

|  |
| --- |
| SIT213 Etape 3  IMT Atlantique |
| Chaire M@D - Maintien à domicile | IMT Atlantique |
| 20 septembre  Créé par :  LE DUC Elouan MAQUIN Philippe LE GRUIEC Clément LE JEUNE Matthieu FRAIGNAC Guillaume |

Table des matières

[Intentions du projet 3](#_Toc50887207)

[3ème étape du projet 4](#_Toc50887208)

[ Objectifs 4](#_Toc50887209)

[ Analyse des actions à mener 5](#_Toc50887210)

[ Tests et validations du programme 6](#_Toc50887211)

[ Résultats attendus Erreur ! Signet non défini.](#_Toc50887212)

[ Tests avec signal de type NRZ Erreur ! Signet non défini.](#_Toc50887213)

[ Tests avec signal de type NRZT Erreur ! Signet non défini.](#_Toc50887214)

[ Tests avec signal de type RZ Erreur ! Signet non défini.](#_Toc50887215)

[Conclusion 6](#_Toc50887216)

[Table des illustrations 7](#_Toc50887217)

# Intentions du projet

Il s’agit de réaliser, par équipe de 4 ou 5 élèves, une maquette logicielle (en Java) simulant un système de transmission numérique élémentaire. On intégrera donc dans la chaîne un bloc de modulation numérique.

Le système sera assemblé suivant une bibliothèque de modules comportant des ports d’entrée, des ports de sortie et des paramètres physiques. Ces derniers pourront être déterminés à partir des activités du module SIT 212. Le système global sera mis au point progressivement sur 5 séances au cours desquelles les modules seront raffinés, complétés, validés et connectés selon un schéma de transmission de type « point-à-point ».

Outre la qualité technique de la réalisation, on insistera sur les points suivants :

1. La qualité de documentation de la maquette logicielle (notamment la Javadoc).
2. Les efforts de validation des résultats de simulation produits par la maquette.
3. La maîtrise du processus de travail : gestion des versions successives de la maquette logicielle et du dossier technique afférent, synergie de l'équipe, démarche qualité. Concernant ce tout dernier critère, le respect des exigences de mise en forme du livrable sera primordial.

# 3ème étape du projet

## Objectifs

Dans cette phase du projet nous allons travailler sur une transmission non-idéale avec canal bruité de type « gaussien » d’un signal analogique. La propagation dans le canal est modélisée de manière théorique par un bruit blanc additif gaussien qu’il conviendra de régler en fonction des paramètres du transmetteur vus à l’étape précédente. On fera attention à surveiller le TEB en réception.



**Figure 1 Modélisation de la chaîne de transmission à l’étape 3.**

Par défaut le simulateur doit utiliser une chaîne de transmission logique, avec un message aléatoire de longueur 100, sans utilisation de sondes et sans utilisation de transducteur.

En plus des paramètres des deux premières étapes on va ici ajouter un bruit gaussien.

L’option **-snrpb s** en transmission analogique bruitée s permet de donner la valeur du rapport signal sur bruit par bit (*Eb/N0* en Db). Nous ferrons attention à utiliser une paramètre flottant.

## Analyse des actions à mener

Nous allons devoir ajouter un bruit blanc gaussien dans le simulateur de signaux analogiques que nous avons développé au cours de la seconde étape. Pour générer le bruit blanc gaussien on utilise la formule suivante :



Par la suite nous allons additionner le signal analogique créer à la précédente étape au bruit blanc gaussien. Comme l’illustre le schéma ci-dessous.



**Figure 2 Ajout du bruit blanc gaussien.**

On obtient ainsi le signal r(n) qui correspond au signal d’entre additionner au bruit blanc. Il faudra vérifier que le bruit généré suit une loi gaussienne (histogramme)

Comme pour l’étape précédente nous retrouvons les 3 fromes à prendre e compte, à savoir :

* NRZ
* NRZT
* RZ

D’après le schéma de la figure 1 nous devons modifier notre Transmetteur analogique « parfait » en un transmetteur analogique « bruitée gaussien ». Les émetteurs, récepteur, source et destination sont les mêmes qu’a la précédente étape. Pour ce qui concerne les sondes pour visualiser les signaux, elles ont déjà été programmées par les enseignants.

Nous devons implémenter le paramètre suivant : -snrpb s. Nous devrons nous assurer de détecter les paramètres du programme et d’assurer de leur conformité en utilisant des Regex par exemple.

## Tests et validations du programme

A partir de la troisième étape nous avons commencé à automatiser nos tests. Ces différents tests ont pour but de vérifier si l’ensemble du programme est correctement installé sur une machine.

# ✓ Résultats attendus

# Nous sommes dans le cas d’une chaine de transmission avec un transmetteur analogique bruité gaussien. Le signal émis doit être identique au signal reçu qu’il soit converti sous une forme NRZ, NRZT ou encore RZ

# Sur les sondes les signaux sources et destinations sont identiques, les signaux en sorti de l’émetteur et du transmetteur sont également identiques. Dans tous les cas le TEB doit être à 0 à la fin de la chaine de transmission si tout se passe bien. Les formes d’ondes choisies doivent être utilisées selon leur spécifications.

# Pour les tests nous utiliserons un signal avec les paramètres suivants :

# - Amplitude max : 5V ;

# - Amplitude min : -5V ;

# - Nb échantillon : 60 ;

# - Seed : 40 ;

# - Longueur (mess) : 20 ;

# - Snrpb : 1.

# Le signal aléatoire généré sera égal à 1101 0011 0100 1001 1111 Ce message est intéressant, il permet de tester différents cas de figure, par exemple deux 1 qui se suive, le passage d’un 0 à un 1 etc… (On utilise le même signal que pour la seconde étape)

# 

# ✓ Tests avec signal de type NRZ

# 

# Exemple d’histogramme obtenue à partir du résultat de 100K simulation et un Snrpb à -20Db

# 

# *Figure 3 Tests du signal de type NRZ*

# « java Simulateur -s -mess 20 -seed 40 -form NRZ -ampl -5 5 -snrpb 1 => TEB : 0.0 »

# Le signal émis correspond bien au signal reçu. Le TEB le confirme et est bien égal à 0. On peut également observer que le bruit ajouter dans transmission est bien décomposer à la réception.

# ✓ Tests avec signal de type NRZT

# 

# Exemple d’histogramme obtenue à partir du résultat de 100K simulation et un Snrpb à -20Db

# 

# *Figure 4 Tests du signal de type NRZT*

# « java Simulateur -s -mess 20 -seed 40 -form NRZT -ampl -5 5 -snrpb 1 => TEB : 0.0 »

# Le signal émis correspond bien au signal reçu. Le TEB le confirme et est bien égal à 0. Tout comme le précèdent signal le bruit à bien été traité.

# ✓ Tests avec signal de type RZ

# 

# Exemple d’histogramme obtenue à partir du résultat de 100K simulation et un Snrpb à -20Db

# 

# *Figure 5 Tests du signal de type NRZ*

# « java Simulateur -s -mess 20 -seed 40 -form RZ -ampl -5 5 -snrpb 1 => TEB : 0.0 »

# Le signal émis correspond bien au signal reçu. Le TEB le confirme et est bien égal à 0. Tout comme le précèdent signal le bruit à bien été traité.

# Conclusion

# Nous avons pu simuler un transmetteur analogique bruitée. Les tests se sont passés comme prévus, les résultats sont donc satisfaisants.

# Dans la prochaine étape nous allons devoir développer une transmission non-idéale avec divers bruits « réels » qui viendront dans un premier temps remplacer le bruit gaussien. Ensuite nous feront l’addition de c’est deux bruits.

# Table des illustrations

[Figure 1 Modélisation de la chaîne de transmission à l’étape 3. Erreur ! Signet non défini.](#_Toc50887442)

[Figure 2 Ajout du bruit blanc gaussien. Erreur ! Signet non défini.](#_Toc50887443)

[Figure 3 Tests du signal de type NRZ Erreur ! Signet non défini.](#_Toc50887446)

[Figure 4 Tests du signal de type NRZT Erreur ! Signet non défini.](#_Toc50887447)

[Figure 5 Tests du signal de type RZ Erreur ! Signet non défini.](#_Toc50887448)