Simulation d’un système de transmission – SIT 213

Auteurs :

- BARTOLI Mathieu
- DUMESTRE Lucas
- FRANCIS Ludovic
- GUEYE Oulimata
- HUG DE LARAUZE Sébastien

FIP2A

2019

TABLE DES MATIERES

BARTOLI Mathieu
3
DUMESTRE Lucas
FRANCIS Ludovic
HUG DE LARAUZE Sébastien

I. INTRODUCTION

Nous avons décidé de débuter le projet en groupe de 4 plutôt qu'en binôme. En effet, la première étape du projet était supposée être effectuée séparément par les binômes, avant qu'ils ne se rejoignent pour former un groupe de 4 pour le reste du projet. Nous avons jugé nécessaire d'avoir plus de temps pour la phase de regroupement afin de pouvoir déployer une infrastructure complexe autours du projet. Cette infrastructure est détaillée dans le Rapport d'Organisation.

L'objectif de ce projet est de réunir nos connaissances acquises en modélisation et validation de logiciels ainsi qu'en simulation de signaux et briques de transmission afin de réaliser une maquette logicielle simulant un système de transmission (cf. Figure 1).

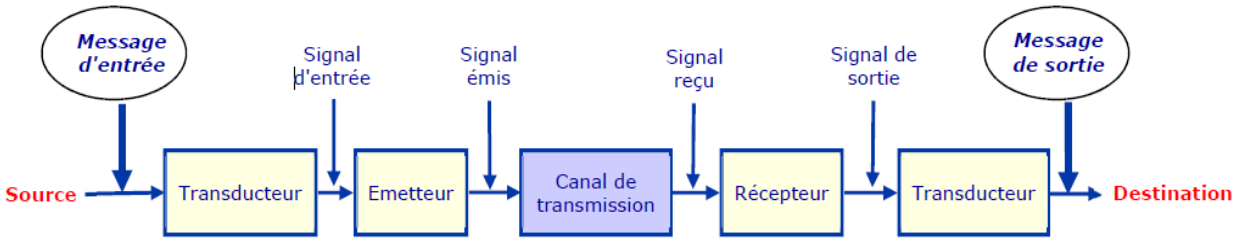

Figure 1 : Système de transmission intégral

Le système complet sera mis au point progressivement sur 5 séances. Dans sa version finale, le système devra être capable de transmettre un message d'un point d'entrée à un point de sortie, en prenant en compte les caractéristiques physiques du canal de transmission. Ce message binaire sera émis par une source (fixe ou aléatoire), puis converti par un transducteur en signal, qui sera injecté dans le canal par un émetteur. Le signal sera ensuite récupéré et traité par le récepteur et le transducteur de réception.

L'étape 1 du projet qui a été effectuée le 13/09/2019 avait pour but de réaliser un système de transmission élémentaire « back-to-back ».

L'étape 2 du projet qui sera effectuée prochainement aura pour but d'effectuer une transmission non bruitée d'un signal analogique.

L'étape 3 du projet qui sera effectuée prochainement aura pour but d'ajouter un bruit blanc gaussien au signal afin de simuler des perturbations qui peuvent survenir sur le canal de transmission.

L'étape 4 du projet qui sera effectuée prochainement aura pour but d'ajouter un phénomène de perturbation que nous retrouvons dans la nature : les trajets multiples qui perturbent le signal en réception.

L'étape 5 du projet qui sera effectuée prochainement aura pour but d'ajouter un codage canal pour introduire de la redondance et ainsi réduire le nombre d'erreurs.

BARTOLI Mathieu
4
DUMESTRE Lucas
FRANCIS Ludovic
HUG DE LARAUZE Sébastien

II. ORGANISATION DU TRAVAIL EN EQUIPE (BE1)

cf. Rapport d'Organisation

III. ETAPE 1: TRANSMISSION ELEMENTAIRE "BACK TO BACK"

La première étape du projet consiste à mettre en place un système de transmission basique, composé uniquement d'une source et d'une destination reliées par un transmetteur logique supposé parfait. Ce système est schématisé par la figure 2 :

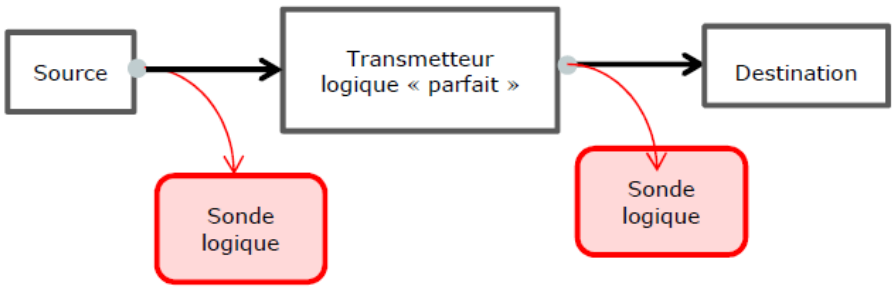

Figure 2: Système de transmission à l'étape 1

La source émet une séquence booléenne fixe ou aléatoire, qui est ensuite relayée par le transmetteur parfait qui se contente de la transmettre vers la destination connectée.

La destination se contente de recevoir le signal du composant auquel elle est connectée. Entre chaque composant, des sondes logiques permettent de visualiser les signaux émis. L'application quant à elle calcule le taux d'erreur binaire (TEB) du système.

La figure 3 reprend le résultat d'une simulation du système réalisé :

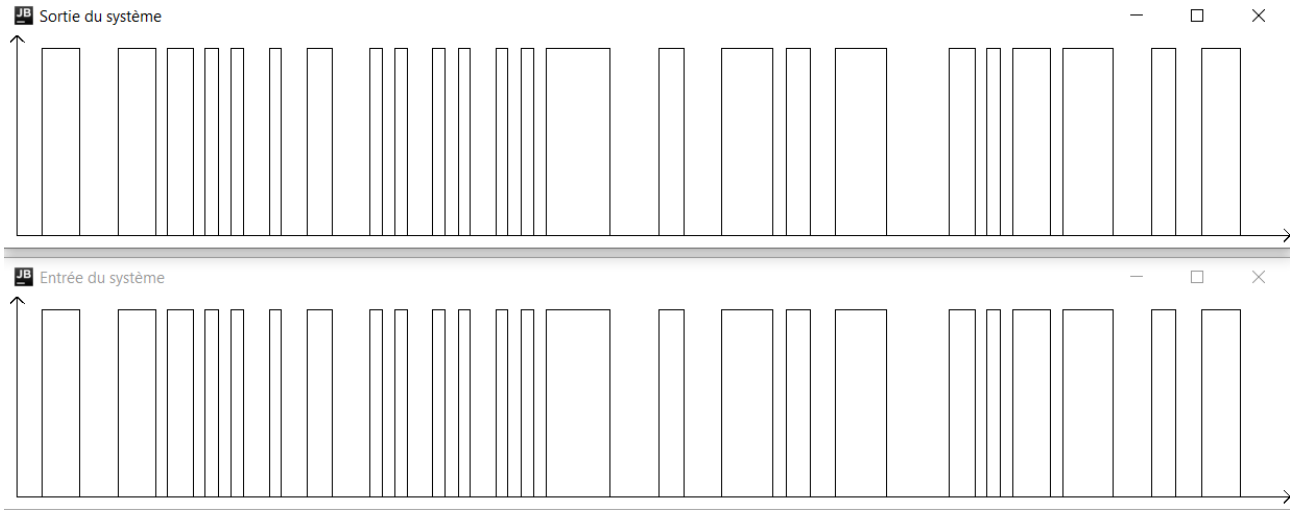

Figure 3: Simulation d'une transmission

Comme on peut s'y attendre pour une transmission supposée parfaite, le Taux d'Erreur Binaire (TEB) est nul : les données reçues sont strictement identiques aux données émises.

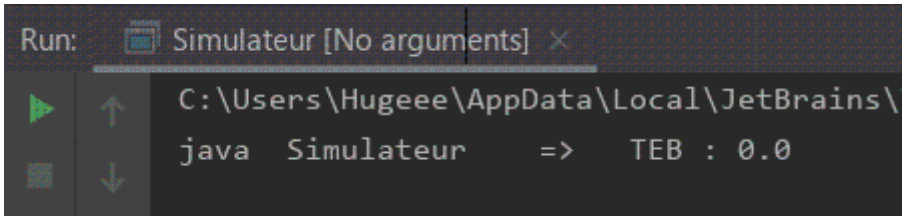

Figure 4: Taux d'erreur binaire à l'étape 1

Au cours de cette étape, nous avons utilisé le code fourni pour réaliser le système de transmission. Nous avons ainsi lié entre elles différentes classes :

- Source
- Transmetteur
- Destination
- Sonde

Aux classes existantes ont été ajoutées plusieurs classes :

- SourceFixe (hérite de Source)
- SourceAleatoire (hérite de Source)
- TransmetteurParfait (hérite de Transmetteur)
- DestinationFinale (hérite de Destination)

Une fois ces éléments liés, nous avons implémenté certains éléments de la commande unique :

- **- mess m** : permet de générer un message aléatoire de taille m

- **- s :** permet d'ajouter des sondes pour visualiser le message transmis en certains points du système
- **-seed s :** permet d'ajouter une germe s qui va influer sur l'aléa généré par la source. Si la germe est identique entre deux simulations, la séquence aléatoire sera identique

L'étape 1 nous a permis de comprendre la logique derrière un système de transmission basique, et ainsi de mieux appréhender les futurs ajouts à ce système, tels que la simplification de nos processus de travail pour optimiser le temps passé sur chaque tâche. Par ailleurs, l'arrivée d'un nouveau membre dans l'équipe projet va nous permettre de revoir notre façon de répartir les taches au sein du groupe. L'ajout d'un émetteur et d'un récepteur à l'étape 2 nous permettra aussi de passer à une transmission analogique.

IV. ETAPE 2 : TRANSMISSION NON-BRUITEE D'UN SIGNAL ANALOGIQUE

La seconde étape du projet consiste à mettre en place un système de transmission plus complexe. En effet, nous y ajoutons un émetteur et un récepteur. De ce fait, le canal de transmission devient analogique, comme on peut l'observer sur la Figure 5 :

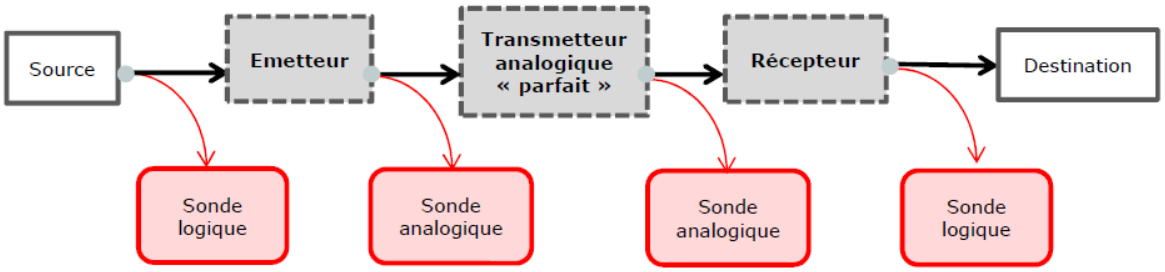

Figure 5 : Système de transmission à l'étape 2

En ajoutant un émetteur, on rend possible une conversion de signal logique à signal analogique. Le récepteur assure la fonction inverse en convertissant le signal analogique transmis par le canal analogique parfait en signal logique.

Le signal numérique émis par la source est composé uniquement de 0 et de 1, puis converti en signal analogique par l'émetteur. Pour cela, l'émetteur va choisir un codage. Nous avons implémenté 3 codages différents :

- RZ : forme d'onde impulsionnelle (cf. figure 6)
- NRZ : forme d'onde impulsionnelle : le premier tier du symbole a une amplitude minimale, puis on trouve une impulsion d'amplitude maximale sur le second tier, suivie d'un dernier tier à l'amplitude minimale (cf. figure 7)
- NRZT : forme d'onde trapézoïdale : temps de monté ou descente sur le premier et dernier tier du symbole

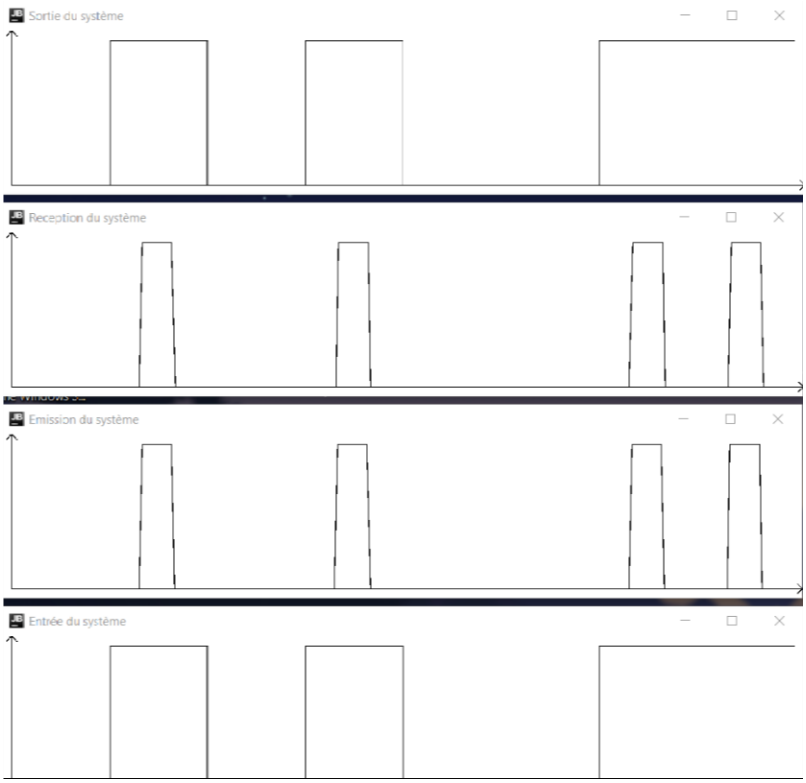

Figure 6 : Simulation du signal avec le codage RZ

BARTOLI Mathieu
8
DUMESTRE Lucas
FRANCIS Ludovic
HUG DE LARAUZE Sébastien

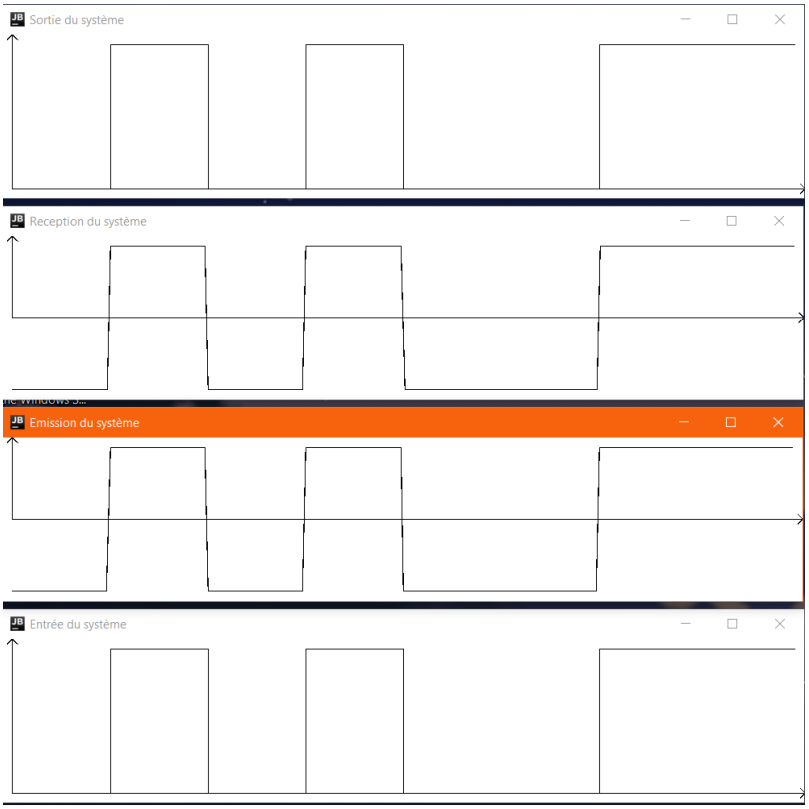

Figure 7 : Simulation du signal avec codage NRZ

Chaque codage a été pris en compte lors de la création de l'émetteur. Par ailleurs, le simulateur prend en paramètres les caractéristiques du signal qui doit être crée :

- Amplitude (minimum et maximum)
- Nombre d'échantillons
- Codage à utiliser

Ces trois paramètres sont initialisés grâce aux options du simulateur qui les prend désormais en charge et va générer un signal à partir de ces données. Comme prévu, le TEB de toutes les transmissions est nul. En effet, il s'agit d'une transmission parfaite : il n'y a pas de bruit, donc le signal reçu est identique au signal émis.

D'un point de vue développement logiciel, nous avons débuté cette seconde étape en modifiant le transmetteur : il a été généraliser afin d'être indépendant du type passé à l'objet. Suite à la répartition des tâches, les personnes concernées ont démarré le développement des fonctionnalité émission/réception. Nous avons aussi ajouté des sondes analogiques, afin de pouvoir observer l'évolution du signal respectivement après et avant les blocs émetteur et récepteur. Pour implémenter les différents types de codage, nous avons choisi de créer un nouveau package « Encoder » contenant une classe abstraite Encoder et une classe qui en hérite pour chaque type de codage. Ce package nous permet de compartimenter une fonctionnalité : le codage (et donc décoder) des symboles en analogique. Après implémentation des nouvelles fonctionnalités, la structure du projet correspond à la figure 8 :

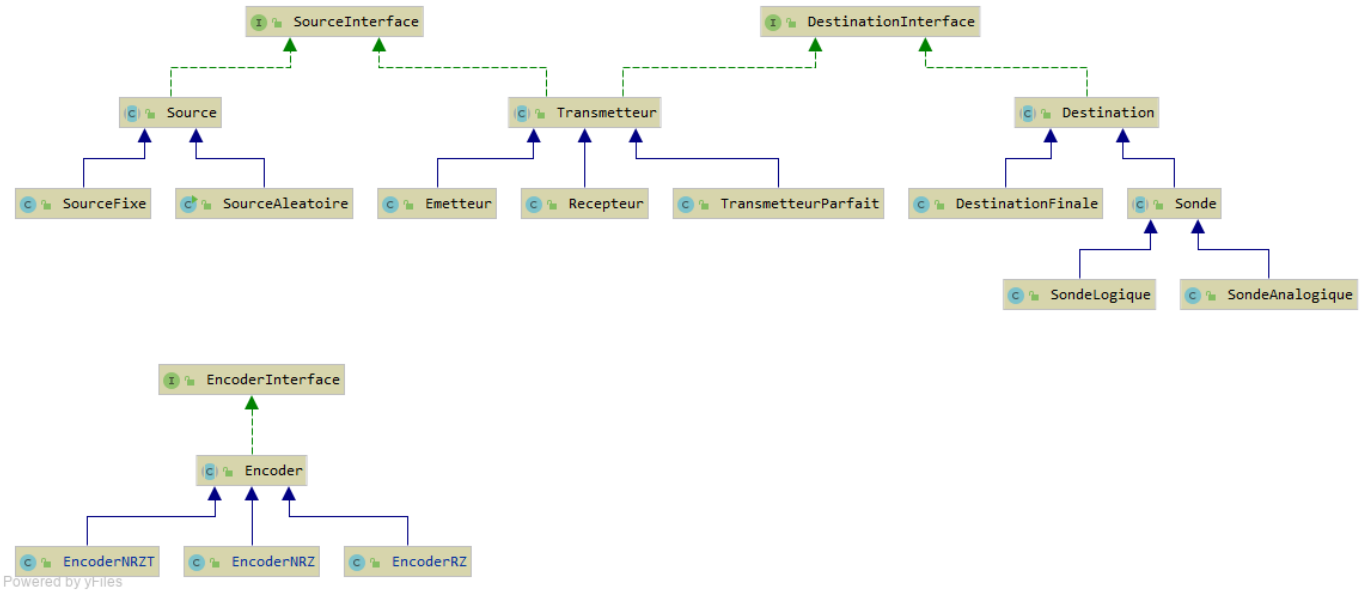

Figure 8 : Diagramme UML à l"étape 2 du projet

Nous avons aussi amélioré le système permettant de récupérer les options passées au simulateur. Pour cela, nous utilisons la librairie common.cli.

Commentaires :

Au cours de cette seconde phase de projet, nous avons eu des retards dans le développement, ce qui a nous a empêcher finaliser l'intégration du codage NRZT. Les autres types de codage ont été implémentés et testés, mais d'après les premiers tests, le NRZT n'est pas fonctionnel et ne le sera pas avant le début de l'étape 3.

Ce retard dans le développement est dû à une mauvaise organisation dans l'équipe. La collaboration entre les 5 membres n'a pas fonctionné comme prévu, ce qui a entrainé des difficultés dans l'implémentation de nouvelles fonctionnalités. Le retard sur le code a aussi influé sur l'heure de remise du rapport. Nous allons corriger ce problème d'organisation dès le début de l'étape 3 du projet en redistribuant les rôles au sein de l'équipe projet, en prenant en compte les capacités de chacun et leur attitude face à la charge de travail au cours des premières étapes.

A l'heure actuelle, nous n'avons pas observé de différence entre les types de codage. En effet, les TEB sont tous nuls car nous sommes dans le cas d'une transmission parfaite. Il sera intéressant d'observer les différences entre les signaux reçu pour chaque type de codage lorsque le signal sera soumis à un bruit. Il s'agit d'ailleurs de l'évolution logique du projet à l'étape 3.

BARTOLI Mathieu
10
DUMESTRE Lucas
FRANCIS Ludovic
HUG DE LARAUZE Sébastien