IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

24/09/2024

BODIN Noé
COLIN Guillaume
DOUANT Antoine
LE COQ Justine

Rapport SIT213

Atelier Logiciel

Simulation d'un système de transmission

Étape 3

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

Introduction

L’étape 3 du projet introduit un scénario de transmission plus réaliste où le canal de communication n’est plus parfait, mais bruité. Dans les systèmes de communication, les signaux sont souvent affectés par des interférences aléatoires, comme le bruit blanc additif gaussien, qui dégrade la qualité de la transmission et augmente le risque d’erreurs à la réception. L’objectif de cette étape est de modéliser un canal bruité en ajoutant ce type de bruit au signal transmis, afin d’étudier l’impact sur la transmission, en particulier en mesurant le Taux d’Erreur Binaire.

Le bruit blanc gaussien est une perturbation qui suit une distribution normale centrée et peut affecter chaque échantillon du signal analogique pendant la transmission. En fonction du rapport signal sur bruit, le bruit ajouté au signal peut rendre la détection des bits plus ou moins difficile pour le récepteur, entraînant des erreurs dans la séquence reçue. Le TEB devient un indicateur clé de la qualité de la transmission, mesurant la proportion de bits erronés par rapport au nombre total de bits transmis.

L'organisation du rapport vise à détailler chaque étape du projet, en expliquant la conception des composants, les tests réalisés, ainsi que les résultats obtenus lors des simulations.

1. Conception des composants

1.1 TransmetteurAnalogiqueBruite

Le TransmetteurAnalogiqueBruite est une extension du Transmetteur qui modélise un environnement de transmission bruité. Ce bruit est aléatoire et suit une distribution gaussienne, ce qui permet de simuler les effets des interférences présentes dans un canal de communication réel.

Fonctionnement du TransmetteurAnalogiqueBruite :

1. **Réception du signal analogique :**
    - Le transmetteur reçoit une séquence analogique depuis le récepteur.
2. **Ajout du bruit gaussien :**
    - Le TransmetteurAnalogiqueBruite y ajoute un bruit gaussien. Le bruit est généré aléatoirement en suivant une distribution normale centrée sur 0, avec une variance/écart-type déterminée par le SNR (sigma = Math.sqrt(puissanceBruit)) ⇒ converti à partir du Eb/N0 en paramètre, calculé par deux méthodes "`calculerPuissanceSignal`" et "`calculerPuissanceBruit`"
    - Chaque échantillon du signal analogique se voit ajouter un bruit indépendant pour simuler les interférences
3. **Transmission vers le Récepteur :**
    - Le signal bruité est ensuite transmis au Récepteur, qui le reçoit et l'analyse. L'ajout de bruit peut entraîner des erreurs de réception, notamment une modification de la forme du signal, ce qui augmentera le TEB à la réception.

2. Connexion des composants

La chaîne de transmission dans le simulateur est composée de plusieurs composants interconnectés qui permettent la génération, la modulation, la transmission et la réception d'informations. La chaîne de transmission bruitée est activée dès que le paramètre -snrpb est ajouté (!=0).

2.1 Connexion dans la chaîne de transmission analogique avec bruit

Dans le cadre d'une transmission analogique avec bruit, la chaîne de composants inclut des étapes supplémentaires pour la modulation, l'ajout de bruit, et la démodulation. La connexion des composants se fait de la manière suivante :

- **Source → Émetteur** :
  La source génère une information logique, qui peut être une séquence binaire fixe ou aléatoire, selon la configuration du simulateur. L'information logique générée est transmise à l'émetteur
- **Émetteur → Transmetteur analogique bruité** :
  L'émetteur reçoit l'information logique et la convertit en un signal analogique en fonction de la modulation sélectionnée (NRZ, NRZT ou RZ) et des autres paramètres (Amin, Amax, nbEchantillonsParBit). Une fois le signal analogique généré, celui-ci est transmis au TransmetteurAnalogiqueBruite, qui modélise un canal bruité en ajoutant un bruit blanc gaussien au signal analogique en fonction du SNR par bit choisi. Ce transmetteur simule les effets du bruit sur la transmission en ajoutant des perturbations.
- **Transmetteur analogique bruité → Récepteur** :
  Le signal analogique bruité est ensuite transmis au récepteur. Ce dernier reçoit le signal dégradé par le bruit et effectue la démodulation en fonction de la modulation sélectionnée à l'émission (NRZ, NRZT, ou RZ). Le récepteur convertit le signal analogique en une séquence logique (booléenne) correspondant aux informations émises
- **Récepteur → Destination finale** :
  Le récepteur transmet enfin l'information logique récupérée à la DestinationFinale, qui reçoit la séquence de bits démodulée et la compare avec l'information

initialement émise par la source. La destination finale utilise ces données pour calculer le Taux d'Erreur Binaire, permettant d'évaluer la qualité de la transmission en prenant en compte les erreurs introduites par le bruit

2.2 Rôles des sondes dans la connexion

Les sondes peuvent être activées dans le simulateur pour observer :

- **Sonde logique à la source** :
  Permet de visualiser l'information binaire émise par la source, avant toute modulation
- **Sonde analogique après l'émetteur** :
  Permet de capturer et visualiser le signal analogique généré par l'émetteur avant l'ajout de bruit par le transmetteur. Cela permet de vérifier que la modulation est correctement effectuée
- **Sonde analogique après le transmetteur analogique bruité** :
  Permet de visualiser le signal analogique une fois le bruit gaussien ajouté, afin d'observer l'impact du bruit sur la forme du signal
- **Sonde logique après le récepteur** :
  Permet de visualiser l'information logique démodulée par le récepteur et de la comparer avec le signal émis par la source, ce qui facilite l'analyse des erreurs éventuelles

3. Tests

3.1 Objectifs des tests

L'objectif principal des tests est de valider le bon fonctionnement de chaque composant du simulateur, ainsi que l'intégration globale des différents modules dans la chaîne de transmission. Les tests ont été conçus pour vérifier que :

- **Chaque composant** (source, transmetteur, émetteur, récepteur, destination) fonctionne conformément à ses spécifications.
- **Les interactions entre les composants** : lorsque connectés, ils produisent les résultats attendus, notamment en termes de transmission correcte des informations, de modulation et de démodulation.
- **Le calcul du TEB** est correct et reflète fidèlement la qualité de la transmission.
- **La robustesse des composants** face à des transmissions bruitées, en analysant les erreurs introduites par le bruit gaussien.

3.2 Structure des tests

Les tests ont été organisés en une suite de tests JUnit, permettant d'exécuter tous les tests de manière groupée. Voici les principales classes de test utilisées :

- **TransmetteurAnalogiqueBruite** : Le TransmetteurAnalogiqueBruite couvre la conversion de l'information logique en signal analogique avec les différentes formes de modulation (NRZ, NRZT, RZ). Il vérifie que la conversion est correcte, que les signaux analogiques générés ont la bonne amplitude pour les bits 0 et 1, et que le bruit gaussien est correctement appliqué pour simuler un canal bruité.
- **Simulateur** : La chaîne de transmission complète est testée, de l'émission à la réception, pour valider l'interaction entre les composants (source, transmetteur, récepteur, destination). Les tests incluent la vérification que l'ensemble fonctionne correctement, même dans des environnements bruités, en calculant le TEB pour des transmissions avec et sans erreurs.

3.3 Exécution des tests

Les tests ont été exécutés via la classe AllTests, qui regroupe l'ensemble des tests unitaires et d'intégration. Cette classe permet d'automatiser l'exécution des tests et de vérifier, en une seule étape, que tous les composants fonctionnent comme attendu, aussi bien individuellement qu'en interaction les uns avec les autres.

Les tests ont été lancés à l'aide d'un script runTests qui exécute tous les tests dans un environnement contrôlé, garantissant ainsi que le code est testé systématiquement avant chaque nouvelle livraison ou itération du projet. Cela prend plusieurs dizaines de secondes.

3.4 Résultats des tests

Pour les simulations avec bruit gaussien, le TEB varie en fonction du rapport SNR par bit, comme attendu. Les tests montrent que le TEB augmente proportionnellement à la dégradation du rapport signal-bruit, reflétant bien l'impact du bruit sur la transmission. Tous les tests sont validés.

3.5 Couverture de code avec Emma

L'outil Emma a été utilisé pour mesurer la couverture des tests sur l'ensemble du projet. Voici les résultats de la couverture de code :

Element	Coverage	Covered Instructions	Missed Instructions	Total Instructions
sit213	87,2 %	5453	802	6255
src	83,2 %	2866	579	3445
transmetteurs	94,0 %	1181	75	1256
Emetteur.java	95,4 %	436	21	457
Recepteur.java	96,2 %	333	13	346
TransmetteurAnalogiqueBruite.java	87,8 %	294	41	335
TransmetteurAnalogiqueParfait.java	100,0 %	47	0	47
TransmetteurParfait.java	100,0 %	36	0	36
Transmetteur.java	100,0 %	35	0	35
visualisations	65,4 %	618	327	945
simulateur	81,0 %	737	173	910
sources	100,0 %	174	0	174
information	96,6 %	113	4	117
destinations	100,0 %	43	0	43
tests	92,1 %	2587	223	2810

Ces résultats montrent que la majorité du code a été testé de manière quasi-exhaustive. La couverture légèrement plus faible de certains éléments, indique que des cas spécifiques peuvent être ajoutés lors des futures itérations pour renforcer la robustesse des tests.

Globalement, la couverture est très satisfaisante, assurant que le projet a été largement vérifié pour détecter d'éventuelles anomalies ou régressions. Certaines classes ou méthodes n'ont pas besoin d'être testées (par exemple, la création des fichiers), ceci influe sur la couverture des tests.

4. Simulation d'une transmission avec un canal bruité (gaussien)

Dans cette section, nous explorons la transmission d'un signal analogique à travers un canal bruité modélisé par un bruit blanc gaussien. L'objectif est de mesurer l'impact du bruit sur la qualité de la transmission en analysant l'évolution du taux d'erreur binaire en fonction du rapport signal sur bruit (SNR). Les performances des différentes modulations (NRZ, NRZT, RZ) sont comparées afin de déterminer leur résistance face aux perturbations. De plus, un histogramme du bruit gaussien est généré pour vérifier la distribution du bruit ajouté au signal.

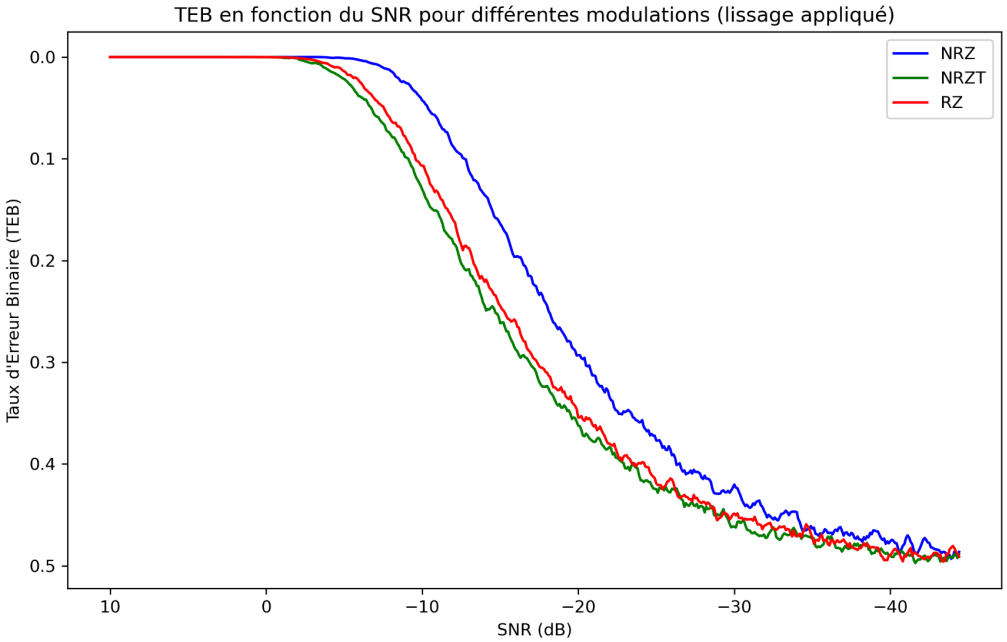
4.1 Évolution du TEB en fonction du SNR

Pour mesurer les performances du système, nous utilisons le simulateur avec le paramètre **-snrpb <float>** qui permet de spécifier le rapport signal sur bruit, calculé à partir de Eb/N0 en paramètre. En effectuant plusieurs simulations, nous obtenons les résultats suivants.
Pour visualiser les performances des différentes modulations en fonction du bruit, nous allons faire des graphes affichant le TEB en fonction du SNR. Un meilleur TEB signifie des meilleures performances.
La classe SimulateurTEB permet de générer 3 fichiers CSV contenant les différents TEB pour différents SNR pour chaque type de modulation, on définit le SNR min et max ainsi que le pas, et le nombre de simulations par SNR (afin de générer une moyenne). Ensuite des boucles font les simulations pour chaque valeur de SNR.

Ensuite, nous utilisons python pour générer les courbes, elles sont lissées pour une meilleure lisibilité, en faisant une moyenne mobile sur 5 points (np.convolve(y, np.ones(window_size) / window_size, mode='same')) (Les librairies Matplotlib, numpy et pandas sont nécessaire).

L'axe des ordonnées indique le Taux d'Erreur Binaire de façon décroissante, un TEB de 0.0 indique qu'aucune erreur a eu lieu, donc un TEB plus faible (haut sur la courbe) indique de meilleures performances. \
Ensuite, l'axe des abscisses indique le SNR en dB de façon décroissante, un SNR élevé (à gauche de la courbe) indique peu de bruit.
On remarque que jusqu'à un SNR de 0 dB, le TEB est stable et il n'y a aucune erreur, pour tous les types de modulations.
Ensuite, les performances diffèrent en fonction de la modulation. Le NRZ est le plus résistant au bruit puisque c'est le dernier à décroitre, puis RZ et NRZT sont proches.

On atteint un plateau après -30 dB avec un TEB de 0.5, soit une chance sur deux. Le signal n'est plus du tout ce qu'il était et aucun bit n'est déterminé. On a donc une chance sur deux d'avoir le bon.
En faisant une moyenne (sur 10 fois, avec messages de 300 bits), on obtient cette courbe (axes inversés) :



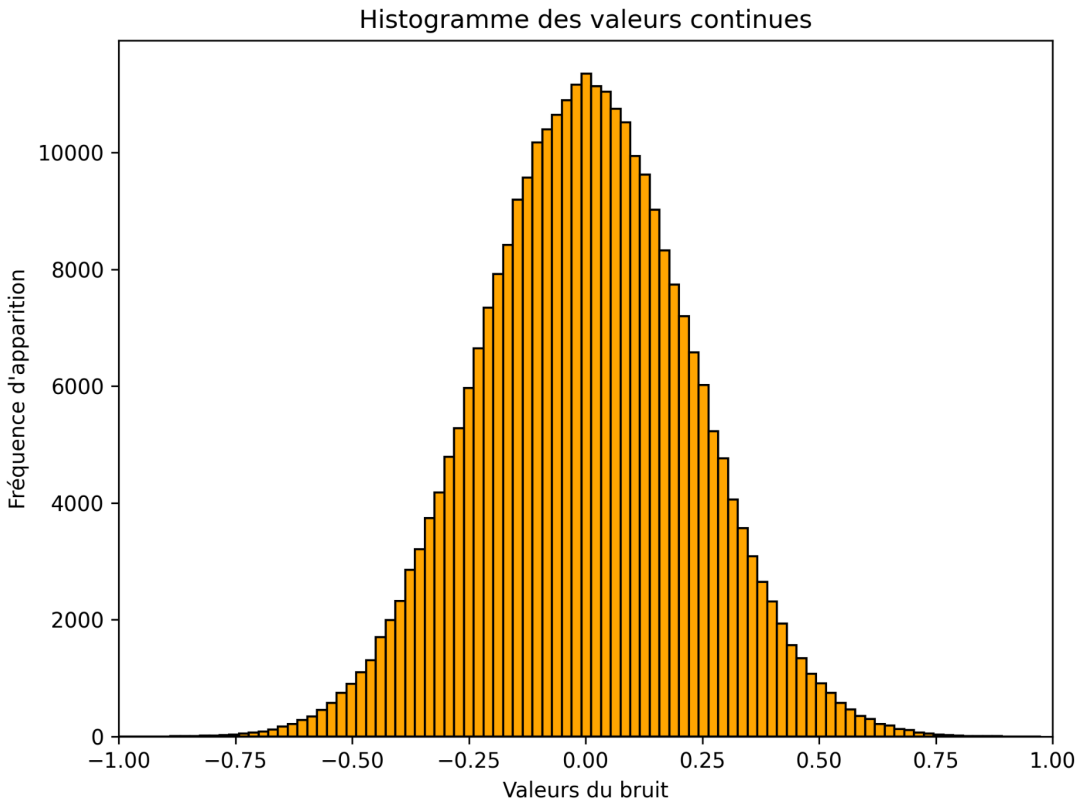
La méthode de détection peut certainement être améliorée pour avoir de meilleures performances pour tous les types de modulations.

Une autre méthode aurait pu être d'utiliser une échelle logarithmique afin de mieux visualiser et analyser les résultats, sur des zones plus precises. Les simulations n'ont pas données de résultats plus intéressant.

4.2 Histogramme du bruit

Pour générer un histogramme du bruit ajouté, nous sauvegardons le bruit généré et trié dans un fichier. Pour cela, il faut que le paramètre genererFichierBruit soit true. Puis, nous utilisons un script python pour générer l'histogramme (La librairie Matplotlib est nécessaire).

Voici le résultat pour un signal de 10 000 bits, avec 30 échantillons par bits et un SNR (sur tout le signal) de 10 dB :



L'histogramme montre une distribution gaussienne centrée sur zéro, confirmant la nature du bruit blanc gaussien. Cela est cohérent avec la théorie du bruit gaussien, où les valeurs sont distribuées symétriquement autour de zéro, avec une fréquence plus élevée pour les valeurs proches de la moyenne et décroissante pour les valeurs extrêmes.

L'écart type dépend bien du niveau du SNR demandé. L'écart type est la racine carrée de la puissance. Cela veut dire qu'un SNR de 10 indique une faible puissance du bruit. Les valeurs sont concentrées en -0,25 et 0,25, ce qui veut dire que le signal change très peu. Dans notre cas, Amax et Amin sont de 0 à 1. Donc avec une faible puissance de bruit.

Le fait que l'écart type et la moyenne correspondent aux attentes théoriques confirme la validité de notre modèle de bruit.

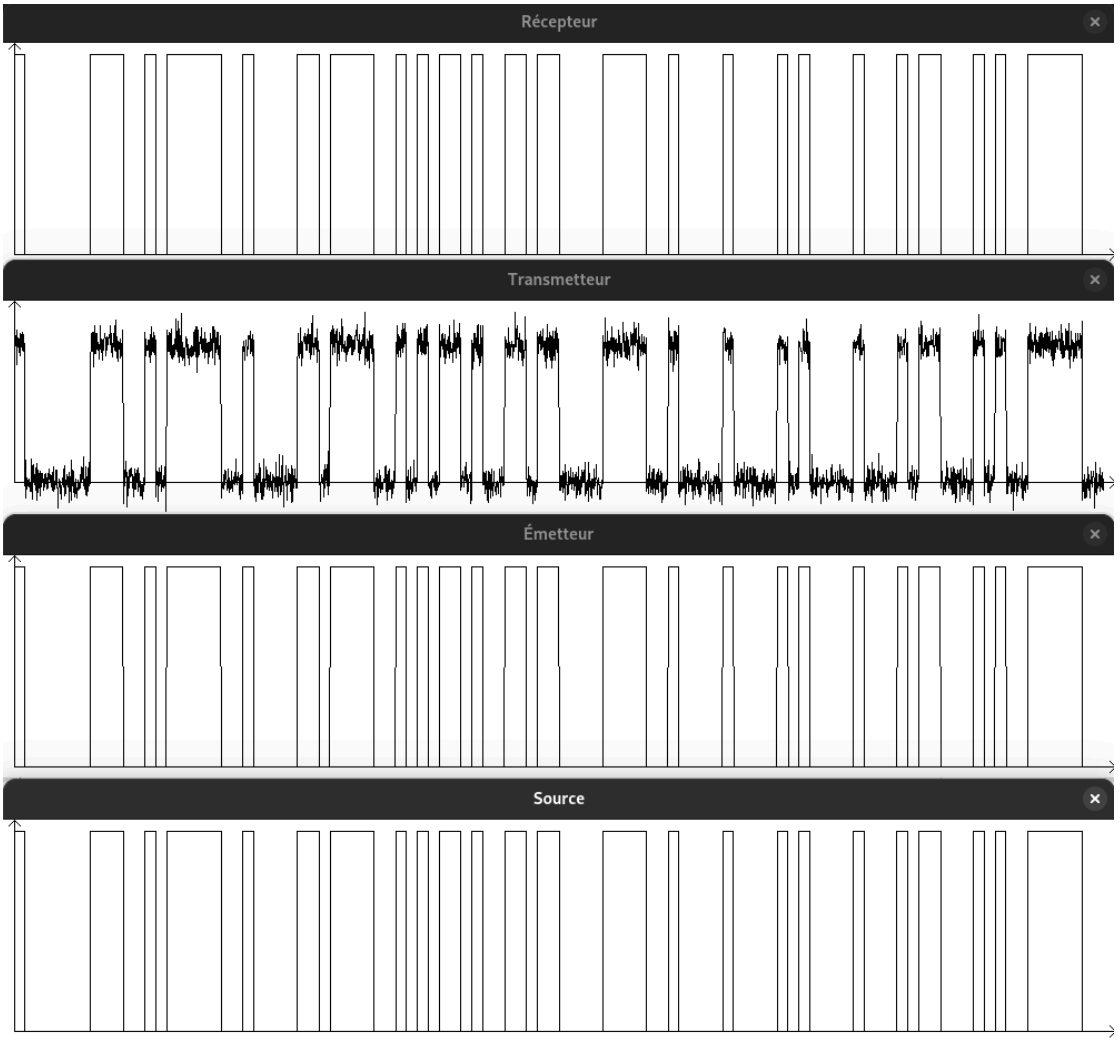
4.3 Analyse des résultats

Afin de valider le fonctionnement du Simulateur avec le bruit, nous avons ajouté des mesures dans le but de vérifier les résultats avec ceux donnés ("Exemples de signaux bruités"). Il faut mettre afficherInformations = true.
Nos simulations ont globalement les mêmes résultats que celles fournies, ici le paramètre snrpb était configure pour le SNR global et pas Eb/N0 comme actuellement, les simualtions restent les mêmes :

Pour un SNR de 20 dB, nous retrouvons les valeurs fournies

```
- Nombre de bits de la séquence : 100
- Nombre d'échantillons par bit : 30
1-> Puissance MOYENNE de la séquence de bits : 0.51
2-> Valeur de sigma (écart-type du bruit) : 0.0714
3-> Puissance moyenne du bruit : 0.0051
4-> Rapport signal-sur-bruit (S/N, en dB) : 19.9879
5-> Rapport Eb/N0 (en dB) : 31.7489
```

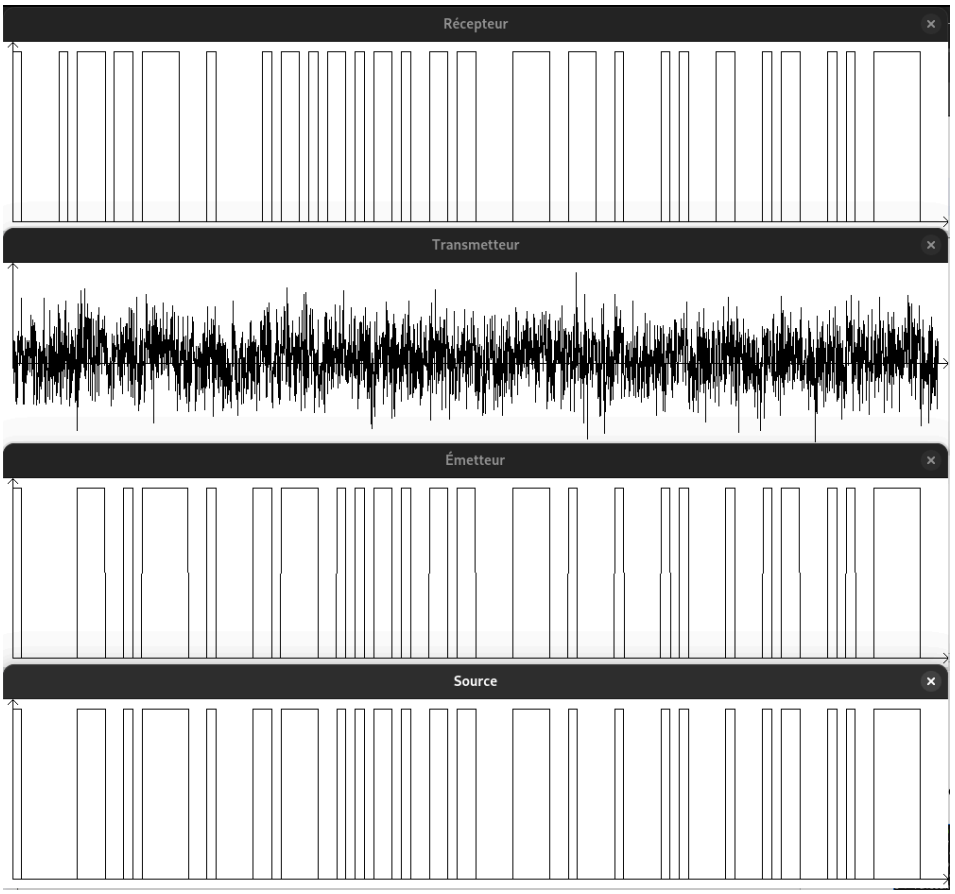
```
~/sit213 main !1 ❯ ./simulateur -s -mess 100 -form NRZ -seed 1 -nbEch 30 -ampl 0.0 1.0 -snrpb 20
- Nombre de bits de la séquence : 100
- Nombre d'échantillons par bit : 30
1-> Puissance MOYENNE de la séquence de bits : 0.45
2-> Valeur de sigma (écart-type du bruit) : 0.0667501392915691
3-> Puissance moyenne du bruit : 0.004455581095443877
4-> Rapport signal-sur-bruit (S/N, en dB) : 20.043081611628907
5-> Rapport Eb/N0 (en dB) : 31.80399420218572
java Simulateur -s  -mess  100  -form  NRZ  -seed  1  -nbEch  30  -ampl  0.0  1.0  -snrpb  20    =>   TEB : 0.0
```



Pour -10 dB. les résultats sont toujours similaires :

```
- Nombre de bits de la séquence : 100
- Nombre d'échantillons par bit : 30
1-> Puissance MOYENNE de la séquence de bits : 0.45999999999999996
2-> Valeur de sigma (écart-type du bruit) : 2.1448
3-> Puissance moyenne du bruit : 4.6261
4-> Rapport signal-sur-bruit (S/N, en dB) : -10.0245
5-> Rapport Eb/N0 (en dB) : 1.7364
```

```
~/sit213 main !1 ❯ ./simulateur -s -mess 100 -form NRZ -seed 1 -nbEch 30 -ampl 0.0 1.0 -snrpb -10
- Nombre de bits de la séquence : 100
- Nombre d'échantillons par bit : 30
1-> Puissance MOYENNE de la séquence de bits : 0.45
2-> Valeur de sigma (écart-type du bruit) : 2.1428632359207036
3-> Puissance moyenne du bruit : 4.591862847860549
4-> Rapport signal-sur-bruit (S/N, en dB) : -10.087763940756174
5-> Rapport Eb/N0 (en dB) : 1.6731486498006394
java Simulateur -s  -mess  100  -form  NRZ  -seed  1  -nbEch  30  -ampl  0.0  1.0  -snrpb  -10    =>   TEB : 0.09
```



Les résultats obtenus lors de la simulation sont en accord avec les théories de transmission sur un canal bruité :

- L'analyse du TEB en fonction du SNR montre que les performances des différentes modulations se détériorent progressivement à mesure que le bruit augmente
- L'histogramme du bruit confirme la nature gaussienne du bruit généré, assurant ainsi la validité des conditions de transmission simulées
- Les valeurs mesurées sont les mêmes que théoriques et fournies.

Conclusion

L'étape 3 du projet a permis de valider l'implémentation des composants de transmission analogique via un canal bruité. À travers cette simulation, nous avons démontré que la chaîne de transmission réagit comme prévu face aux perturbations, notamment avec l'ajout d'un bruit blanc additif gaussien. Les tests ont montré que le Taux d'Erreur Binaire varie en fonction du rapport signal sur bruit, confirmant que le bruit affecte la qualité de la transmission comme attendu.

L'intégration du module TransmetteurAnalogiqueBruite a été réalisée avec succès et son fonctionnement a été testé de manière approfondie dans des environnements bruités. Cette étape constitue une avancée significative dans la simulation de scénarios plus réalistes, où le bruit et les interférences influencent la transmission des données. Les résultats obtenus démontrent la robustesse du système face aux perturbations, tout en ouvrant la voie à l'exploration d'autres types de bruit ou à l'introduction de mécanismes de correction d'erreurs pour améliorer encore les performances du système dans des environnements non parfaits.