



**INSTITUT
POLYTECHNIQUE
DE PARIS**



IP PARIS

Télécom paris

D1: Simulation d'un système avec des composants idéaux

Project C-RAN

Developed by:

Said Agouzal

Fernandes Maciel Edson

Academic year: 2024/2025

Contents

1	Introduction	1
2	Description des codes BCH étudiés	1
3	Méthodologie	1
4	Implémentation des composants clés	2
4.1	Encodage BCH	2
4.2	Modulation et démodulation	2
4.3	Canal de transmission	2
4.4	Détection des symboles	3
4.5	Décodage et correction d'erreurs	3
4.6	Évaluation des performances	3
5	Analyse des résultats	3
5.1	Génération et transmission des données	3
5.2	Comparaison des performances	4
5.3	Calcul des gains de codage	4
5.4	Comparaison de la capacité de détection des equalizers	5
6	Optimisations implémentées	7
6.1	Précalcul des syndromes	7
6.2	Traitement efficace des séquences	7
7	Conclusion	7

Rapport détaillé du Projet sur les codes BCH et canaux de transmission

1 Introduction

Ce Projet porte sur l'implémentation et l'évaluation des performances des codes correcteurs d'erreurs BCH (Bose-Chaudhuri-Hocquenghem) dans différents contextes de transmission numérique. L'objectif est d'analyser le comportement de ces codes face à différents types de canaux de transmission et d'évaluer leur efficacité en termes de taux d'erreur binaire (BER) par rapport au SNR (rapport signal sur bruit). On étudiera dans une 2nd partie la détection des symboles de modulation avec l'utilisation d'equalizer (ou détecteurs sous-optimaux), toujours dans le contexte d'un canal sélectif en fréquence.

2 Description des codes BCH étudiés

Les codes BCH constituent une famille de codes cycliques très utilisés pour la correction d'erreurs dans les systèmes de communication numérique. Dans ce Projet, deux types de codes BCH sont étudiés :

1. **BCH-1** : Code capable de corriger jusqu'à 1 erreur ($t = 1$)

- Longueur du code : 31 bits
- Longueur du message : 26 bits
- Bits de parité : 5 bits
- Polynôme générateur :

$$g(x) = 1 + x^2 + x^5$$

2. **BCH-2** : Code capable de corriger jusqu'à 2 erreurs ($t = 2$)

- Longueur du code : 31 bits
- Longueur du message : 21 bits
- Bits de parité : 10 bits
- Polynôme générateur :

$$g(x) = 1 + x^3 + x^5 + x^6 + x^8 + x^9 + x^{10}$$

3 Méthodologie

Le Projet est organisé selon la méthodologie suivante :

1. **Génération des séquences** : Création de séquences de bits aléatoires

2. **Codage BCH** : Application des codes BCH-1 et BCH-2 aux séquences générées
3. **Modulation** : Transformation des bits en symboles selon différentes modulations (BPSK, QAM-16)
4. **Transmission** : Passage des signaux à travers différents canaux (AWGN, sélectifs en fréquence)
5. **Démodulation** : Conversion des symboles reçus en bits
6. **Décodage** : Application des algorithmes de décodage (ML - Maximum Likelihood)
7. **Analyse des performances** : Calcul du BER et comparaison des gains

On retrouve donc certains blocs d'une chaîne de transmission numérique "classique". Pour la partie du sujet qui traite de la modulation, nous n'utiliserons pas de codage BCH (et donc de décodage), cela sera fait dans la partie Extra.

4 Implémentation des composants clés

4.1 Encodage BCH

La fonction `BCH.m` permet de générer les mots de code BCH-1 ($t = 1$) et BCH-2 ($t = 2$) à partir d'un message d'entrée. Cette fonction utilise le calcul du reste de la division par un polynôme générateur, implémenté dans `moduloG.m` qui calcul le reste de division de $m(x) \cdot x^{n-k}$ sur $g(x)$ en utilisant le circuit ce dessous :

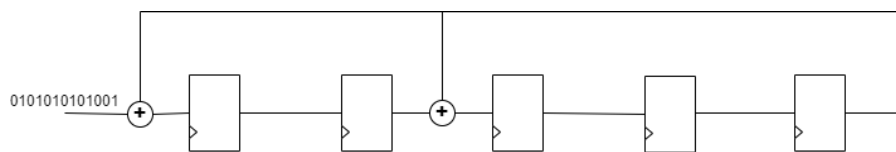


Figure 1: Circuit de shiftage pour le code **BCH-1**.

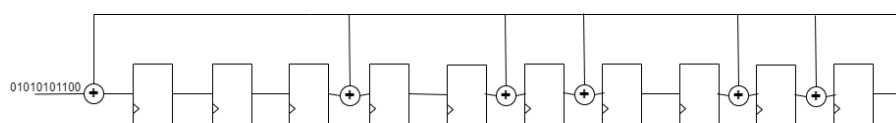


Figure 2: Circuit de shiftage pour le code **BCH-2**.

4.2 Modulation et démodulation

- `modulation.m` : Transforme les séquences de bits en symboles selon différentes modulations (BPSK, QAM)
- `demodulation.m` : Convertit les symboles reçus en bits après passage dans le canal

4.3 Canal de transmission

La fonction `selective_canal.m` simule différents types de canaux :

- Canal AWGN (bruit blanc gaussien additif)
- Canaux sélectifs en fréquence avec différents profils de délai et d'atténuation

4.4 Détection des symboles

Pour la détection des symboles, nous avons utilisé 3 méthodes de détection qui sont :

- `threshold_detector.m` : Cette fonction permet de faire un détecteur à seuil pour prendre une décision sur le symbole qui a été envoyé. Il ne corrige cependant pas les interférences intersymboles (IIS).
- `ZFE.m` : Cette fonction met en place un equalizer ZF, qui en faisant la pseudo inverse de la réponse du canal et couplé à une détection à seuil, permet de détecter les symboles tout en éliminant l'interférence. Il modifie cependant le bruit.
- `DFE.m` : Implémente un equalizer à retour de décision (DFE) permettant de supprimer les interférences inter-symboles en réalisant des opérations itératives basées sur les symboles précédemment détectés. Il exploite la décomposition QR pour transformer le canal en une structure triangulaire, facilitant la suppression progressive de l'ISI.

4.5 Décodage et correction d'erreurs

Deux implémentations du décodeur Maximum Likelihood sont proposées :

- `ML.m` : Version standard qui calcule les syndromes pour chaque motif d'erreur
- `ML2.m` : Version optimisée qui utilise des tables de syndromes précalculées via `initialisation_synd.`

4.6 Évaluation des performances

La fonction `BER.m` calcule le taux d'erreur binaire en comparant les bits décodés avec les bits originaux.

5 Analyse des résultats

Le script principal `plot_programme_BCH_2.m` réalise les expérimentations suivantes :

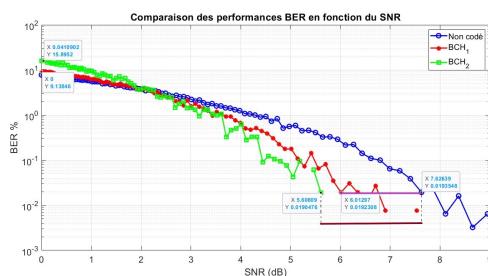


Figure 3: Pourcentage BER en fonction de SNR

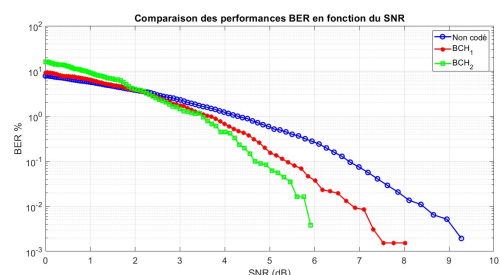


Figure 4: lissage de Figure 3

5.1 Génération et transmission des données

- 1000 séquences aléatoires de 31 bits
- Passage à travers 1000 canaux différents avec des SNR variables
- Modulation BPSK à travers un canal gaussien

5.2 Comparaison des performances

La figure compare les performances en BER de trois scénarios dans un canal AWGN avec modulation BPSK : transmission non codée, codage BCH corrigeant une erreur, et codage BCH corrigeant deux erreurs. On observe que le BER diminue avec l'augmentation du SNR pour les trois cas, mais que le codage BCH améliore significativement la performance. À faible SNR (0-3 dB), la différence est minime, car le bruit est dominant. À partir de 4 dB, les codes BCH montrent une nette amélioration, avec une meilleure correction pour BCH-2, qui surpasse BCH-1 pour $\text{SNR} > 5$ dB. Cela confirme l'efficacité croissante du codage en réduisant le BER, particulièrement en présence d'un bruit modéré à faible.

5.3 Calcul des gains de codage

- Gain empirique pour BCH-1 et BCH-2
- Comparaison avec les gains théoriques

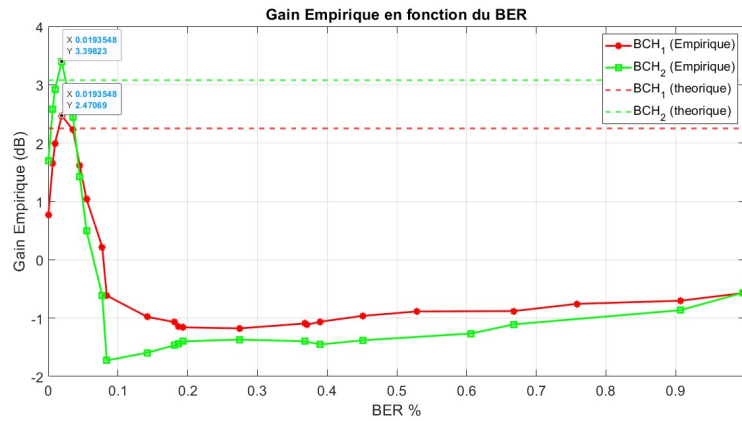


Figure 5: Comparaison entre gain de codage theorique et empirique

Les résultats montrent que :

- Les gains empiriques observés se rapprochent des valeurs théoriques calculées :
 - Pour BCH-1 : gain théorique de $10 \times \log_{10}((R \times (t + 1)))$ dB
 - Pour BCH-2 : gain théorique de $10 \times \log_{10}(R \times (t + 2))$ dB
 - Avec $R = k/n$ et le decodage et en dicision dure car on compare directement au syndrome

Le graphique présente le gain empirique en fonction du taux d'erreur binaire (BER) pour deux codes BCH : l'un capable de corriger une erreur (BCH-1) et l'autre capable d'en corriger deux (BCH-2). On observe que pour des BER faibles, le gain empirique est positif, avec une différence notable entre BCH-1 et BCH-2, en accord avec les gains théoriques indiqués en pointillés. Cependant, à mesure que le BER augmente, le gain diminue et devient négatif, indiquant que l'ajout de redondance ne compense plus la détérioration du signal. La divergence avec les valeurs théoriques est due à des approximations dans le modèle où le calcul est fait pour $\text{BER} \rightarrow 0$.

5.4 Comparaison de la capacité de détection des equalizers

Dans cette partie, le script principal est `Main.m`. nous considérons dans ce script la chaîne numérique composé de :

- trames de longueur $N = 100$,
- 3 différents canaux,
- Différentes modulation (BPSK, 8-QAM et 16-QAM),
- Différentes manières de faire une décision sur les symboles (detection à seuil, ZF, DFE),
- Calcul et comparaison de BER pour les différents equalizers et la detection à seuil.

Nous pouvons afficher dans un premier temps, la réponse impulsionnelle des 3 canaux normalisés:

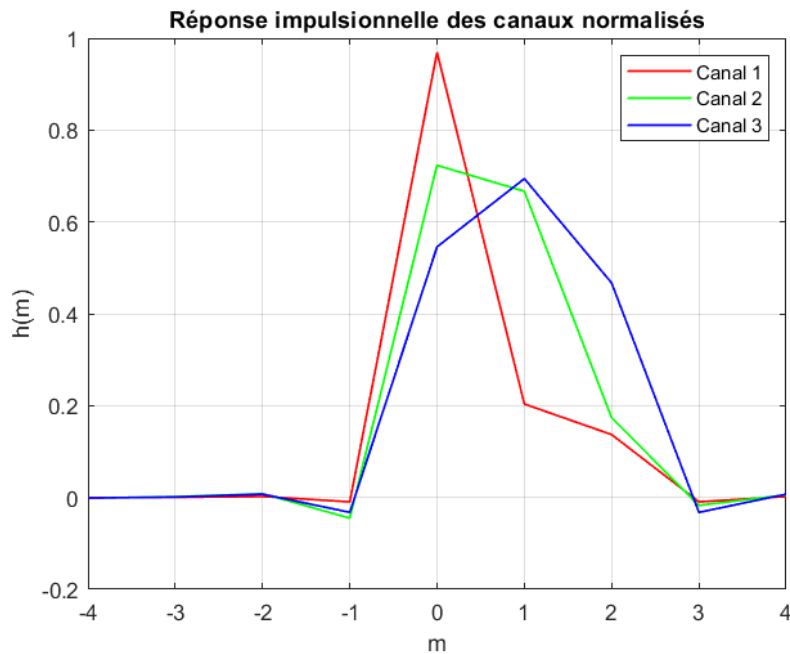


Figure 6: Réponse impulsionnelle des 3 canaux normalisés

On observe la présence d'interférence inter-symboles à cause des coefficients de $h(m)$ non négligeable jusqu'à $m=3$ devant $h(0)$.

Nous allons donc exécuter le script `Main.m` afin de comparer les BER des différents canaux en fonction du SNR ainsi que de la technique de detection utilisé.

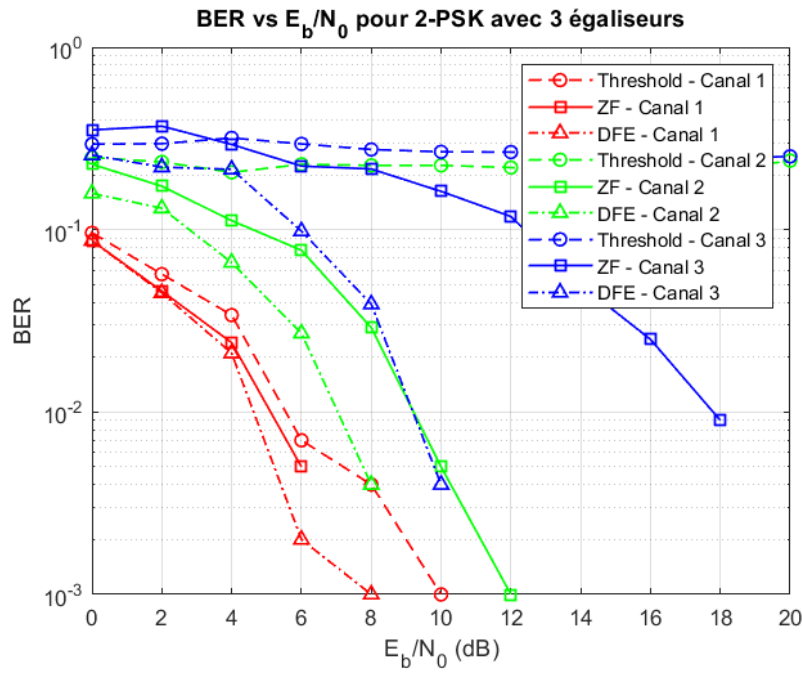


Figure 7: BER calculé à partir de différents equalizers en fonction du SNR pour une BPSK

On observe que le BER diminue rapidement pour le canal 1, nous avons un BER très faible dès lors que nous atteignons un SNR de 10 dB. Pour le canal 2, on observe que le détecteur à seuil n'est pas fiable car à un BER élevé pour un SNR assez grand (10 dB+). Enfin pour le canal 3, l'ISI induit par ce canal affecte également le ZF qui admet un BER de 10^{-2} pour un SNR de 18 dB. Enfin, pour l'equalizer DFE, nous avons un bon rapport SNR BER même pour le canal 3. (un $BER < 10^{-2}$ pour un SNR de 10 dB).

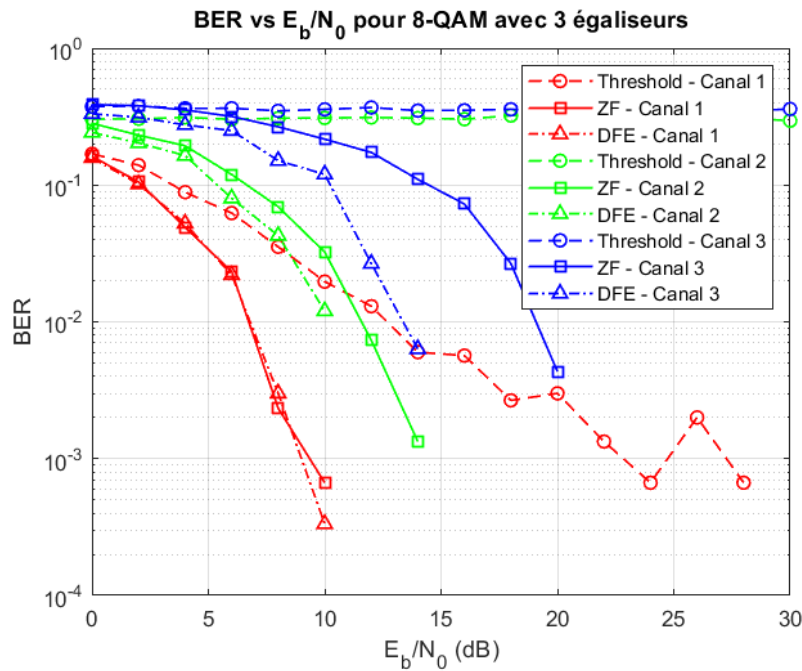


Figure 8: BER calculé à partir de différents equalizers en fonction du SNR pour une 8QAM

On remarque que comparé à la BPSK, le BER diminue plus lentement, ce qui est dû au nombre de bits plus important pour une modulation 8QAM ($3E_b = E_s$). Pour le canal 1, on observe que le ZF (à SNR = 10 dB) a le même BER que le threshold (à SNR = 27 dB), ce qui montre bien

l'efficacité du ZF avec la présence d'IIS par rapport à threshold. DFE à un meilleur BER que ZFE mais reste proche pour le canal 1. Pour le canal 2 et 3, la detection à seuil à un BER trop élevé par rapport au DFE. Tandis que le DFE se révèle moins efficace pour le canal 3.

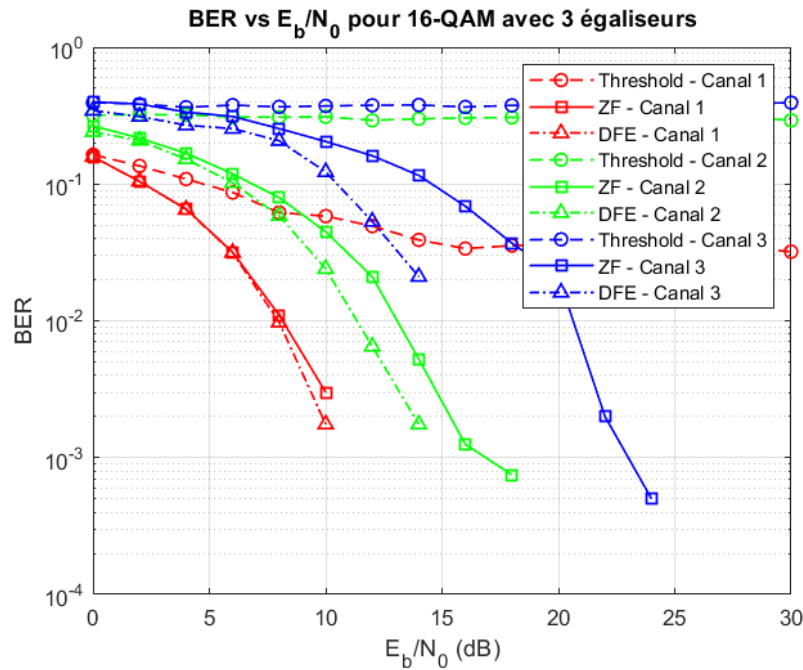


Figure 9: BER calculé à partir de différents equalizers en fonction du SNR pour une 16QAM

La figure est assez similaire à la 8QAM, à la différence où l'on observe que la detection à seuil atteint plus difficilement un BER faible par rapport aux 2 égaliseurs pour le canal 1. On peut donc conclure de ces figures, que plus le débit utilisé pour les modulations est élevé et les canaux sensibles aux IIS, plus il faudra un égalisateur performant. Ici le DFE est l'égalisateur le plus efficace.

6 Optimisations implémentées

6.1 Précalcul des syndromes

- La fonction `initialisation_synd.m` précalcule tous les syndromes possibles pour les motifs d'erreur
- `ML2.m` utilise ces tables pour accélérer le décodage

6.2 Traitement efficace des séquences

- Restructuration des données pour un traitement par blocs
- Utilisation de fonctions vectorisées pour améliorer la performance

7 Conclusion

Ce Projet démontre l'efficacité des codes BCH dans l'amélioration de la fiabilité des transmissions numériques en présence de bruit. Les points clés observés sont :

-
1. Les codes BCH offrent un compromis entre la redondance ajoutée (taux de codage) et la capacité de correction d'erreurs
 2. Le gain de codage augmente avec le nombre d'erreurs corrigibles (t)
 3. L'efficacité des codes BCH est particulièrement notable dans les canaux à bruit gaussien

Ces résultats confirment l'intérêt des codes BCH pour les systèmes de communication numérique où la fiabilité de transmission est cruciale.