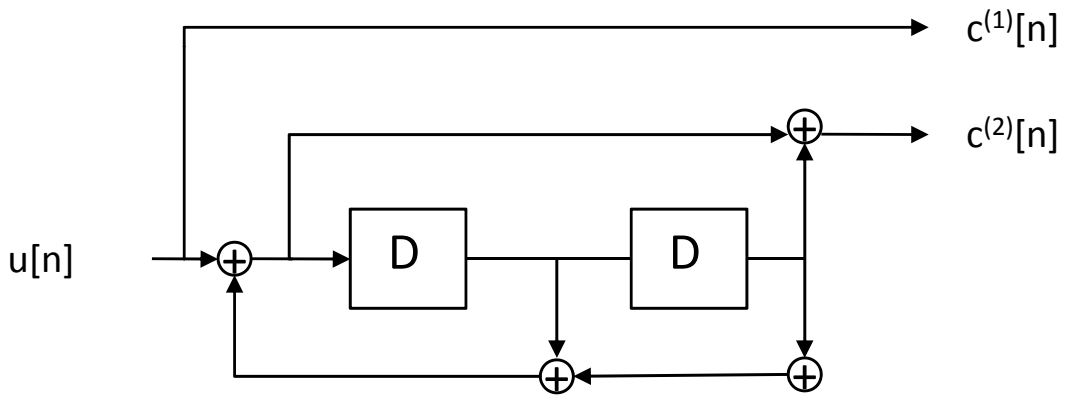
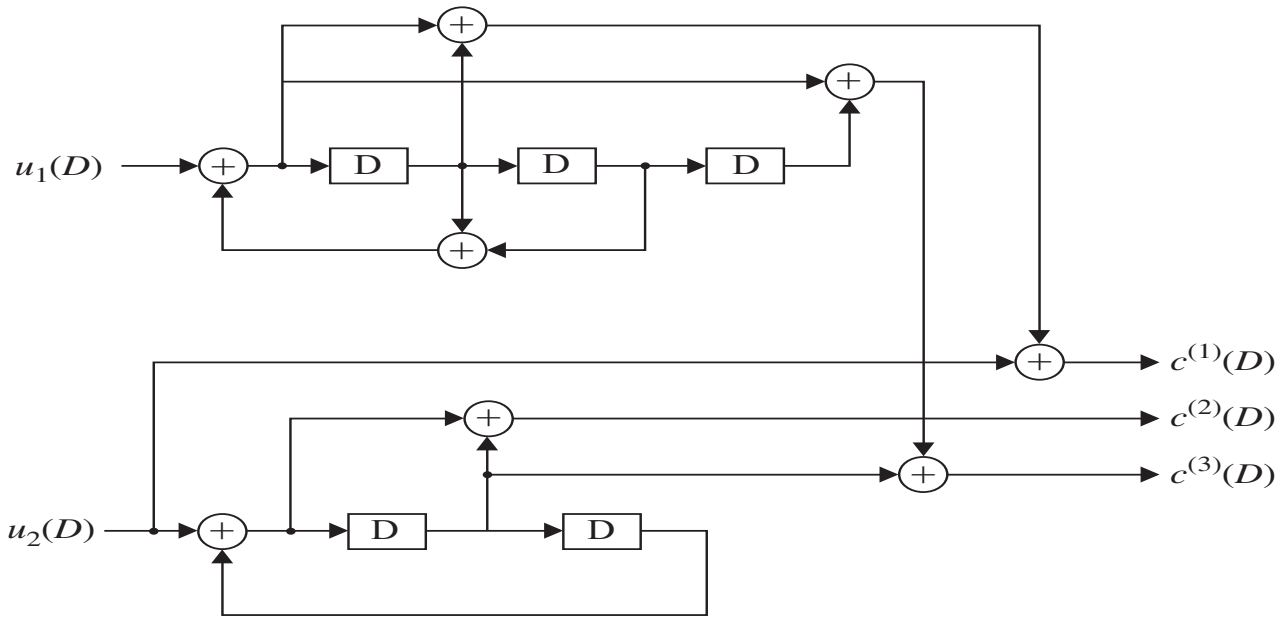


FIGURE: représentation du code récursif (1,5/7)

Codes convolutifs

Notations - exemple pour un code rendement $R = k/n$



$$G(D) = \begin{pmatrix} \frac{1+D}{1+D+D^2} & 0 & 1+D \\ 1 & \frac{1}{1+D} & \frac{D}{1+D^2} \end{pmatrix}$$

$$\underline{u}(D) = [u_1(D), u_2(D)], \underline{c}(D) = [c_1(D), c_2(D), c_3(D)]$$

Codes convolutifs

Modèle d'état et représentation par machine à états finis

- Représentation par machine à états finis : pour un code de type $R = 1/n$,

$$\{u_k\} \rightarrow \boxed{\underline{S}\text{n}} \rightarrow \{c_k\}$$

où $c_k = [c_k^{(1)}, c_k^{(2)}, c_k^{(n)}]$ et $\underline{S}_k$ représente l'état interne du registre à décalage.

- Représentation fonctionnelle associée :
  - Equation d'évolution : passage d'un état à $\underline{S}_{k-1}$ à $\underline{S}_k$.

$$\underline{S}_k = F_1(\underline{S}_{k-1}, u_k)$$

  - Equation d'observation : génération des sorties observables c_k.

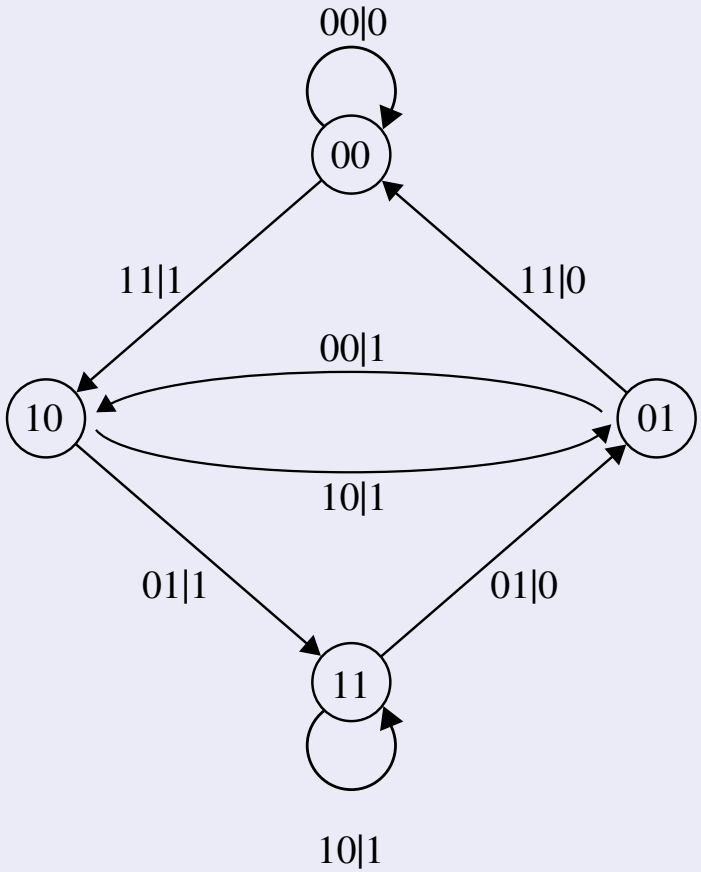
$$c_k = F_2(\underline{S}_{k-1}, u_k)$$

- Exemple : Code $(7, 5)_8$, $\underline{S}_k = [u_k, u_{k-1}]$.

Codes convolutifs

Représentation graphique en Diagramme d'Etat

Code convolutif $(7, 5)_8$



Codes convolutifs

Représentation graphique en treillis

- Definition : représentation graphique du code dans son espace d'état en considérant la dimension temporelle.
  - Noeuds : noeuds du graphe associé à un état $\underline{S}_k$.
  - Transitions : branches entre deux noeuds associées à $F_1(.)$.
  - Etiquettes : informations portées par les branches (u_k, c_k) et données par $F_2(.)$
- Propriétés :
  - Pour un treillis de longueur L, tout chemin du treillis de $\underline{S}_0$ à $\underline{S}_L$ est mot de code obtenu par concaténation des étiquettes c_k du chemin.
  - Le chemin où tous les $\underline{S}_k$ sont l'état **0** (représentation binaire naturelle) est le mot de code nul.
  - *Événement minimal :* chemin qui par de l'état $\underline{S}_k = \mathbf{0}$ et ne revient à cet état que à $\underline{S}_{k+l}$ pour un certain l. On lui associe un poids de Hamming grâce aux étiquettes du chemin.
  - d_{min} est donnée par le poids minimal d'un évênement minimal.

Code convolutif : représentation en treillis

Code $(7, 5)_8$

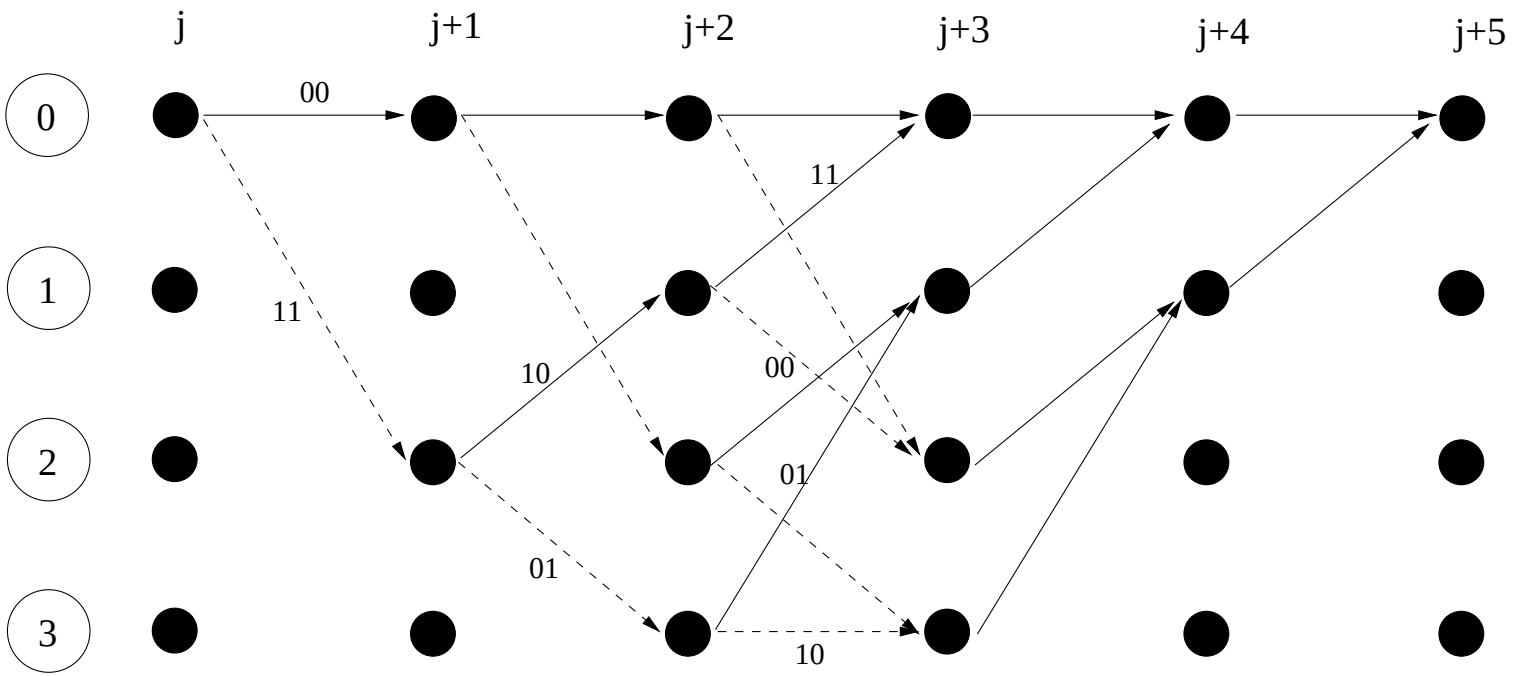


FIGURE: Treillis du code (7,5)

Décodage MAP bit

Notations 1/2

Critère MAP bit

$$\begin{aligned} \hat{u}_n &= \arg \max_{u_n} p(u_n|\mathbf{y}) \\ &= \text{signe}(L(u_n)) \end{aligned} \tag{1}$$

avec

- mapping BPSK : $\{\text{`0'} \leftrightarrow +1, \text{`1'} \leftrightarrow -1\}$.
- $u_n \in \{-1, +1\}, \forall n \in [1, L]$
- LLR MAP (Log Likelyhood Ratio) :

$$L(u_n) = \log \left[\frac{p(u_n = +1|\mathbf{y})}{p(u_n = -1|\mathbf{y})} \right]$$

Décodage MAP bit

Notations 2/2

Critère MAP bit

- Longueur de treillis $L = K + Nf$ (K bits d'info. + Nf bits de fermeture).
- Notations vectorielles :

$$\mathbf{c} = [c_1, c_2, \dots c_L]$$

(mot de code émis) avec $c_k = [c_k^{(1)}, c_k^{(2)}, \dots, c_k^{(n_c)}]$

$$\mathbf{y} = [y_1, y_2, \dots y_L]$$

(mot de code reçu) avec $y_k = [y_k^{(1)}, y_k^{(2)}, \dots, y_k^{(n_c)}]$ et $y_k^{(j)} = c_k^{(j)} + b_k^{(j)}$

$$\mathbf{y}_k^l = [y_k, y_{k+1}, \dots y_{l-1}, y_l]$$

Décodage MAP par bit

Algorithme BCJR 1

LLR MAP revisité

$$\begin{aligned} L(u_n) &= \log\left[\frac{p(u_n = +1|\mathbf{y})}{p(u_n = -1|\mathbf{y})}\right] \\ &= \log\left[\frac{\sum_{\mathcal{S}^+} p(s_{n-1} = s', s_n = s, \mathbf{y})}{\sum_{\mathcal{S}^-} p(s_{n-1} = s', s_n = s, \mathbf{y})}\right] \end{aligned} \tag{2}$$

Notations

- Ensemble des transitions associées à $u_n = +1$:

$$\mathcal{S}^+ = \{(s', s) \text{ où } (s_{n-1} = s') \mapsto (s_n = s) | u_n = +1\}$$

- Ensemble des transitions associées à $u_n = -1$:

$$\mathcal{S}^- = \{(s', s) \text{ où } (s_{n-1} = s') \mapsto (s_n = s) | u_n = -1\}$$

Décodage MAP par bit

Algorithme BCJR 2

Factorisation de $p(s', s, \mathbf{y})$

$$p(s_{n-1} = s', s_n = s, \mathbf{y}) = \alpha_{n-1}(s')\gamma_n(s', s)\beta_n(s) \quad (3)$$

$$(4)$$

$$\alpha_n(s) = p(s_n = s, \mathbf{y}_1^n) \quad (5)$$

$$\beta_n(s) = p(\mathbf{y}_{n+1}^L | s_n = s) \quad (6)$$

$$\gamma_n(s', s) = p(s_n = s, y_n | s_{n-1} = s') \quad (7)$$

Récursions forward-backward

$$\alpha_n(s) = \sum_{s'} \gamma_n(s', s)\alpha_{n-1}(s') \quad (8)$$

$$\beta_{n-1}(s') = \sum_{s} \gamma_n(s', s)\beta_n(s) \quad (9)$$

Décodage MAP par bit

Algorithme BCJR 3

Calcul des probabilités de transitions

$$\begin{aligned} \gamma_n(s', s) &= p(s_n = s, y_n | s_{n-1} = s') && (10) \\ &= p(y_n | s', s).p(s | s') && (11) \\ & && (12) \end{aligned}$$

avec

$$p(s|s') = \begin{cases} 0 & , \text{ si } \{s' \to s\} \text{ non valide} \\ \pi(u_n) & , \text{ sinon} \end{cases}$$

$$\gamma_n(s', s) = p(y_n | c_n(s', s))\pi(u_n)\mathbb{1}_{s' \to s} \quad (13)$$

avec
$\pi(u_n)$ probabilité à priori de u_n
$c_n(s', s)$ les bits associés à l'étiquette $(s' \to s)$

Décodage MAP par bit

Algorithme BCJR 4

Calcul des probabilités de transitions : cas Gaussien

$$y_n^{(m)} = c_n^{(m)} + b_n^{(m)}, b_n^{(m)} \sim \mathcal{N}(0, \sigma_b^2)$$

$$\gamma_n(s', s) \propto \pi(u_n) \exp\left(\frac{\sum_{m=1}^{n_c} |y_n^{(m)} - c_n^{(m)}(s', s)|^2}{2\sigma_b^2}\right) \mathbb{1}_{s' \to s}$$

Initialisation (fermeture treillis)

$$\alpha_0(0) = 1 \quad , \quad \alpha_0(s) = 0 \text{ sinon} \tag{14}$$

$$\beta_L(0) = 1 \quad , \quad \beta_L(s) = 0 \text{ sinon} \tag{15}$$

Décodage MAP par bit

Algorithme BCJR dans domaine logarithmique

Définitions dans le domaine logarithmique

$$\begin{aligned}
\tilde{\alpha}_n(s) &\triangleq \log\left(\alpha_n(s)\right) \\
&= \log \sum_{s'} \exp\left(\tilde{\alpha}_{n-1}(s') + \tilde{\gamma}_n(s', s)\right) \qquad (16)
\end{aligned}$$

$$\begin{aligned}
\tilde{\beta}_{n-1}(s') &\triangleq \log\left(\beta_n(s')\right) \\
&= \log \sum_{s} \exp\left(\tilde{\beta}_n(s) + \tilde{\gamma}_n(s', s)\right) \qquad (17)
\end{aligned}$$

$$\tilde{\gamma}_n(s', s) \triangleq \log\left(\gamma_n(s', s)\right) \qquad (18)$$

$$\begin{aligned}
L(u_n) &= \log\Big(\sum_{\mathcal{S}^+} \exp\left(\tilde{\alpha}_{n-1}(s') + \tilde{\gamma}_n(s', s) + \tilde{\beta}_n(s)\right)\Big) \\
&\quad - \log\Big(\sum_{\mathcal{S}^-} \exp\left(\tilde{\alpha}_{n-1}(s') + \tilde{\gamma}_n(s', s) + \tilde{\beta}_n(s)\right)\Big) \qquad (19)
\end{aligned}$$

Décodage MAP par bit

Algorithme BCJR dans domaine logarithmique

Opérateur $\max^*(x, y)$

$$\max(x, y) = \log\left(\frac{e^x + e^y}{1 + e^{-|x-y|}}\right) \tag{20}$$

$$\begin{aligned} max^*(x, y) &\triangleq \log(e^x + e^y) \\ &= \max(x, y) - \log(1 + e^{-|x-y|}) \end{aligned} \tag{21}$$

$$\begin{aligned} max^*(x, y, z) &\triangleq \log(e^x + e^y + e^z) \\ &= max^*(max^*(x, y), z) \end{aligned} \tag{22}$$

Décodage MAP par bit

Algorithme BCJR dans domaine logarithmique

Log-MAP (log-BCJR)

$$\tilde{\alpha}_n(s) = \max_{s'}{}^*(\tilde{\alpha}_{n-1}(s') + \tilde{\gamma}_n(s', s)) \tag{23}$$

$$\tilde{\beta}_{n-1}(s') = \max_{s}{}^*(\tilde{\beta}_n(s) + \tilde{\gamma}_n(s', s)) \tag{24}$$

$$\begin{aligned} L(u_n) = & \max_{\mathcal{S}^+}{}^*\left(\tilde{\alpha}_{n-1}(s') + \tilde{\gamma}_n(s', s) + \tilde{\beta}_n(s)\right) \\ & -\max_{\mathcal{S}^-}{}^*\left(\tilde{\alpha}_{n-1}(s') + \tilde{\gamma}_n(s', s) + \tilde{\beta}_n(s)\right) \end{aligned} \tag{25}$$

- Implémentation simplifiée par l'opérateur max(.) et utilisation de lookup table.
- stabilité numérique accrue.

Décodage MAP par bit

Algorithme BCJR dans domaine logarithmique

Log-MAP (log-BCJR) : cas Gaussien

$$\tilde{\gamma}_n(s', s) = \log(\gamma_n(s', s) = -\log(4\pi\sigma^2) - \frac{1}{2\sigma^2}\sum_{m=1}^{n_c} |y_n^{(m)} - c_n^{(m)}(s', s)|^2$$

en se réferant au critère MAP final, la constante peut-être supprimée.

Max-Log-MAP

- Remplacer l'opérateur $\max^*(.)$ par l'opérateur $\max(.)$ seul.
- Complexité diminuée, mais perte de performances (raisonnable).
  $\Rightarrow$ décodeur implémenté en pratique

Turbo-codes parallèles

Structure du codeur

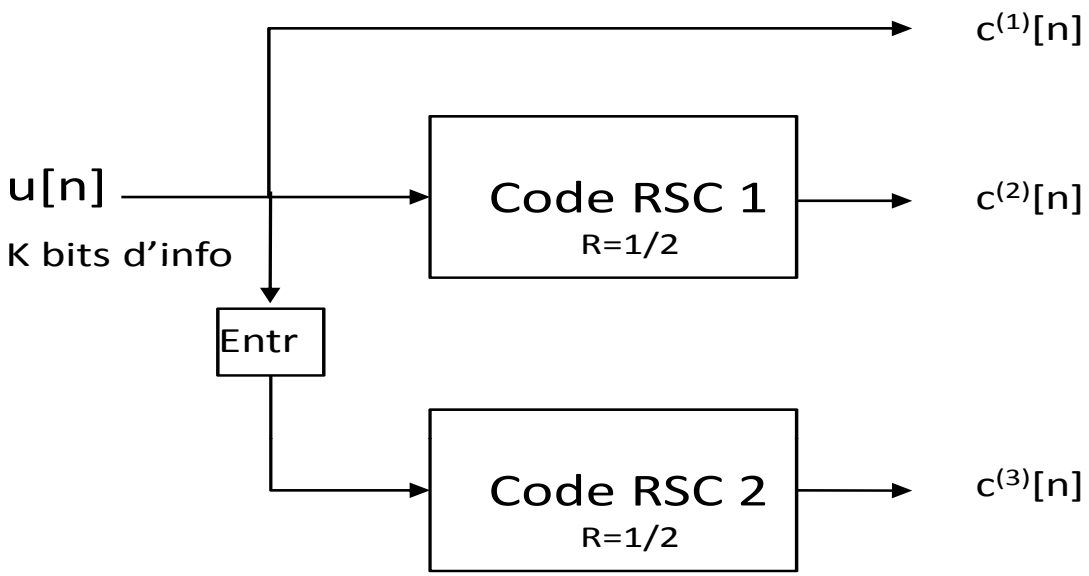


FIGURE: Structure d'un turbo code parallèle : chaque code est un code récursif systématique. Ici $R = 1/2$ pour chaque code constituant.

- K bits d'information sont codés avec le codeur RSC 1,
- les bits d'info. sont entrelacés et codés par le codeur RSC 2,
- Seule la partie redondance est prise en compte sur le codeur RSC 2.

Turbo-codes parallèles

Structure du codeur : exemple UMTS

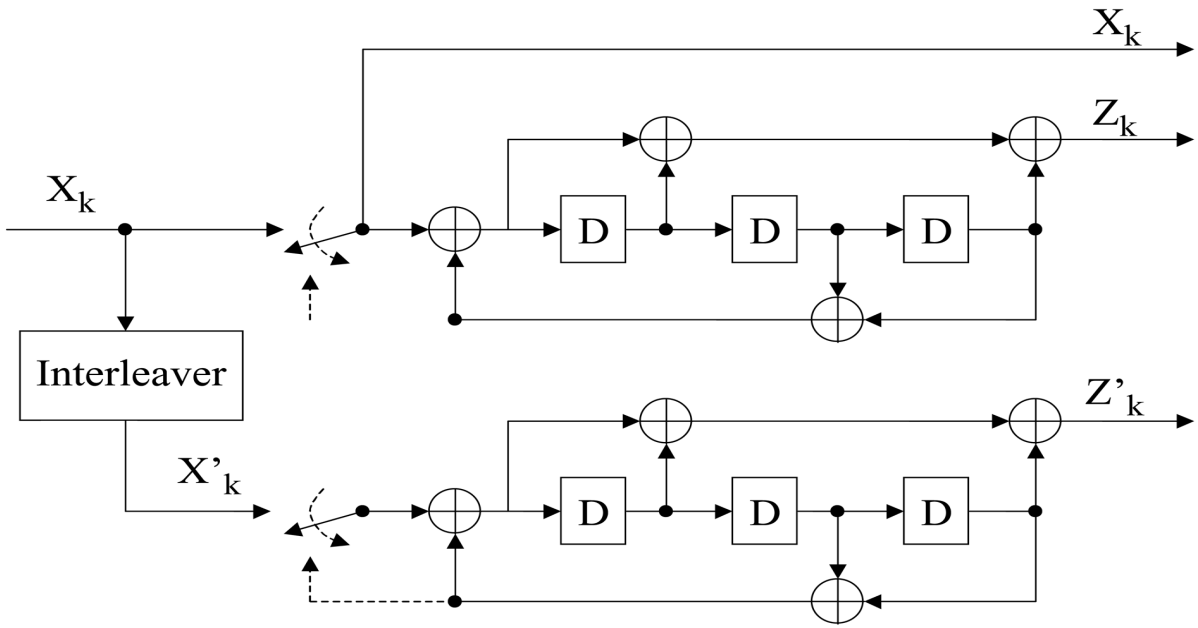


FIGURE: Structure d'un turbo code parallèle pour l'UMTS (3GPP)

Turbo-codes parallèles

Notion d'information extrinsèque 1/3

Cas d'un codeur récursif systématique de rendement $R = 1/n_c$

on supposera que $c_n^{(1)} = u_n$

$$L(u_n) = \log\left[\frac{\sum_{S^+} p(s_{n-1}=s', s_n=s, \mathbf{y})}{\sum_{S^-} p(s_{n-1}=s', s_n=s, \mathbf{y})}\right] \quad (26)$$

$$= L_c(u_n) + L_a(u_n) + L_{ext}(u_n) \quad (27)$$

avec

- Info. canal : $L_c(u_n) = \log\left(\frac{p(y_n^{(1)}|u_n=+1)}{p(y_n^{(1)}|u_n=-1)}\right)$
- Info. a priori : $L_a(u_n) = \log\left(\frac{p(u_n=+1)}{p(u_n=-1)}\right)$
- Info. extrinsèque :

$$L_{ext}(u_n) = \log\left[\frac{\sum_{S^+} \alpha_{n-1}(s') \prod_{k=2}^{n_c} p(y_n^{(k)}|c_n^{(k)}(s',s))\beta_n(s)}{\sum_{S^-} \alpha_{n-1}(s') \prod_{l=2}^{n_c} p(y_n^{(l)}|c_n^{(l)}(s',s))\beta_n(s)}\right]$$

$$= \log\left[\frac{\sum_{S^+} \alpha_{n-1}(s')\gamma_n^e(s',s)\beta_n(s)}{\sum_{S^-} \alpha_{n-1}(s')\gamma_n^e(s',s)\beta_n(s)}\right] \quad (28)$$

Turbo-codes parallèles

Notion d'information extrinsèque 2/3

Info. extrinsèque (domaine logarithmique)

$$\begin{aligned} L_{ext}(u_n) &= \max_{\mathcal{S}^+}{}^{*}\left(\tilde{\alpha}_{n-1}(s') + \tilde{\gamma}_n^e(s', s) + \tilde{\beta}_n(s)\right) \\ &\quad -\max_{\mathcal{S}^-}{}^{*}\left(\tilde{\alpha}_{n-1}(s') + \tilde{\gamma}_n^e(s', s) + \tilde{\beta}_n(s)\right) \end{aligned}$$

$$\tilde{\gamma}_n^e(s', s) = \log\left(\gamma_n^e(s', s)\right)$$

Correspondance entre domaine probabilité vers domaine logarithmique : $L(u_n) \Longleftrightarrow p(u_n)$

$$\begin{aligned} p(u_n) &= \frac{\exp\left(L_a(u_n)/2\right)}{1 + \exp\left(L_a(u_n)\right)} \exp\left(u_n L_a(u_n)/2\right) \\ &= \Lambda_n \exp\left(u_n L_a(u_n)/2\right) \end{aligned} \tag{29}$$

Turbo-codes parallèles

Notion d'information extrinsèque 3/3

Cas Gaussien (domaine logarithmique)

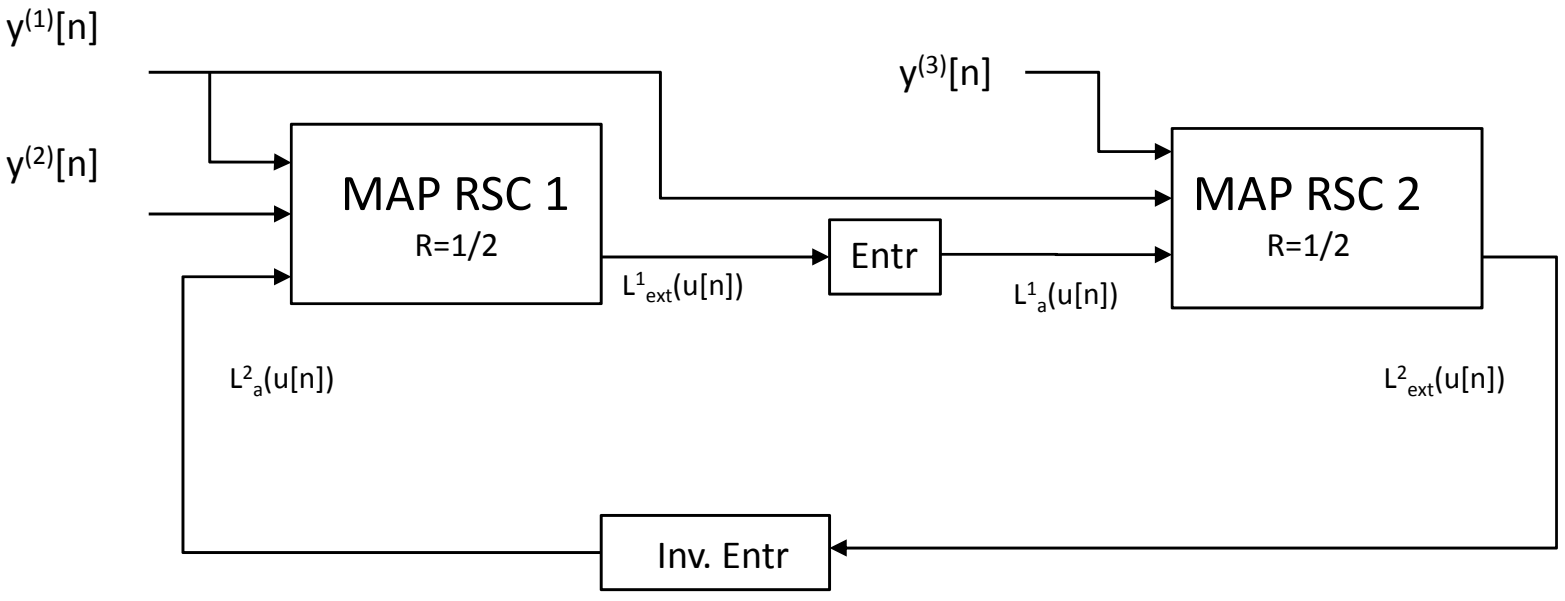
- Info. canal : $L_c(u_n) = \frac{2}{\sigma^2} y_n^{(1)}$,
- Métriques de branches :

$$\begin{aligned}
\tilde{\gamma}_n(s', s) &= u_n \frac{L_a(u_n)}{2} + \frac{1}{2\sigma^2} \sum_{m=1}^{n_c} |y_n^{(m)} - c_n^{(m)}(s', s)|^2 \\
&= u_n \frac{L_a(u_n)}{2} + \sum_{m=1}^{n_c} c_n^{(m)}(s', s) \frac{y_n^{(m)}}{\sigma^2} \\
&= u_n \frac{L_a(u_n)}{2} + \sum_{m=1}^{n_c} c_n^{(m)}(s', s) \frac{L_c(y_n^{(m)})}{2} \qquad (30)
\end{aligned}$$

$$\tilde{\gamma}_n^e(s', s) = \sum_{m=2}^{n_c} c_n^{(m)}(s', s) \frac{L_c(y_n^{(m)})}{2} \qquad (31)$$

Turbo-codes parallèles

Décodeur 1/2



- Les décodeurs MAP des deux codes constituants vont s'échanger une information extrinsèque de manière itérative relative aux bits d'information communs.

Turbo-codes parallèles

Décodeur 2/2

- Critères MAP au deux décodeurs à l'itération (ℓ) :

$$\begin{aligned} L^{(\ell)}_{RSC_1}(u_n) &= \frac{2}{\sigma^2} y^{(1)}_n + L^{(\ell-1)}_{a,1}(u_n) + L^{(\ell)}_{e,1}(u_n) \\ &= \frac{2}{\sigma^2} y^{(1)}_n + L^{(\ell-1)}_{e,2}(u_{\pi^{-1}(n)}) + L^{(\ell)}_{e,1}(u_n) \end{aligned} \quad (32)$$

$$L^{(\ell)}_{RSC_2}(u_{\pi(n)}) = \frac{2}{\sigma^2} \pi(y^{(1)}_{\pi(n)}) + L^{(\ell-1)}_{e,1}(u_{\pi(n)}) + L^{(\ell)}_{e,2}(u_{\pi(n)}) \quad (33)$$

avec $\pi(.)$ et π^{-1} représentent les opérations d'entrelacement et de désentrelacement

Turbo-codes parallèles

Performances

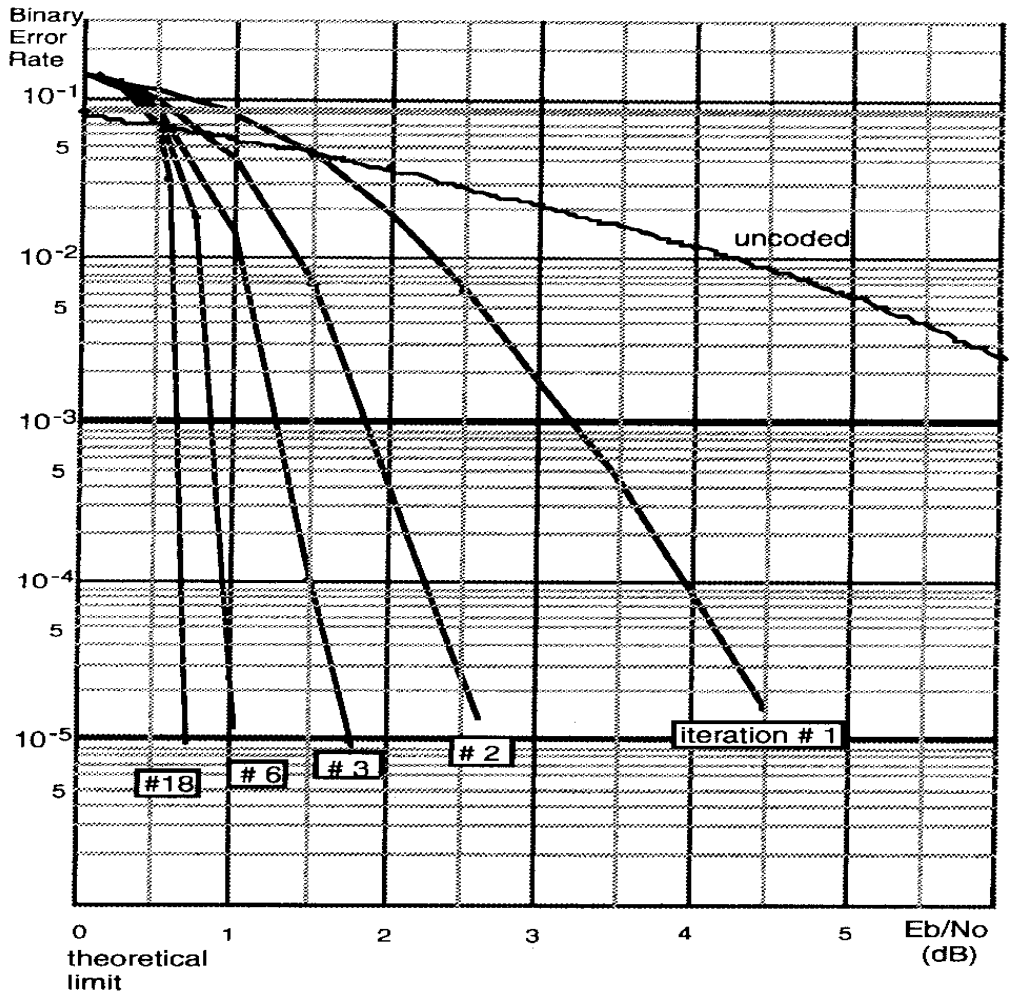
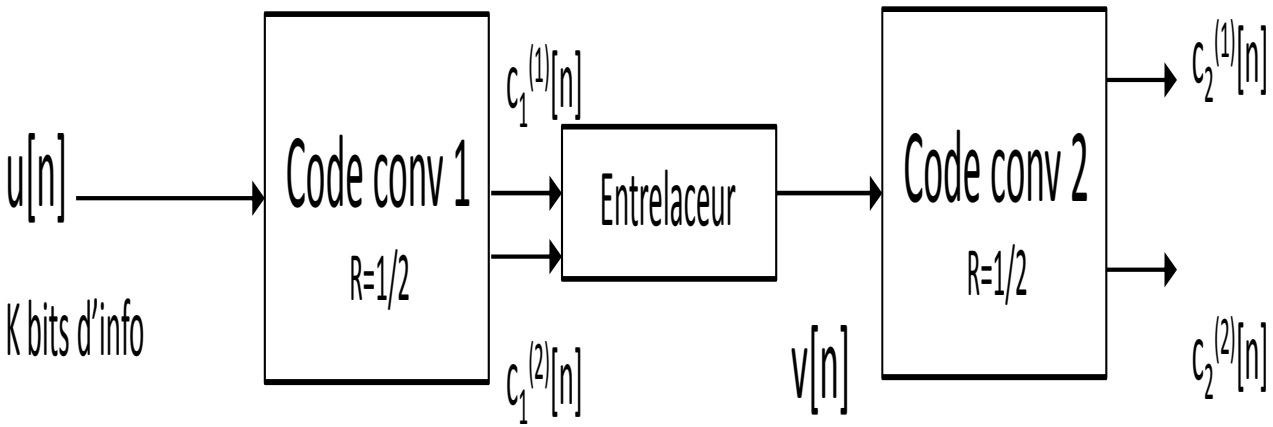


Fig. 9. BER given by iterative decoding ($p = 1, \cdots\cdots 18$) of a rate $R = 1/2$ encoder, memory $\nu = 4$, generators $G_1 = 37, G_2 = 21$, with interleaving 256×256.

Turbo-codes série

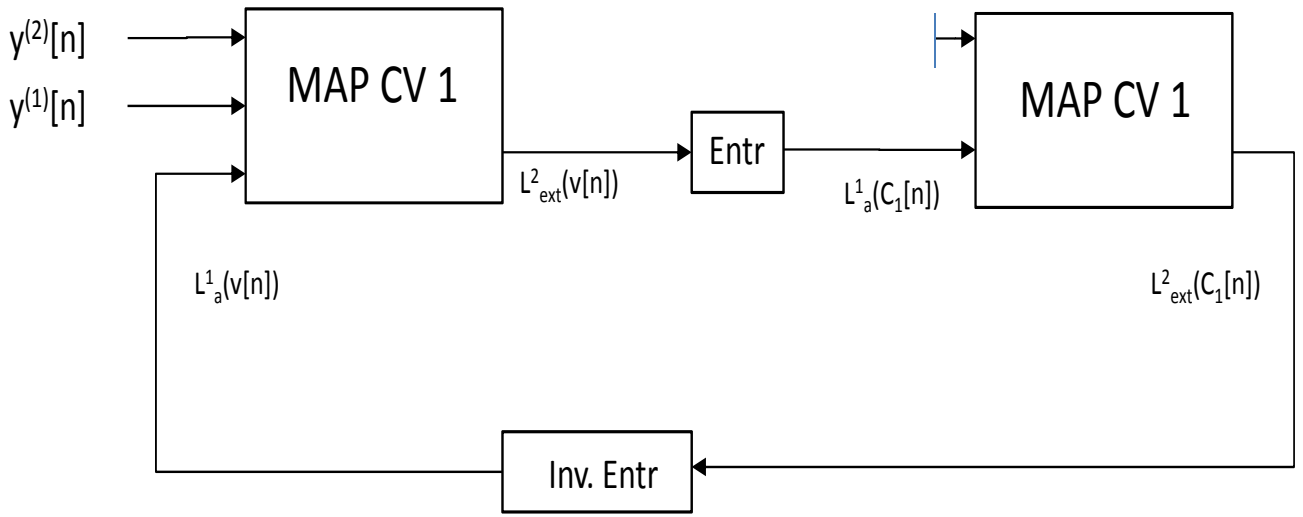
Structure du codeur



- K bits d'information sont codés avec le codeur 1 (code externe),
- les bits codés sont alors entrelacés et puis codés par le codeur 2 (code interne),
- les deux décodeurs associés ne fonctionneront pas forcément de la même manière que pour le cas parallèle,
- seul le codeur 2 doit être récursif pour avoir gain d'entrelacement

Turbo-codes série

Structure du décodeur 1/2



Décodeur code 1

- Idem au décodage d'un bloc du cas parallèle.

Turbo-codes série

Structure du décodeur 2/2

Décodeur code 2

- Critère MAP modifié :

$$
\begin{aligned}
L_1(c_n^{(m)}) &= \max_{\mathcal{C}^+}{}^{*}\left(\tilde{\alpha}_{n-1}(s') + \tilde{\gamma}_n(s', s) + \tilde{\beta}_n(s)\right) \\
&\quad -\max_{\mathcal{C}^-}{}^{*}\left(\tilde{\alpha}_{n-1}(s') + \tilde{\gamma}_n(s', s) + \tilde{\beta}_n(s)\right) \qquad (34) \\
&= L_2^e(c_n^{(m)}) + L_1^e(c_n^{(m)}) \qquad (35) \\
\tilde{\gamma}_n(s', s) &= \sum_{m=1}^{n_c} c_n^{(m)}(s', s)\frac{L_2^e(c_n^{(m)}(s', s))}{2} \qquad (36)
\end{aligned}
$$

avec

$$\mathcal{C}^+ = \{(s', s) \text{ où } (s_{n-1} = s') \mapsto (s_n = s) | c_n^{(m)} = +1\}$$

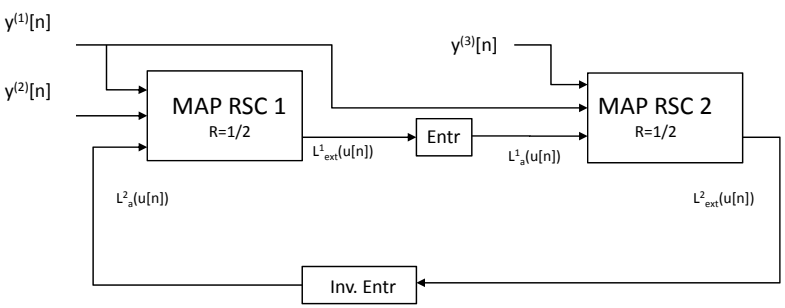
$$\mathcal{C}^- = \{(s', s) \text{ où } (s_{n-1} = s') \mapsto (s_n = s) | c_n^{(m)} = -1\}$$

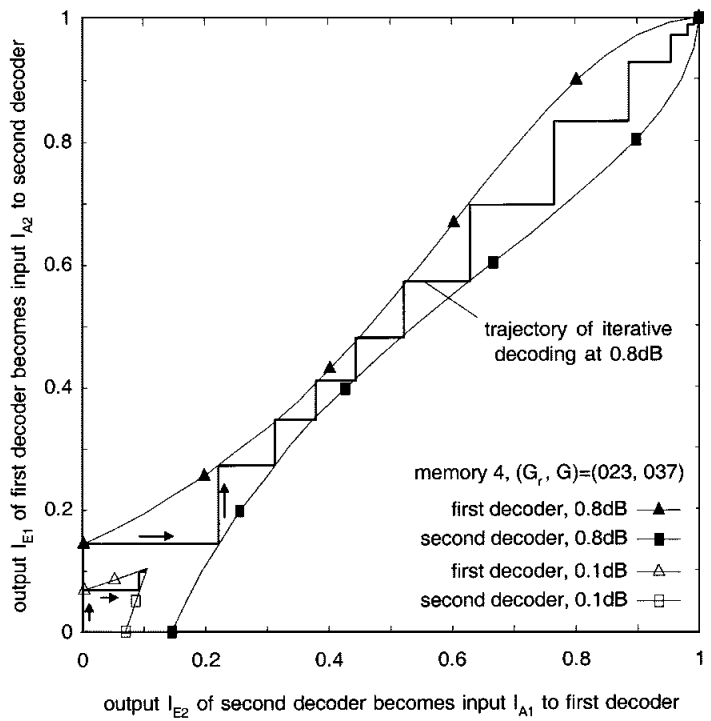
Analyse EXIT charts

Présentation générale

Motivations

- Analyser le comportement d'un turbo-récepteur pour pouvoir prédire les performances en fonction du rapport signal à bruit.





Analyse EXIT charts

Présentation générale

Idée :

- Pouvoir analyser le comportement entrée-sortie de chaque bloc SISO indépendamment : on cherche à déterminer le comportement moyen en sortie par rapport au comportement moyen en entrée suivant une figure de mérite donnée,
- Ce comportement en entrée est fonction de l'info. extrinsèque entrée et des probabilités de transitions du canal : le canal est connu mais nécessité de modéliser le compotement statistique des informations extrinsèques,
- La figure de mérite en entrée sera associée au info. a priori entrantes (extrinsèques des autres blocs),
- de même, la figure de mérite sortante sera associée aux info extrinsèques fournies par le bloc SISO,
- pour simplifier l'analyse, utiliser une mesure de performance mono-domensionnelle (scalaire) pour caractériser le processus de décodage itératif

Analyse EXIT charts

Présentation générale

EXtrinsic Information Transfer Charts :

- Information mutuelle entre un log-rapport de vraisemblance L et le bit émis C (considérés comme variables aléatoires) :

$$I(L;C) = \frac{1}{2}\sum_{c=\pm 1}\int_{\mathbb{R}} f(l|c)\log_2\left(\frac{2f(l|c)}{f(l|c=+1)+f(l|c=-1)}\right)dl$$

- Hypothèses :
  - Modulation BPSK, entrelacement parfait,
  - C variable binaire de loi uniforme,
  - Symétrie de la densité : $f(l|C=-1) = f(-l|C=+1)$,
  - Consistance de la densité (symétrie exponentielle) : $f(-l|C=c) = f(l|C=c)e^{-c.l}$

$$\Downarrow$$

$$\begin{aligned} I(L;C) &= 1 - \mathbb{E}_{L|C=+1}(\log_2(1+e^{-l})) \\ &= 1 - \int_{\mathbb{R}} f(l|c=+1)\log_2(1+e^{-l})dv \end{aligned}$$

Analyse EXIT charts

Information mutuelle d'une densité consistante : estimation.

Estimation

- Utilisation du mot de code nul :

$$I(L; C) \approx 1 - \frac{1}{N} \sum_n \log_2 \left(1 + e^{-l_n}\right)$$

- Mot de codes indifférents et connus :

$$I(L; C) \approx 1 - \frac{1}{N} \sum_n \log_2 \left(1 + e^{-c_n l_n}\right)$$

- Mot de codes indifférents et inconnus :

$$I(L; C) \approx 1 - \frac{1}{N} \sum_n \mathcal{H}_b\left(\frac{e^{+|l_n|/2}}{e^{+|l_n|/2} + e^{-|l_n|/2}}\right)$$

$$\mathcal{H}_b(p) = -p \log_2 (p) - (1 - p) \log_2 (1 - p)$$

Analyse EXIT charts

Information a priori et extrinsèque

- Modèle Gaussien des messages a priori :

$$I = m.c + b,\ b \sim \mathcal{N}(0, \sigma^2 = 2m),\ c = \pm 1$$

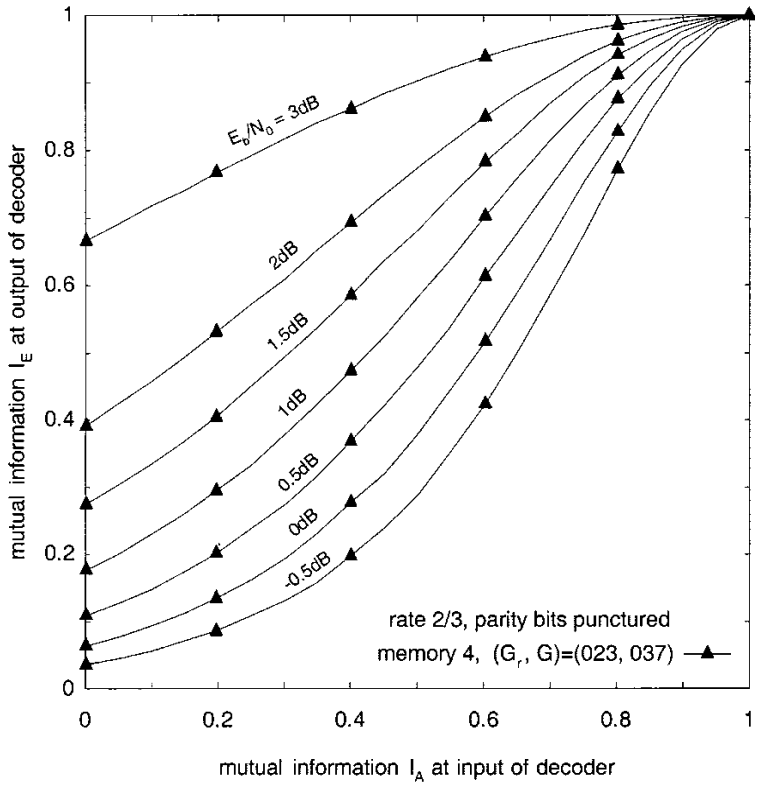
- Information mutuelle associée :

$$I(L; C) = 1 - \frac{1}{\sqrt{2\pi\sigma^2}} \int_{\mathbb{R}} \exp\left(-\frac{(I - \sigma^2/2)^2}{2\sigma^2}\right) \log_2\left(1 + e^{-I}\right) dI$$

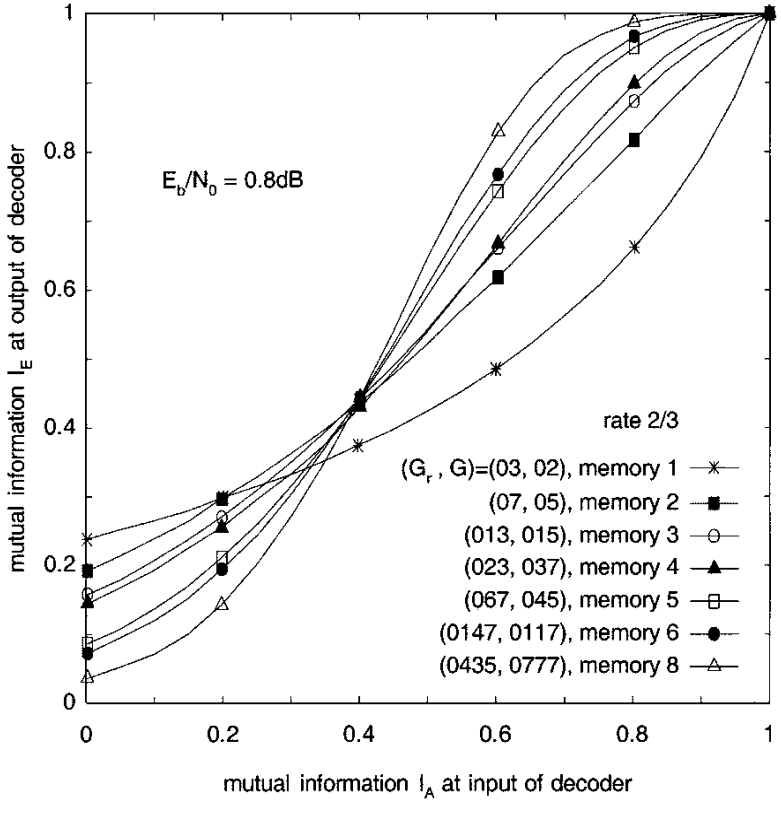
- Information mutuelle extrinsèque : Elle est évaluée **sans approximation Gaussienne** à l'aide des histogrammes des densités en sortie ou à l'aide des estimateurs précédents.

Analyse EXIT charts

Concaténation parallèle 1/3



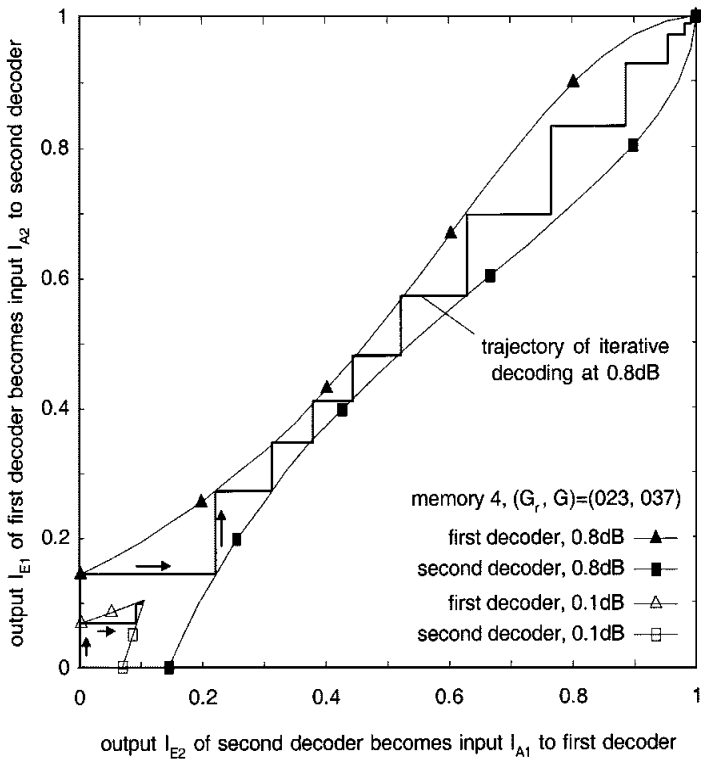
Influence de E_b/N_0

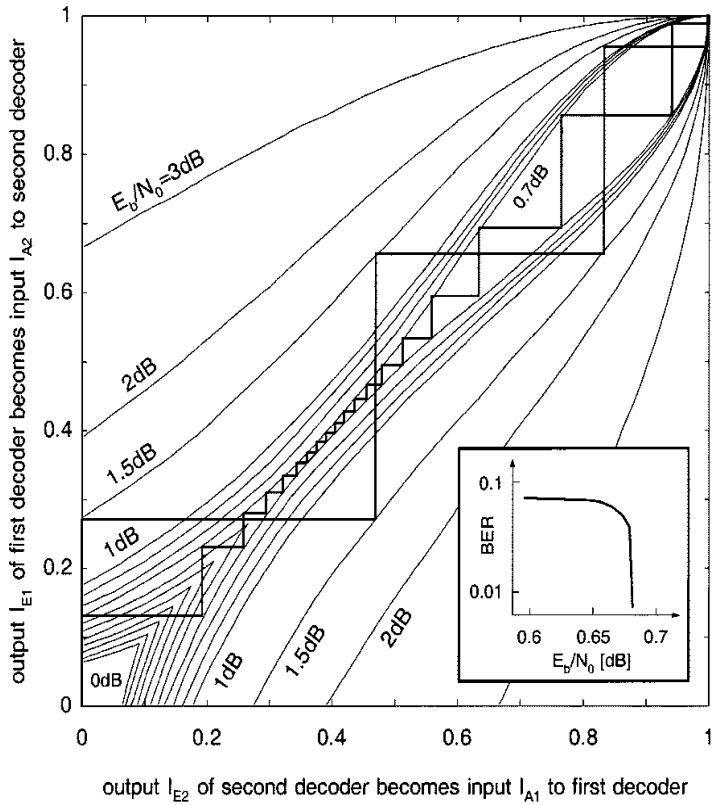


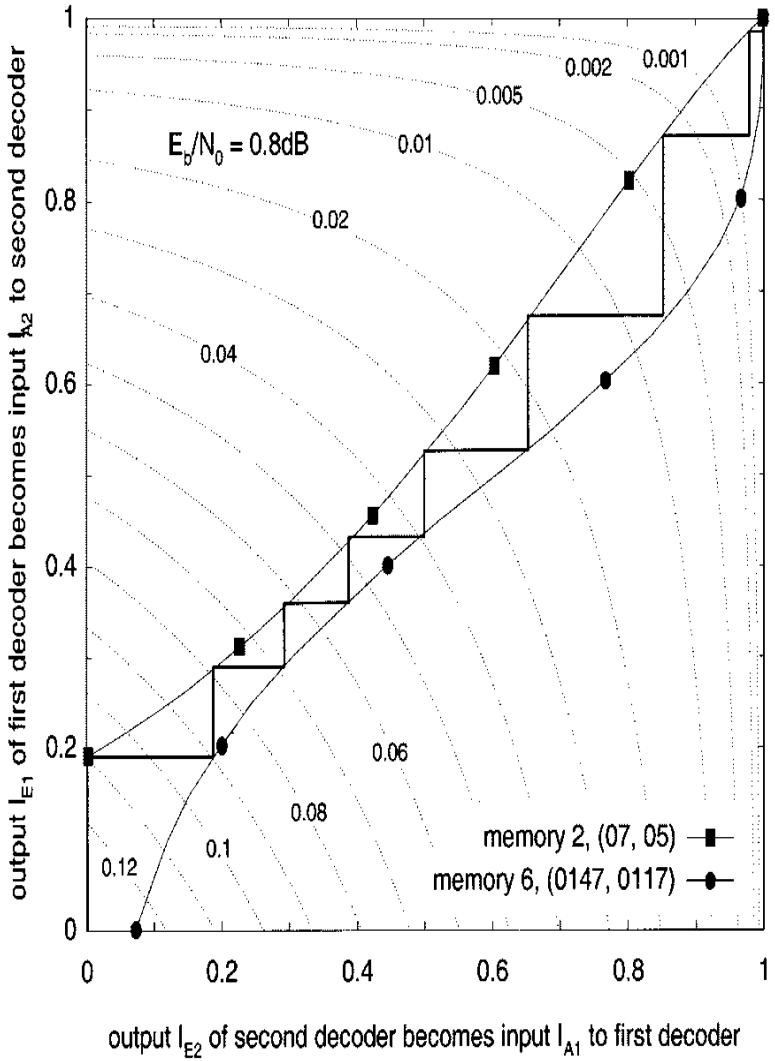
influence de la mémoire

Analyse EXIT charts

Concaténation parallèle 2/3



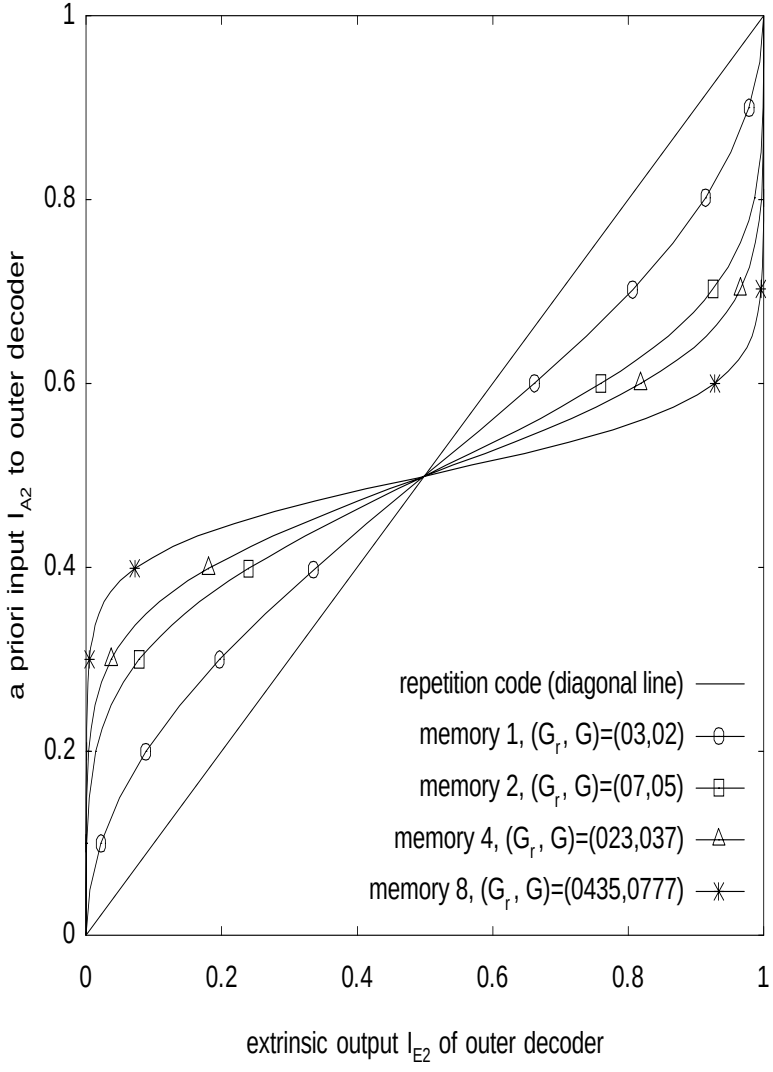


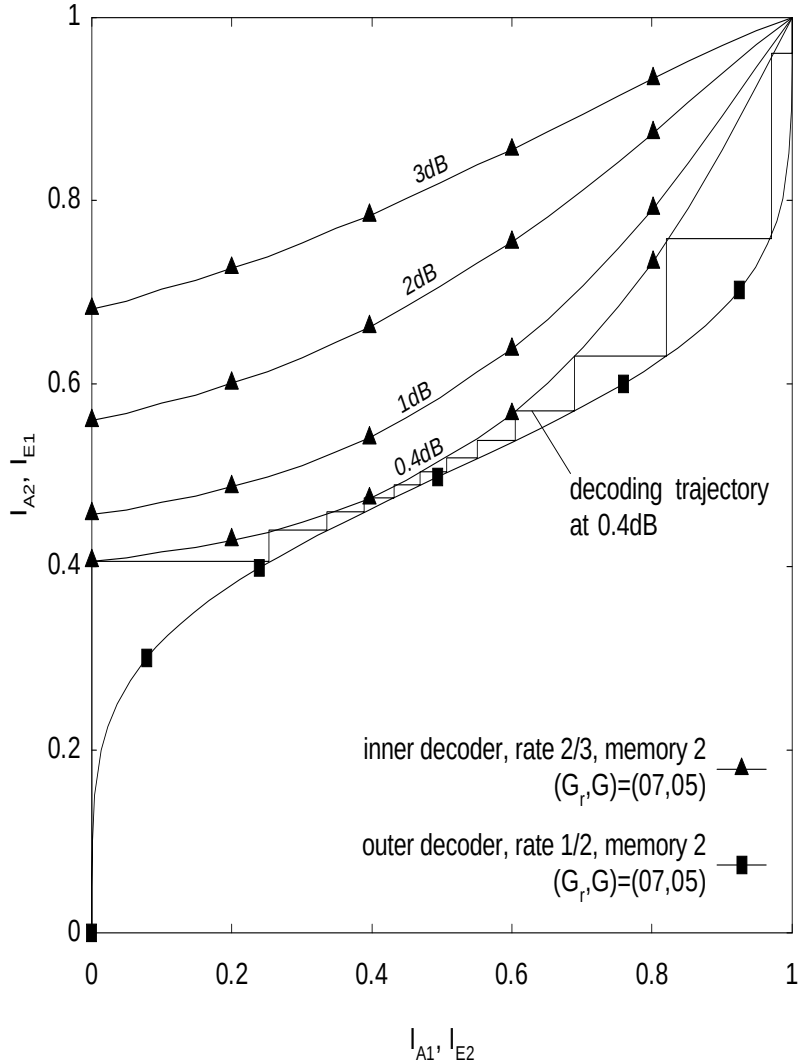


$$P_b \approx erfc\left(\frac{1}{2\sqrt{2}}\sqrt{8R\frac{E_b}{N_0} + J^{-1}(I_A)^2 + J^{-1}(I_E)^2}\right)$$

Analyse EXIT charts

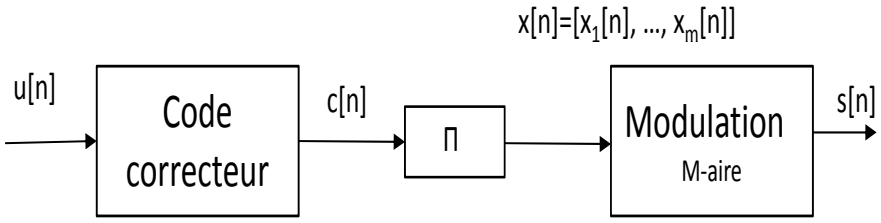
Concaténation série





Analyse EXIT charts

Modulations codées à bits entrelacés



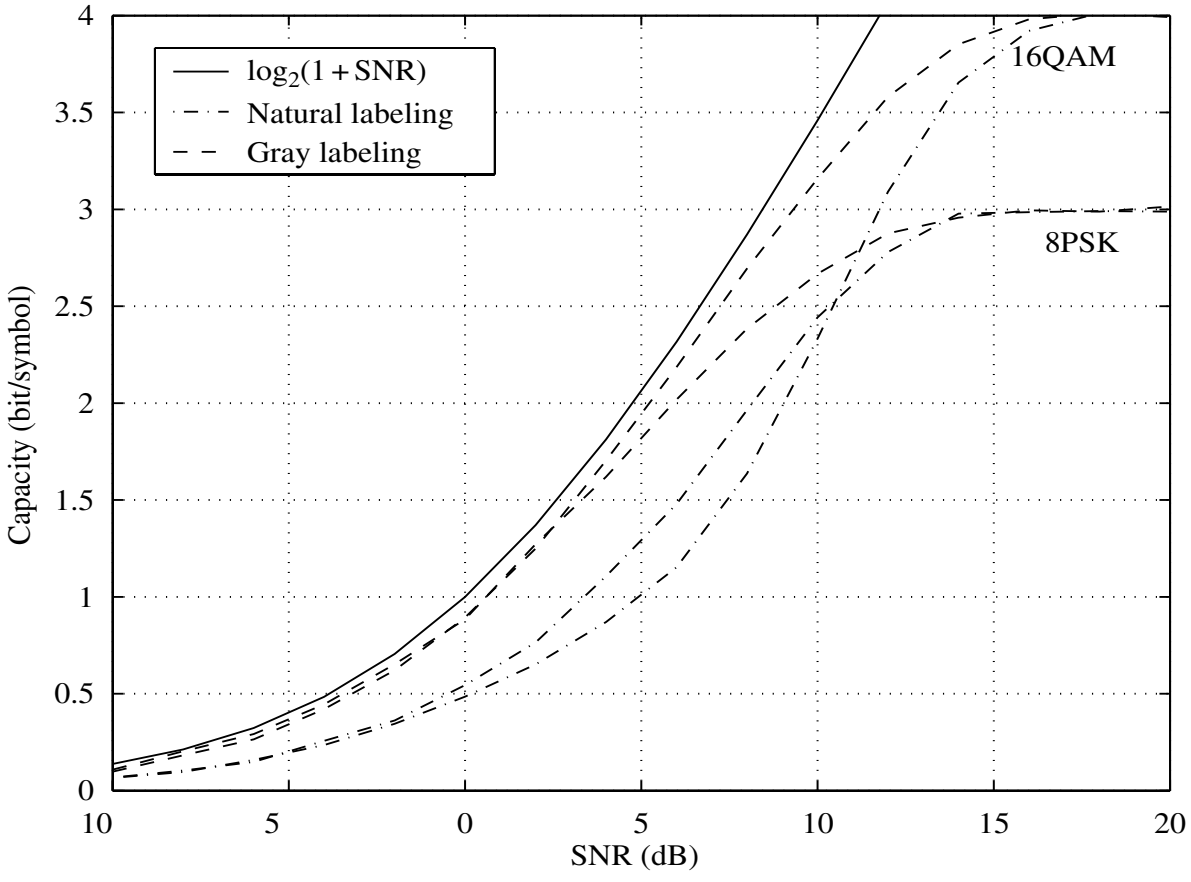
Bit-Interleaved Coded Modulation

- système de transmission à haute efficacité spectrale : constellation M-aire $\mathcal{S}$ avec $M = 2^m$.
- Capacité atteignable dépend du mapping utilisé :

$$C = m - \frac{1}{2} \sum_{k=0}^{m-1} \sum_{c=0}^{1} \mathbb{E}\left(\log_2 \left(\frac{\sum_{s_i \in \mathcal{S}} p(y|s_i)}{\sum_{s_j \in \mathcal{S}_c^k} p(y|s_j)} \right) \right)$$

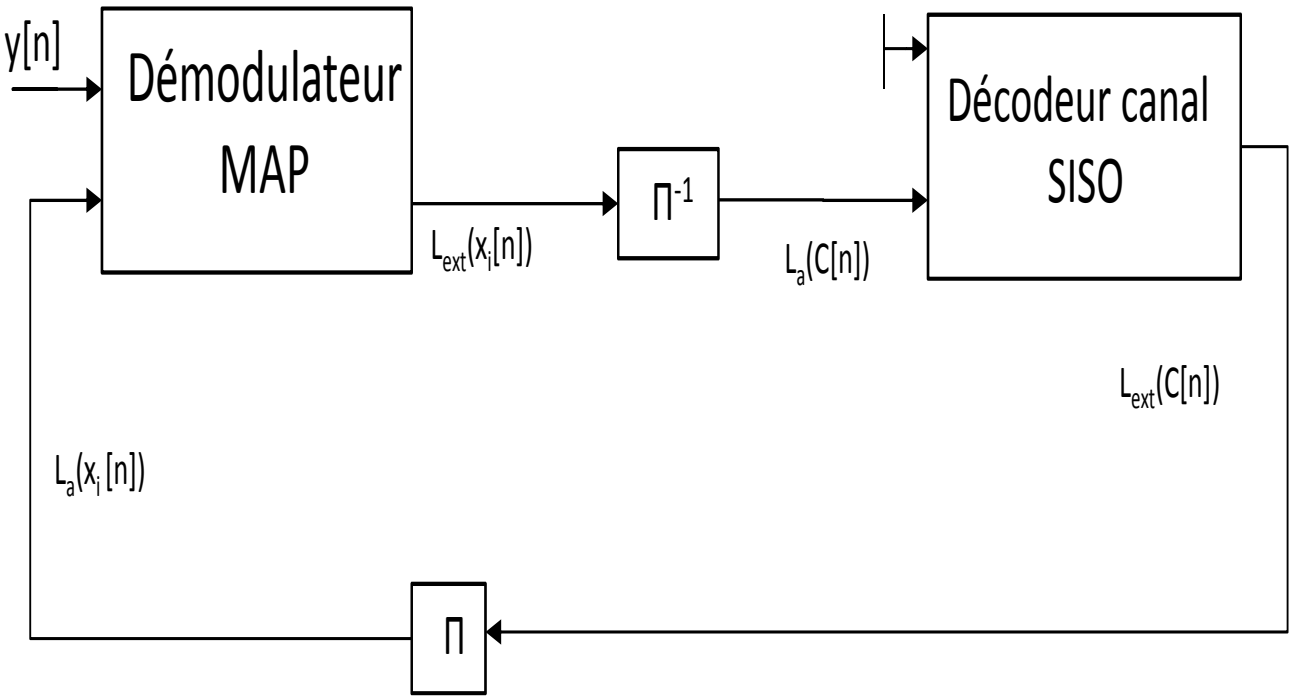
Analyse EXIT charts

Modulations codées à bits entrelacés



Analyse EXIT charts

Démodulation et décodage itératifs



Analyse EXIT charts

Démodulateur MAP

Cas générale

- Les vecteurs binaires $x[n] = [x_1[n] \cdots x_m[n]]$ sont "mappés" sur des symboles $s[n] \in \mathcal{S}$.
- Log-rapport de vraisemblance bit :

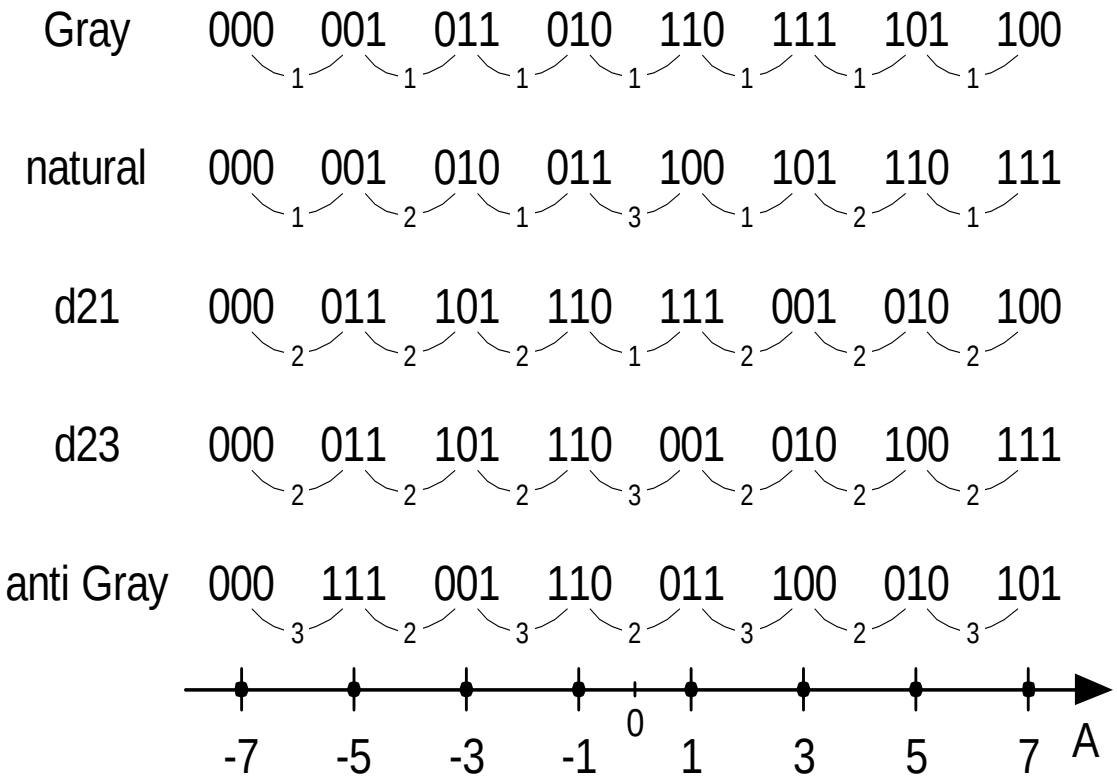
$$\begin{aligned} L(x_i[n]) &= \log\left(\frac{\sum_{s[n]\in\mathcal{S}_0^i} \exp\left(-\frac{\|y[n]-s[n]\|^2}{2\sigma^2}\right)\prod_m \pi(x_m[n])}{\sum_{s[n]\in\mathcal{S}_1^i} \exp\left(-\frac{\|y[n]-s[n]\|^2}{2\sigma^2}\right)\prod_m \pi(x_m[n])}\right) \\ &= L_a(x_i[n]) + L_e(x_i[n]) \end{aligned}$$

- Log-rapport de vraisemblance extrinsèque bit :

$$L(x_i[n]) = \log\left(\frac{\sum_{s[n]\in\mathcal{S}_0^i} \exp\left(-\frac{\|y[n]-s[n]\|^2}{2\sigma^2}\right)\prod_{m;m\neq i} \pi(x_m[n])}{\sum_{s[n]\in\mathcal{S}_1^i} \exp\left(-\frac{\|y[n]-s[n]\|^2}{2\sigma^2}\right)\prod_{m;m\neq i} \pi(x_m[n])}\right) \quad (37)$$

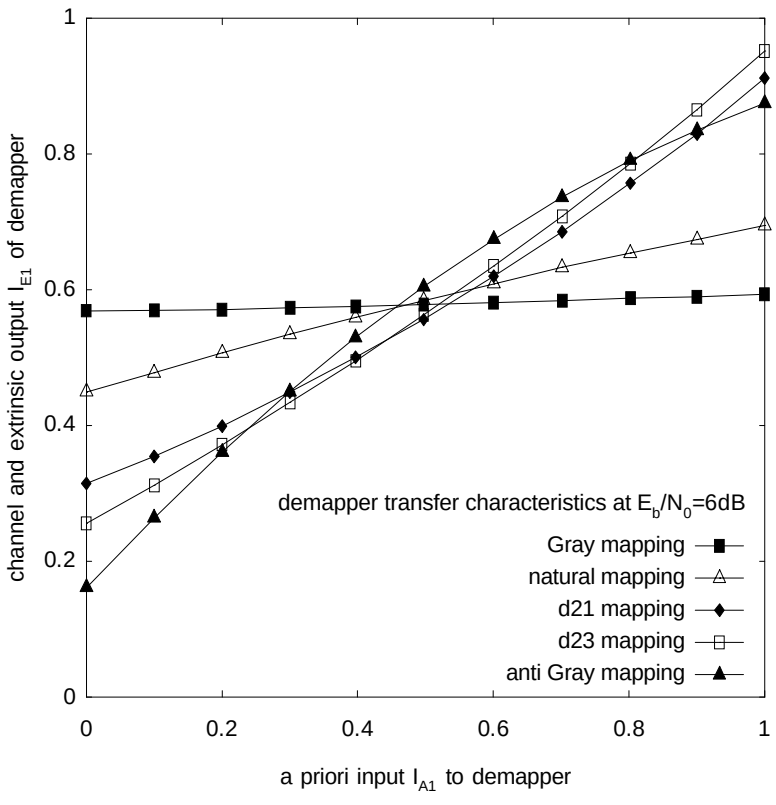
Analyse EXIT charts

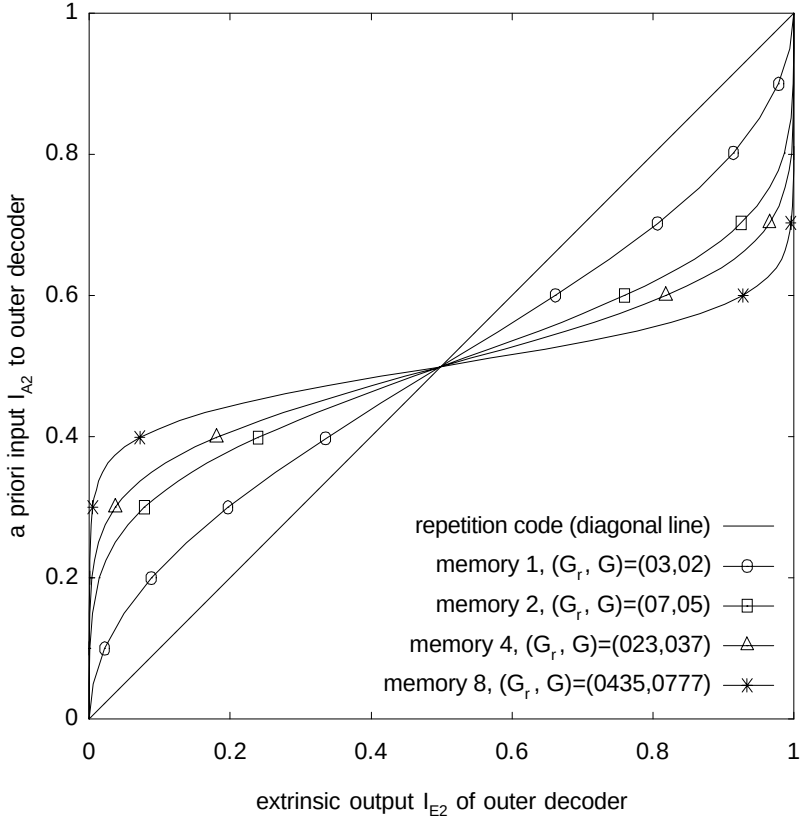
Démodulation et décodage itératifs : analyse EXIT



Analyse EXIT charts

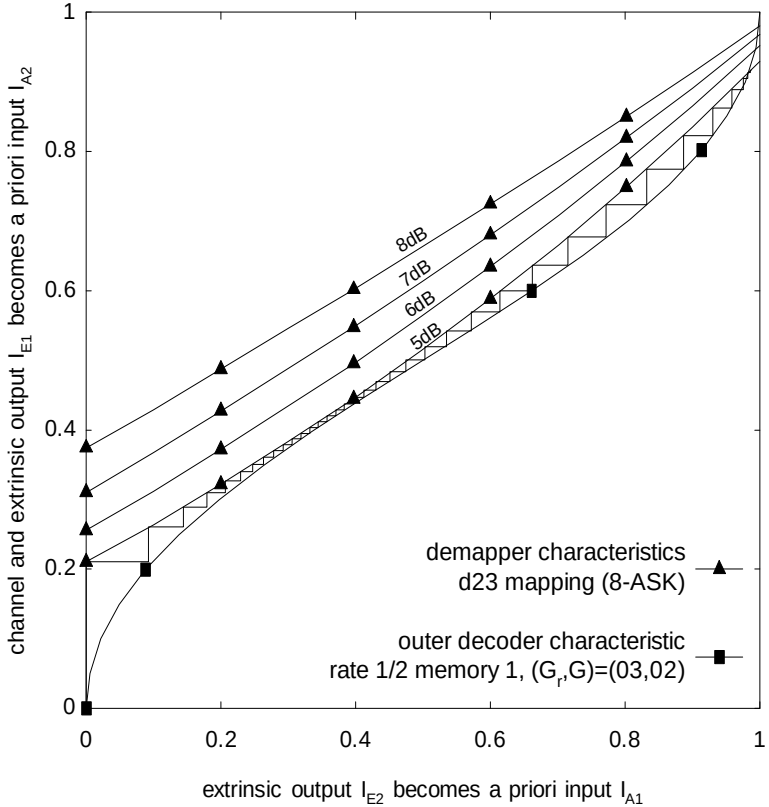
Démodulation et décodage itératifs : analyse EXIT

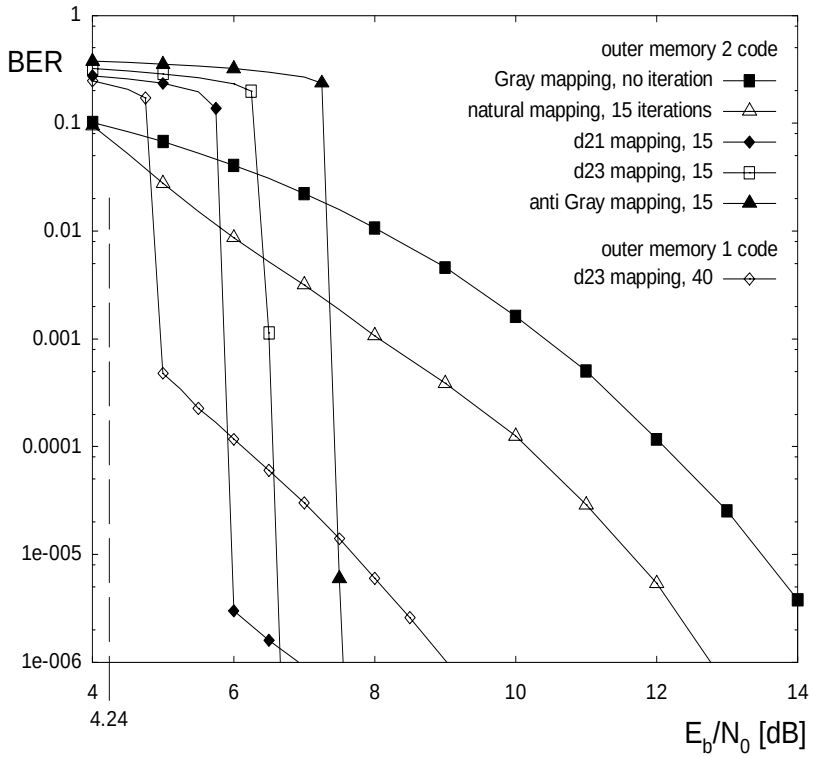




Analyse EXIT charts

Démodulation et décodage itératifs : analyse EXIT





$\Rightarrow$ Pas de gain en décodage itératif pour le mapping de Gary

Analyse EXIT charts

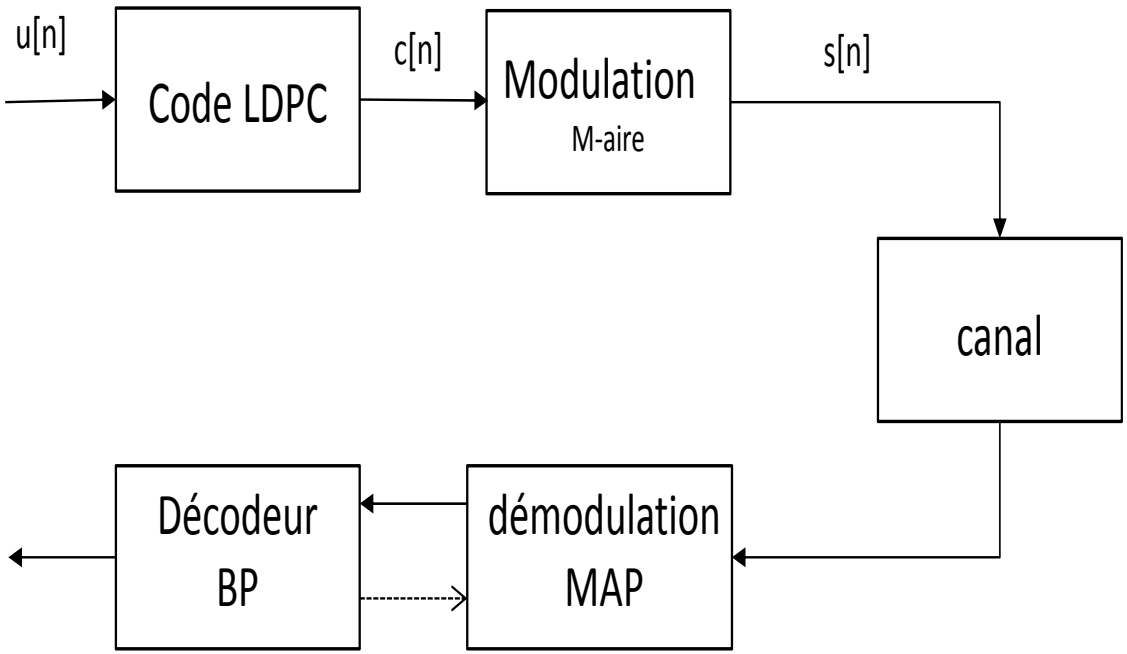
Démodulateur MAP avec Mapping de Gray

Cas du Mapping de Gray : pas d'itérations

$$
\begin{aligned}
L(x_i[n]) &= \log\left(\frac{\sum_{s[n]\in\mathcal{S}_0^i}\exp\left(-\frac{\|y[n]-s[n]\|^2}{2\sigma^2}\right)}{\sum_{s[n]\in\mathcal{S}_1^i}\exp\left(-\frac{\|y[n]-s[n]\|^2}{2\sigma^2}\right)}\right)\\
&= \max_{s[n]\in\mathcal{S}_0^i}{}^*\left(-\frac{\|y[n]-s[n]\|^2}{2\sigma^2}\right)\\
&\qquad -\max_{s[n]\in\mathcal{S}_1^i}{}^*\left(-\frac{\|y[n]-s[n]\|^2}{2\sigma^2}\right)\\
&= \max_{s[n]\in\mathcal{S}_0^i}{}^*\left(\frac{\mathrm{Re}(y)\mathrm{Re}(s)+\Im(y)\Im(s)}{\sigma^2}\right)\\
&\qquad -\max_{s[n]\in\mathcal{S}_1^i}{}^*\left(\frac{\mathrm{Re}(y)\mathrm{Re}(s)+\Im(y)\Im(s)}{\sigma^2}\right)
\end{aligned}
$$

BICM LDPC

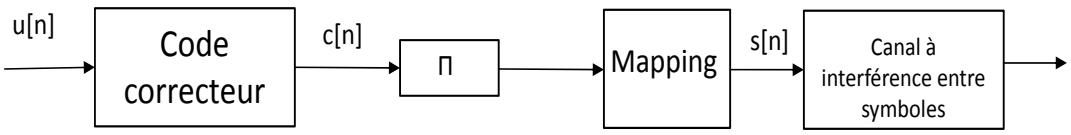
Principe générale

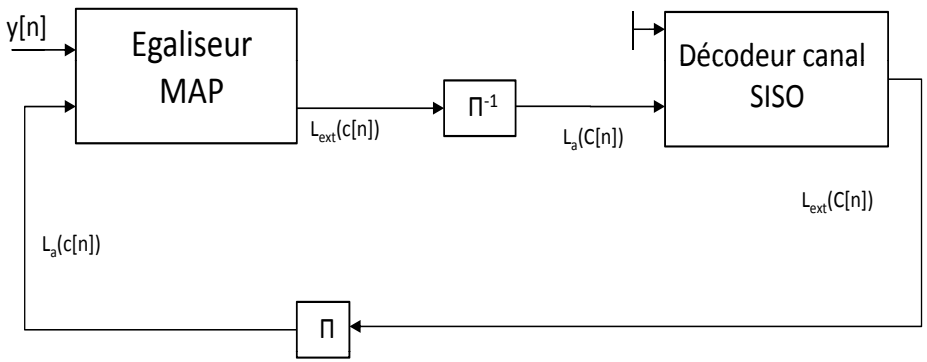


- pas d'entrelaceur nécessaire, contrairement au cas codes en treillis,
- pas d'itération nécessaire avec le démappeur si mapping de Gray

Turbo-égalisation

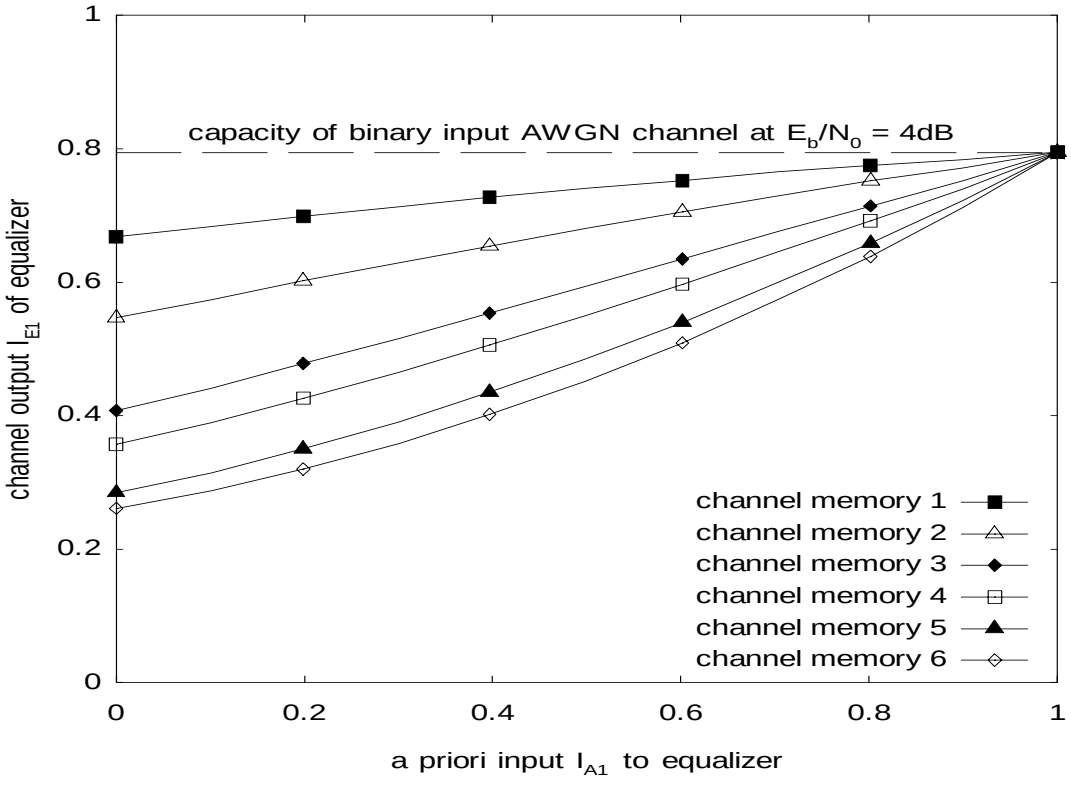
Principe générale





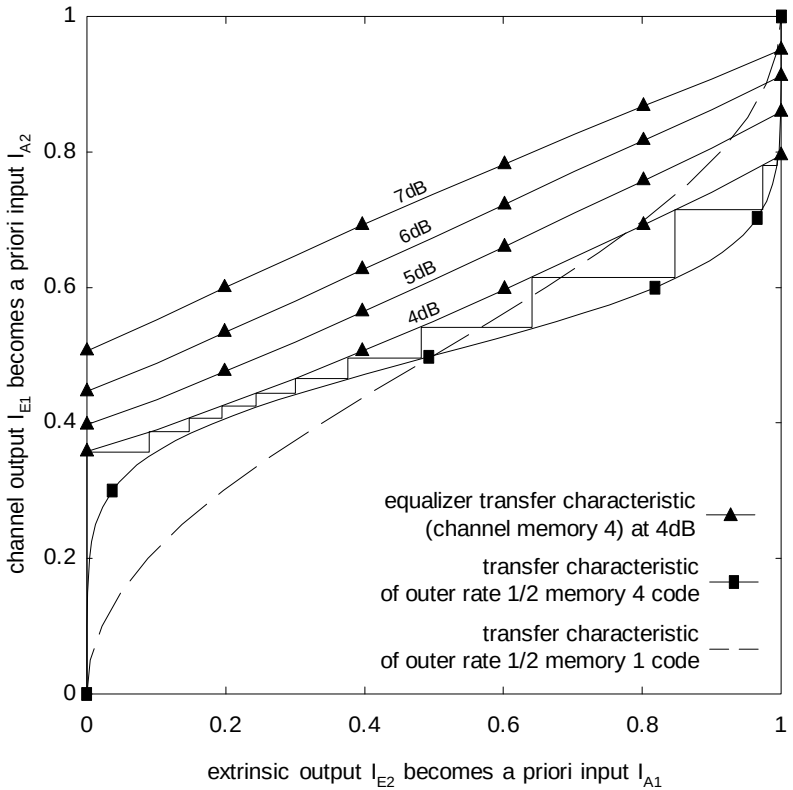
Turbo-égalisation

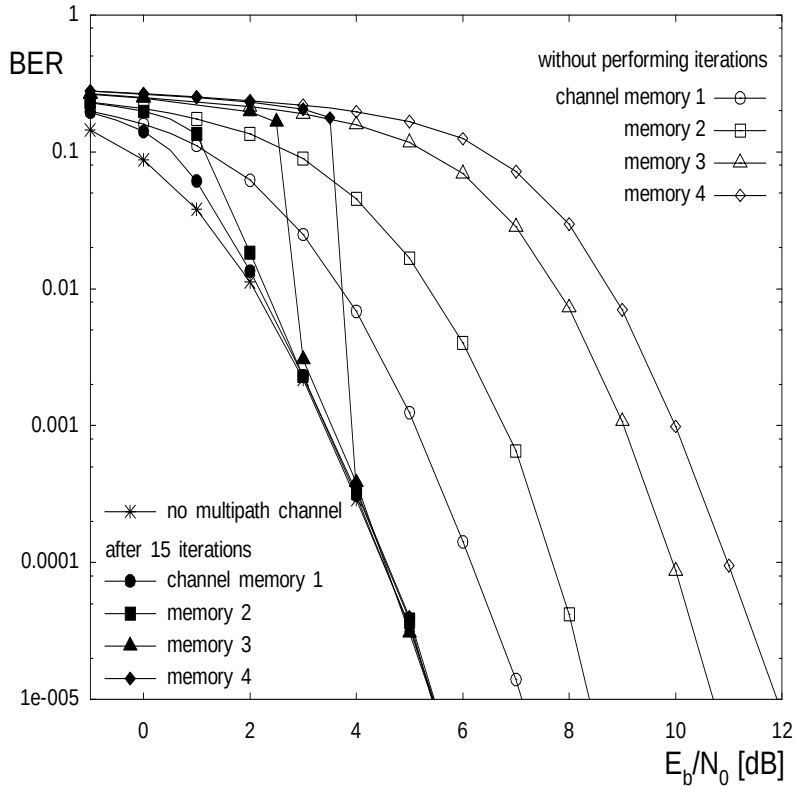
Analyse EXIT d'un canal



Turbo-égalisation

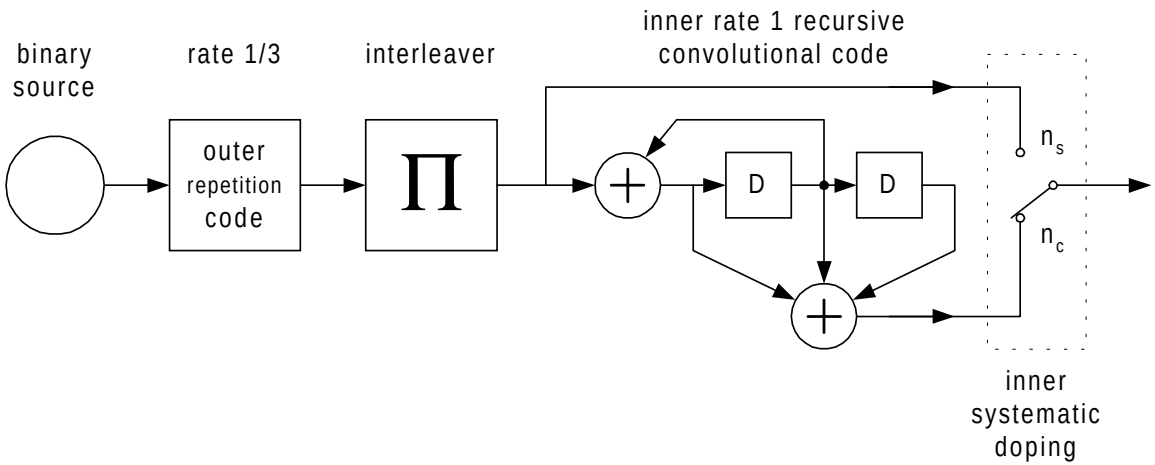
Analyse EXIT en turbo-égalisation





Codes concaténés en série

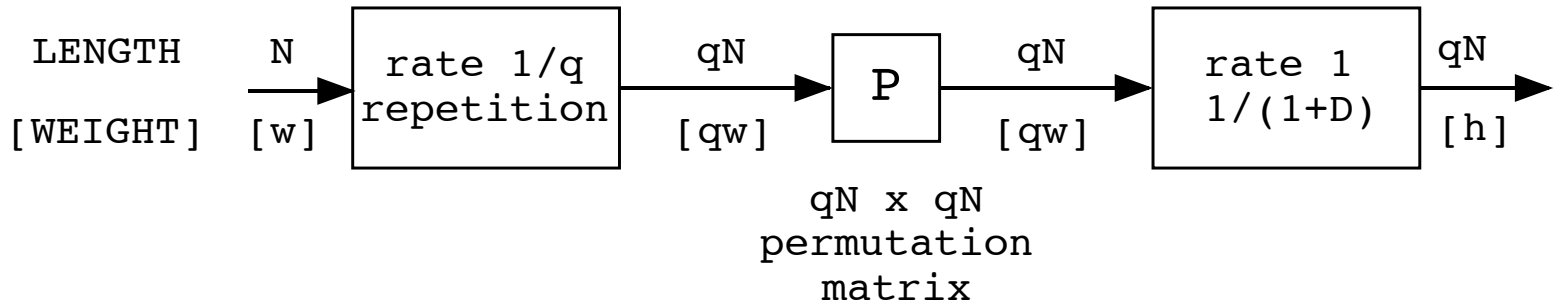
Des turbo-codes aux codes LDPC 1/2



Structure RA like proposée par S. ten Brink'00

Codes concaténés en série

Des turbo-codes aux codes LDPC 2/2



Codes RA originaux proposés par Divsalar, Allerton 98

Bibliographie

- A. Glavieux and all, *Channel coding in communication networks : from theory to turbocodes*, Volume 3 de Digital Signal Image Processing Series, John Wiley Sons, 2007.
- Claude Berrou and all, *Codes and Turbo Codes*, Collection IRIS Series, IRIS International, Springer, 2010.
- W.E. Ryan, Shu Lin, *Channel codes : classical and modern*, Cambridge University Press, 2009.
- Shu Lin, Daniel J. Costello, *Error control coding : fundamentals and applications*, Édition 2, Pearson-Prentice Hall, 2004.
- T. Richardson, R. Urbanke, *Modern coding theory*, Cambridge University Press, 2008.