

Présentation

SIT213

Groupe C1 : N. Bodin, G. Colin, A. Douant, J. Le Coq
FIP A2 Rennes - 22/10/2024

Table des matières

1 Introduction

Présentation du projet et des attentes du client

2 Organisation de l'équipe

Description de la structure de l'équipe, des outils utilisés et du processus de validation

3 Produit développé

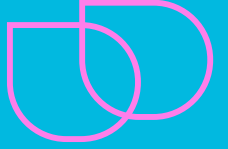
Explication des modèles physiques, des schémas et des conditions de simulation

4 Cas d'étude

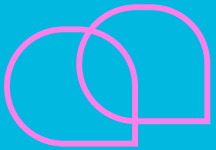
Analyse des résultats pour les environnements bruités et multi-trajets

5 Conclusion

Synthèse des performances obtenues et des perspectives d'amélioration



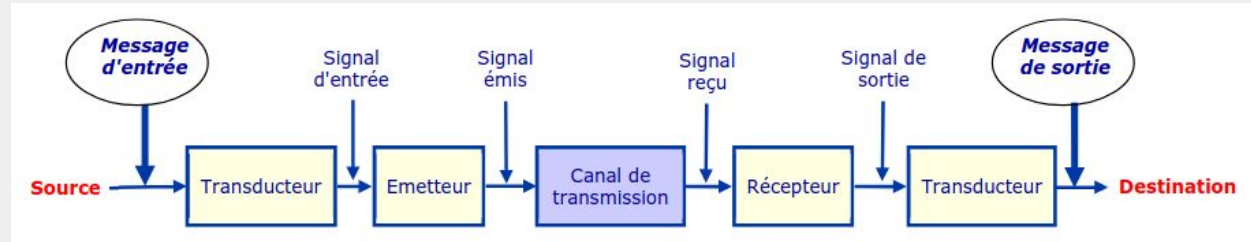
Introduction



Introduction

Objectif du projet

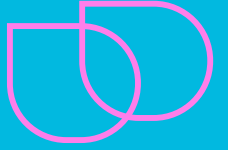
- Création d'un simulateur de chaîne de transmission :
 - Source
 - Logique
 - Analogique
 - Bruit
 - Multi-trajets
 - Codage
 - Destination (Calcul TEB)



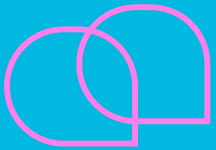
Introduction

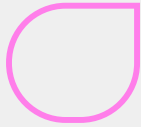
Contexte client

- Déploiement de réseaux de capteurs câblés
- Utilisation de notre simulateur
- Environnement 1 : Transmission bruitée avec capteurs sur batterie
 - Canal : Bruit blanc additif gaussien
 - Objectif : Optimiser la consommation énergétique tout en garantissant un $TEB \leq 10^{-3}$
 - Contrainte : Émetteurs alimentés par batterie
- Environnement 2 : Transmission sur canal multi-trajets
 - Canal : Multi-trajets avec trajets secondaires
 - Objectif : Garantir un débit binaire élevé tout en maintenant un $TEB \leq 10^{-2}$



Organisation de l'équipe

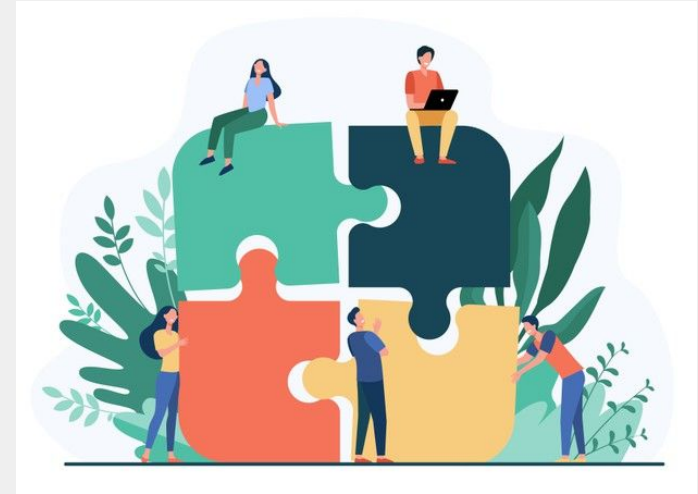


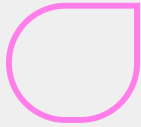


Organisation de l'équipe

Structure de l'équipe

- Noé : Chef de projet
 - Responsable de la gestion de l'équipe
- Guillaume : Relecteur
 - Relecture du code
- Antoine : Rédacteur
 - Responsable de la rédaction des rapports
- Justine : Relecteur
 - Relecture des livrables
- **Tout le monde participe au code**

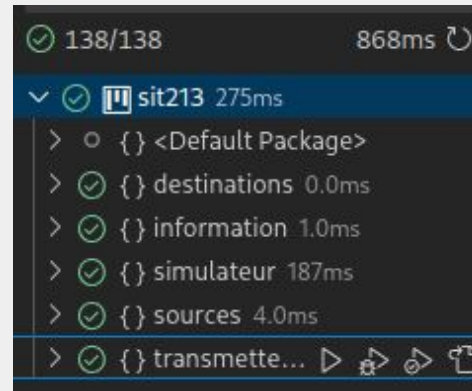




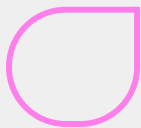
Organisation de l'équipe

Outils

- Git - Gitlab
- JUnit - Java
- Emma
- Eclipse / VS Code
- Google Drive



Element	Coverage
sit213	83,3 %
tests	90,8 %
src	75,0 %
transmetteurs	95,4 %
simulateur	51,7 %
visualisations	64,0 %
sources	100,0 %
information	96,6 %
destinations	100,0 %

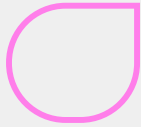


Organisation de l'équipe

Circuit de validation

- Branche Git
- Écriture du code
- Tests local
- Merge
- Tests global



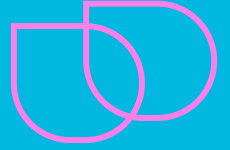


Organisation de l'équipe

Aspects testabilité

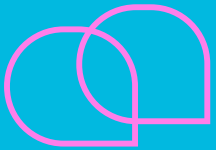
- Test dans la classe codée (méthode main)
- Implémentation dans la classe Simulateur
- Ajout classe de test JUnit
- Classe AllTests
- Vérification théorie
- Emma -> Objectif 100% couverture parfois impossible -> 100% != qualité

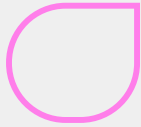




Produit développé

Simulateur de transmission numérique

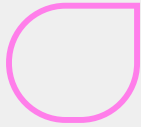




Produit développé

Modèles physiques implémentés

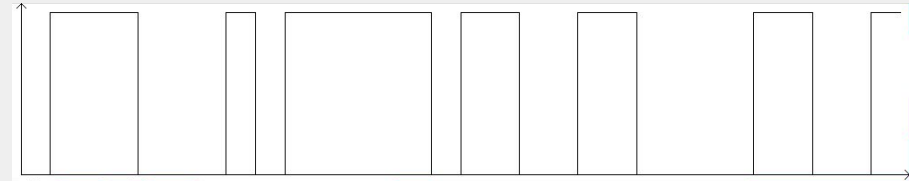
- Analogique
 - Forme RZ, NRZ, NRZT
 - Amplitude
 - Nombre d'échantillons par bits
- Bruit blanc gaussien
 - SNR dB
 - Eb/NO dB
- Multi-trajets
 - Décalage et atténuation
- CodageEmission, DecodageReception
 - Codeur



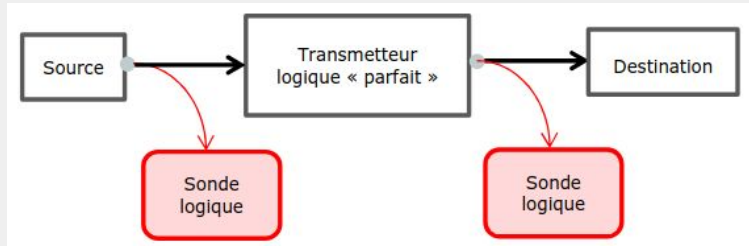
Produit développé

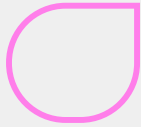
Transmission logique

- Source Fixe ou Aléatoire
- Pas vraiment d'intérêt physique
- Mais nécessaire pour :
 - Source et Destination
 - Calcul TEB



```
~/sit213 main > ./simulateur -s -mess 30  
java Simulateur -s -mess 30 => TEB : 0.0
```

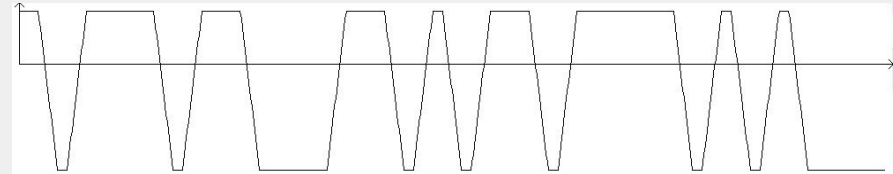




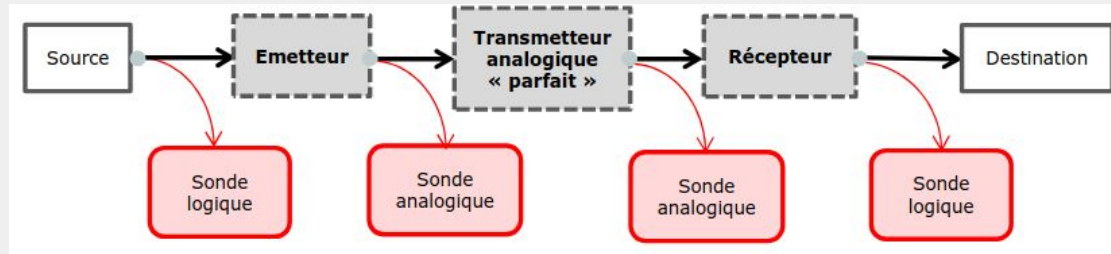
Produit développé

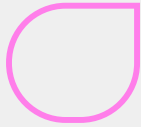
Signal analogique

- Forme d'onde (RZ, NRZ, NRZT)
- Amplitude (A_{min} , A_{max})
- Nombre d'échantillons par bits



```
~/sit213 main > ./simulateur -s -mess 30 -form NRZT -ampl -2 1  
java Simulateur -s -mess 30 -form NRZT -ampl -2 1 => TEB : 0.0
```

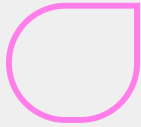




Produit développé

Schémas de réception utilisés

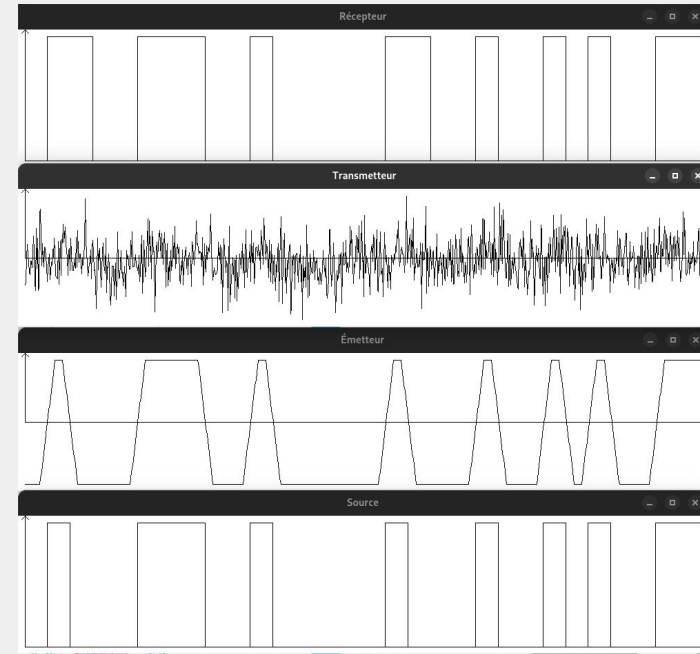
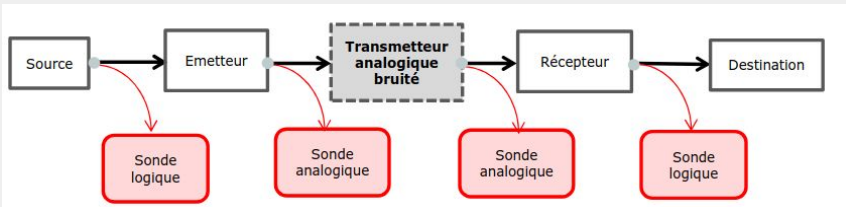
- Seuil de détection = $(A_{min} + A_{max})/2$:
- On vérifie la moyenne du bit :
 - NRZT et NRZ moyenne sur **tout** le bit
 - RZ sur le **deuxième tier** du bit avec **$A_{min} = 0$**
- Performances :
 - Changement pour NRZT :
 - -3dB pour même TEB



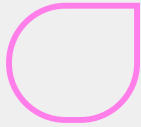
Produit développé

Canal bruité

- SNR ou E_b/N_0 (SNRpb)
- Bruit blanc gaussien
- Calcul avec :
 - Calcul puissance signal
 - Conversion SNR dB \rightarrow linéaire
 - Calcul puissance bruit
 - Calcul sigma
 - `random.nextGaussian() * sigma`



```
~/sit213 main > ./simulateur -s -mess 30 -form NRZT -ampl -1 1 -snrpb 0  
java Simulateur -s -mess 30 -form NRZT -ampl -1 1 -snrpb 0 => TEB : 0.0666667
```

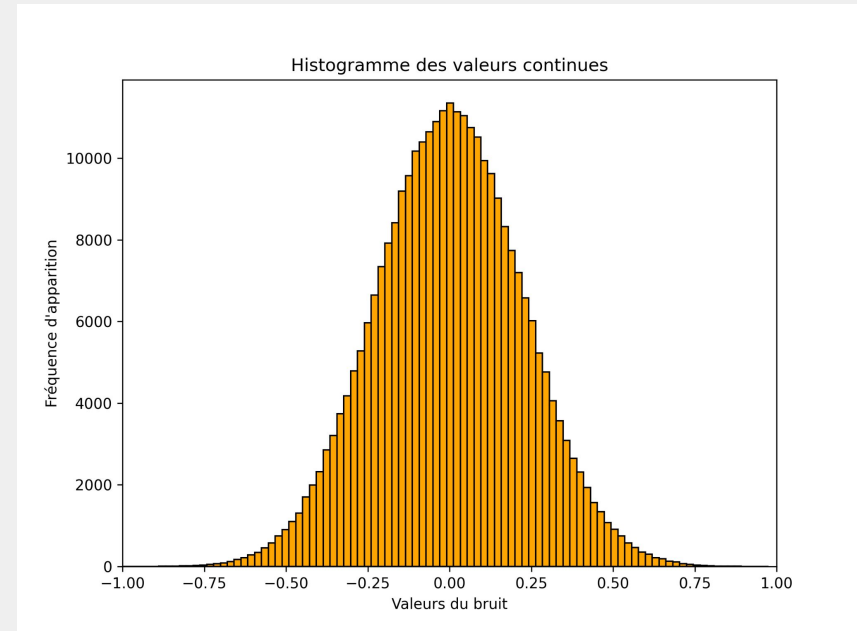
Produit développé

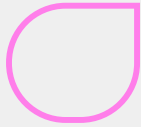
Canal bruité

Vérification :

- Histogramme
- Calcul puissances

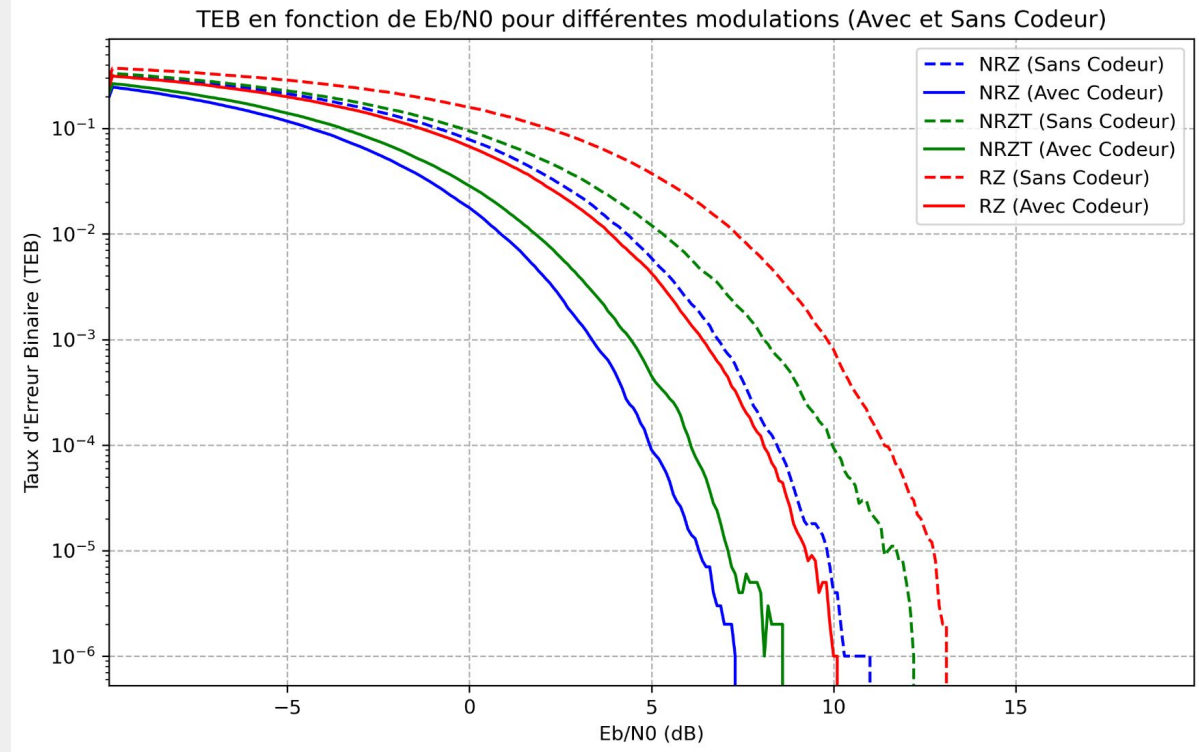
```
noe@inspiron-noe:~/sit213$ ./simulateur -snr 15
- Nombre de bits de la séquence : 100
- Nombre d'échantillons par bit : 30
1-> Puissance MOYENNE de la séquence de bits : 0.17666666666666667
2-> Valeur de sigma (écart-type du bruit) : 0.07468104274693668
3-> Puissance moyenne du bruit : 0.005577258145769785
4-> Rapport signal-sur-bruit (S/N, en dB) : 15.007338684191971
5-> Rapport Eb/N0 (en dB) : 26.768251274748785
java Simulateur -snr 15 => TEB : 0.0
```

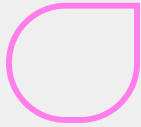




Produit développé

- Messages de 200 000 bits
- Amplitude -1 et 1
- 30 échantillons par bits

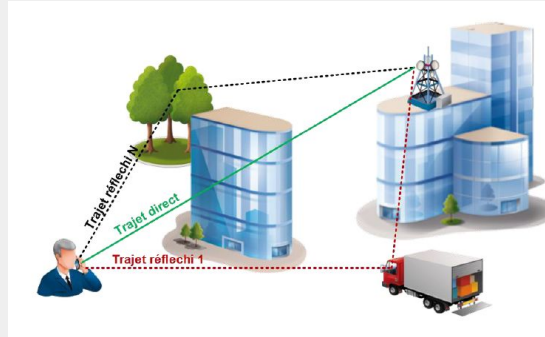




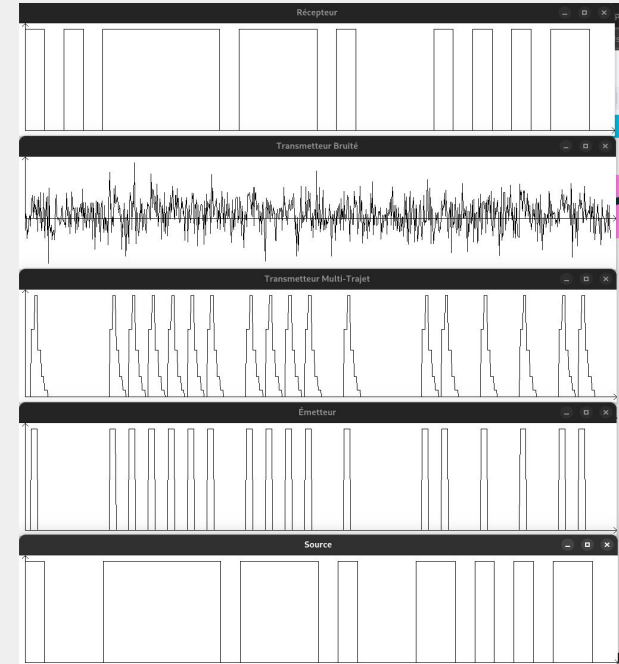
Produit développé

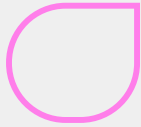
Multi-trajets

- Décalage en nombre d'échantillons
- Atténuation $0 < \alpha < 1$
- Avec ou sans AWGN
- Jusqu'à 5 trajets



```
~/sit213 main > ./simulateur -s -mess 30 -ti 5 0.5 10 0.2 15 0.1 -snrpb 5  
java Simulateur -s -mess 30 -ti 5 0.5 10 0.2 15 0.1 -snrpb 5 => TEB : 0.06666667
```

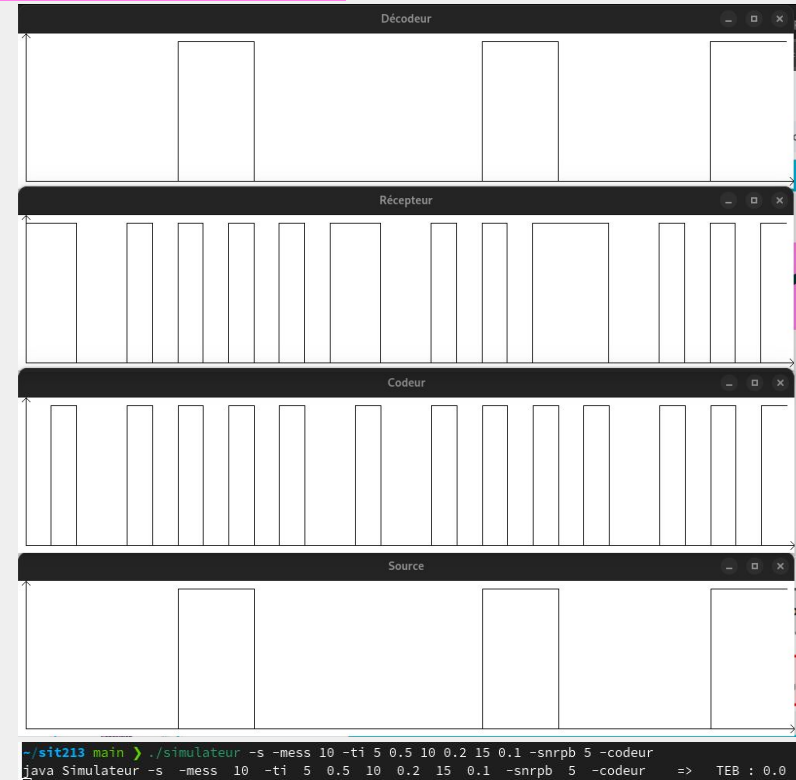
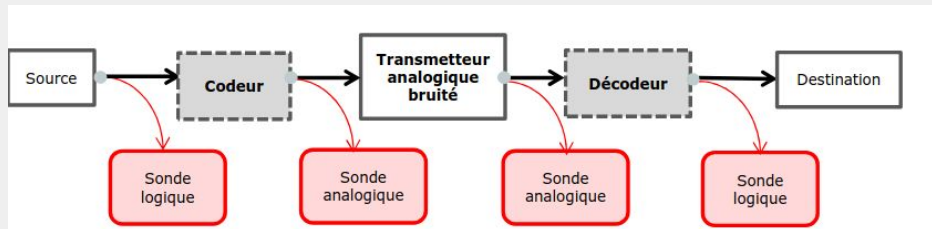


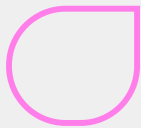


Produit développé

Codeur

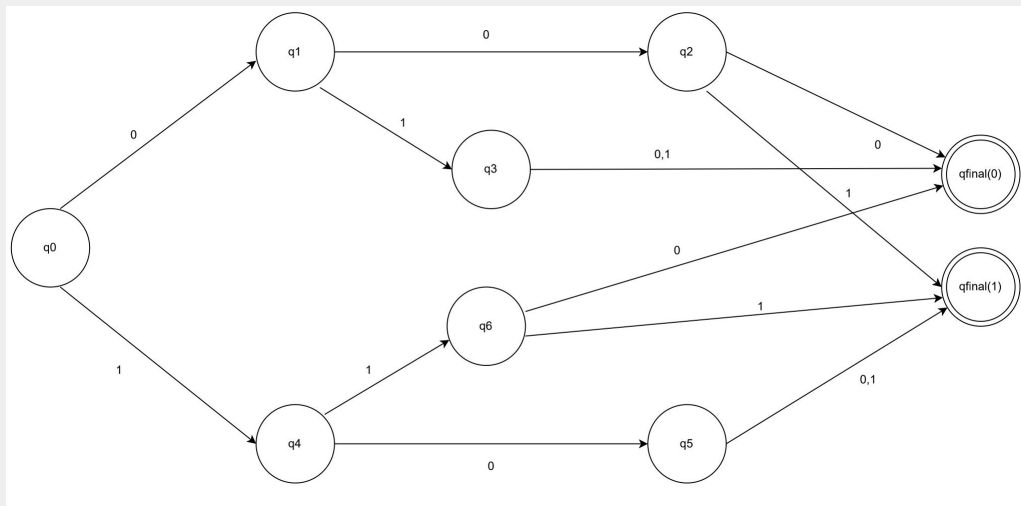
- Correction d'erreurs
- Avec n'importe quel type de transmissions
- Automate





Produit développé

Codeur - Automate



Automate du codeur

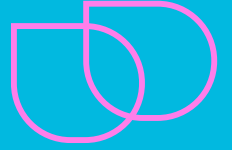
```
public static boolean automate(boolean[] bits) {
    if (bits.length != 3) {
        throw new IllegalArgumentException("Le tableau de bits doit avoir exactement 3 éléments.");
    }

    Etat etat = Etat.Q0; // état initial

    // Transition based on the first bit
    if (bits[0]) { // bit 1: 1
        if (bits[1]) { // bit 2: 1
            etat = (bits[2]) ? Etat.QFINAL1 : Etat.QFINAL0; // bit 3: 1 -> 1, 0 -> 0
        } else { // bit 2: 0
            etat = Etat.QFINAL1; // Always 1
        }
    } else { // bit 1: 0
        if (bits[1]) { // bit 2: 1
            etat = Etat.QFINAL0; // Always 0
        } else { // bit 2: 0
            etat = (bits[2]) ? Etat.QFINAL1 : Etat.QFINAL0; // bit 3: 1 -> 1, 0 -> 0
        }
    }

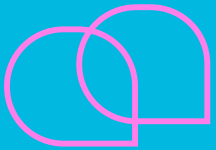
    // Return final output based on the final state
    return etat == Etat.QFINAL1;
}
```

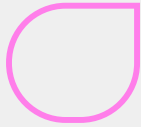
Implémentation



Cas d'étude soumis par le client

Étape 6

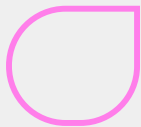




Cas d'étude soumis par le client

Environnement 1 : Transmission avec bruit blanc additif gaussien (BBAG)

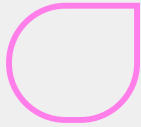
- Trouver E_b/N_0 : à partir du graphique (choisir meilleure forme)
- Calculer Puissance Reçue : $P_r = E_b * R_b$, $E_b = E_b/N_0$ (linéaire) * N_0 et R_b = débit binaire
- Calculer Puissance Émise : $P_t = P_r * 10^{(A/10)}$
- Calculer Autonomie : $Autonomie = Énergie\ batterie / P_t * (24\ heures * 3600\ secondes/heure)$
- Refaire avec Codeur, meilleur E_b/N_0 mais 3 fois plus de bits



Cas d'étude soumis par le client

Environnement 1 : Transmission avec bruit blanc additif gaussien (BBAG)

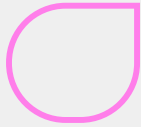
- Solution proposée : **NRZ**
- Puissance d'émission : **7,07 μ W**
- Durée de vie de la batterie : **4,9 jours**
- Ajout du code correcteur : autonomie de **3,4 jours**



Cas d'étude soumis par le client

Environnement 2 : Canal de propagation multi-trajets

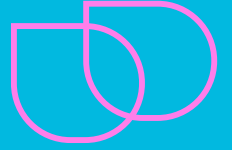
- Choisir fréquence d'échantillonnage : **$F_e = 10\text{MHz}$**
- Débit dépend de F_e et nombre d'échantillons par bits : **$D = F_e / \text{NbEch}$**
- On fait varier nbEch pour faire varier le débit, on mesure le TEB dans les conditions :
Simulation avec décalage de $\Delta t * F_e$ échantillons et atténuation 0.5
- On trouve un débit pour le TEB cible
- On fait avec le codeur, débit /3 mais moins d'erreurs : **$\text{NbEch} = F_e / 3 * D$**



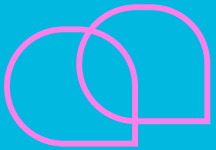
Cas d'étude soumis par le client

Environnement 2 : Canal de propagation multi-trajets

- Forme : **NRZ**
- Débit binaire maximal : **54 kbit/s**
- Conditions pour un $TEB \leq 10^{-3}$: **Ajout du codeur (presque plus d'erreurs)**
- Pour amplitude **0 et 1**, si **-1 et 1** débit infini...



Conclusion



Conclusion

Limites et perspectives

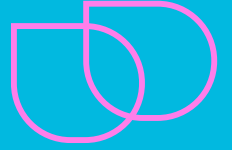
- Limité en bande de base
- Bruit blanc gaussien, modèle courant mais pas parfait
- Multi-trajets, utile mais peu pratique
- De façon générale simulateur fonctionnel, mais basique
- Compliqué de comparer/utiliser pour un cas réel
- Ajout de modulation sur fréquence porteuse (PSK, QAM, ...)
- Meilleurs codes correcteurs d'erreurs
- Meilleurs modèles de bruits
- Amélioration du récepteur

Conclusion

Leçons tirées

- La relecture est importante
- Travail sur des branches séparées
- Vérification à chaque merge
- Analyse approfondi des demandes
- Communication au sein de l'équipe
- Ne pas utiliser de LinkedList





Des questions ?

