

Projet de Communication Numérique

2022

Simulation de chaine de transmission avec python

Réalisé par :

OTHMANE ELAZRI
ANASS EL HALLANI
YOUNES EL BELGHITI

Encadrée par :

Pr.JILBAB ABDELILAH

Résumé

Sachant que nous voulons optimiser la chaîne de transmission, et à cause de la perte de temps à recalculer et tester les coefficients de la chaîne pour chaque cas toute seul. pour cette raison, le simulateur a été créé pour faciliter et automatiser les tâches de calcul.

Pour mettre en oeuvre ce projet de simulation de chaîne de transmission en python, l'interface est constitue du blocs, le premier bloc s'intéresse au séquence binaire à envoyée au récepteur, premierement cette séquence est traitée par un codeur en ligne, qui réorganise cette séquence pour avoir une grande énergie et facilite la reconstruction de l'horloge à l'extrémité de réception, qui passe ensuite au filtre d'émission, qui nous aide à éliminer les interférences inter-symboles (IES) à l'aide d'un filtre blanchissant, après tout ce traitement, nous atteignons la dernière partie de l'émetteur, la modulation qui déplace un signal d'une fréquence basse à une fréquence porteuse pour faciliter la transmission dans un canal (air, câble, fibre, etc.).

Maintenant après que notre information arrive à notre récepteur leur première station de passage c'est le démodulateur qui redéplace notre information du fréquence porteuse a une domaine fréquentiel basse, Donc, à ce point, le signal s'accumule sur le signal de sortie du filtre d'émission, mais avec le bruit provenant du canal, nous avons donc besoin du filtre de réception pour obtenir un signal plus clair pour le traitement suivant, le dernier bloc de décision contient deux partie, la première est pour la reconstruction d'horloge qui donne des impulsions à chaque T_s . la deuxième partie de décision donne la séquence binaire reçus .

Sommaire

1	Introduction	5
2	LA CHAINE DE TRANSMISSION NUMÉRIQUE	6
2.1	L'émetteur	7
2.1.1	Séquence binaire émis	7
2.1.2	Codeur en ligne	7
2.1.3	Filtre d'émission	13
2.1.4	Modulation Numérique	15
2.2	Canal de transmission	17
2.3	Récepteur	18
2.3.1	Démodulation	18
2.3.2	Démodulateur ASK :	18
2.3.3	Démodulateur BPSK :	19
2.3.4	Démodulateur QPSK :	19
2.3.5	Filtre de réception	21
2.3.6	Prise de décision :	22
2.3.7	Taux d'erreurs binaire TEB	22
3	Software utiliser pour le simulation	23
3.1	Langage de programmation	23
3.1.1	Python	23
3.1.2	Microsoft Visual Studio code	24
3.1.3	Numpy	24
3.1.4	PyQt5	25
3.1.5	Matplotlib	25
4	Experimentation et discussions des résultats	26
4.1	La page d'accueil	26
4.2	La page de chaine	27
4.3	Séquence binaire emis	27
4.4	Code en ligne	28
4.4.1	*Coudeur RZ	28

4.4.2	*Codeur NRZ	28
4.4.3	*Codeur Miller	29
4.4.4	*Codeur Manchester	29
4.4.5	*Codeur HDBN cas de N=3	30
4.5	Filtre d'émission	30
4.5.1	Filtre d'émission avec le coefficient de retombé $\alpha=0$	30
4.5.2	Filtre d'émission avec le coefficient de retombé $\alpha=0.5$	31
4.5.3	Filtre d'émission avec le coefficient de retombé $\alpha=1$	31
4.6	modulation	32
4.6.1	*Modulation ASK	32
4.6.2	*Modulation PSK	32
4.6.3	*Modulation FSK	33
4.7	Canal de Propagation(avec Bruit)	33
4.7.1	Bruit avec varinace=0.1	33
4.7.2	Bruit avec varinace=0.3	34
4.7.3	Bruit avec varinace=1	34
4.8	Demodulation	35
4.8.1	Demodulation adapter au modulation	35
4.9	Filtre de Reception	35
4.10	décision	36
4.11	Horloge	36
4.12	Séquence binaire reçu	37
4.12.1	Conclusion :	37

Table des figures

2.1	CHAINE DE TRANSMISSION NUMÉRIQUE	6
2.2	Data viens du CAN	7
2.3	Sortie de codeur NRZ	8
2.4	DSP de NRZ	8
2.5	Sortie de codeur RZ	9
2.6	DSP de RZ	9
2.7	Sortie de codeur Miller	10
2.8	DSP de Miller	10
2.9	Sortie de codeur Manchester	11
2.10	DSP de Manchester	12
2.11	Sortie de codeur HDBn	12
2.12	DSP de HDBn	13
2.13	Filtre blanchissant	13
2.14	Critère de Nyquist temporel et fréquentiel	14
2.15	Chaine de transmtion (data, codeur NRZ, filtre blanchissant et filtre de Nyquist)	15
2.16	Diagramme de l'œil	15
2.17	Modulation ASK	16
2.18	Modulation PSK	16
2.19	Modulation FSK	17
2.20	canal de transmsion	18
2.21	ASK Demodulation	19
2.22	BPSK Demodulation	19
2.23	QPSK Demodulation	20
2.24	PSK Demodulation	20
2.25	21
2.26	FSK Demodulation	21
3.1	Python	23
3.2	Microsoft Visual Studio code	24
3.3	Numpy	24
3.4	PyQt5	25

3.5	Matplotlib	25
4.1	La page d'accueil	26
4.2	Page de chaine	27
4.3	Séquence binaire emise	27
4.4	Codeur en ligne RZ	28
4.5	Codeur en ligne NRZ	28
4.6	codeur en ligne Miller	29
4.7	Codeur en ligne Manchester	29
4.8	Codeur en ligne HDB3	30
4.9	Filtre d'émission $\alpha=0$	30
4.10	Filtre d'émission $\alpha=0.5$	31
4.11	Filtre d'émission $\alpha=1$	31
4.12	Modulation ASK	32
4.13	Modulation PSK	32
4.14	Modulation FSK	33
4.15	Canal de Propagation(avec Bruit) $\text{varinace}=0.1$	33
4.16	Canal de Propagation(avec Bruit) $\text{varinace}=0.3$	34
4.17	Canal de Propagation(avec Bruit) $\text{varinace}=1$	34
4.18	Demodulation	35
4.19	Filtre de Reception	35
4.20	décision	36
4.21	Horloge	36
4.22	Séquence binaire recue	37

Chapitre 1

Introduction

Historiquement, la transmission analogique a dominé l'industrie des télécommunications. Aujourd'hui, les ordinateurs sont présents dans la plupart des endroits.

Les réseaux de télécommunication ont entraîné de profonds changements dans la technologie de transmission et ont évolué vers la transmission numérique.

L'échange d'informations entre deux ou plusieurs interlocuteurs équivaut à la transmission d'informations. Les systèmes de communication modernes nécessitent des débits de données toujours plus élevés pour traiter des quantités croissantes d'informations. Aux contraintes liées au traitement de l'information, il doit également être protégé lorsqu'il est envoyé dans des environnements perturbés tels que des erreurs de canal de transmission. En effet, la qualité du service rendu en matière de communication se résume à deux mots clés : fiabilité et rapidité.

La qualité peut être améliorée en augmentant la puissance de transmission et réduire le bruit. Malheureusement, cette solution implique des coûts énergétiques et des technologies importantes, ce qui limite fortement leur utilisation.

Notre projet est composée de quatre principaux chapitres. Tout d'abord, nous allons exposer quelques rappels et définitions sur les différents étapes d'une chaîne de transmission numérique tel que les codages source et canal, la modulation, démodulation...

Ensuite, nous allons nous intéresser à la réalisation d'une interface graphique utilisant le langage python.

L'objectif principal de ce travail est faire une simulation d'une chaîne de transmission numérique.

Chapitre 2

LA CHAÎNE DE TRANSMISSION NUMÉRIQUE

Les systèmes de transmission numérique envoient des informations de émetteur à récepteur via un support physique tel que le câble, la fibre optique ou même Propagé par canal radio. Le signal transmis peut être directement des sources numériques, telles que les réseaux de données, ou des sources analogiques (voix, image..) mais converti en format numérique. les tâches d'un système de transmission est le passage d'informations de l'émetteur au récepteur Plus de fiabilité possible. Schéma fonctionnel du système de transmission numérique est donne a la figure ci-dessous.

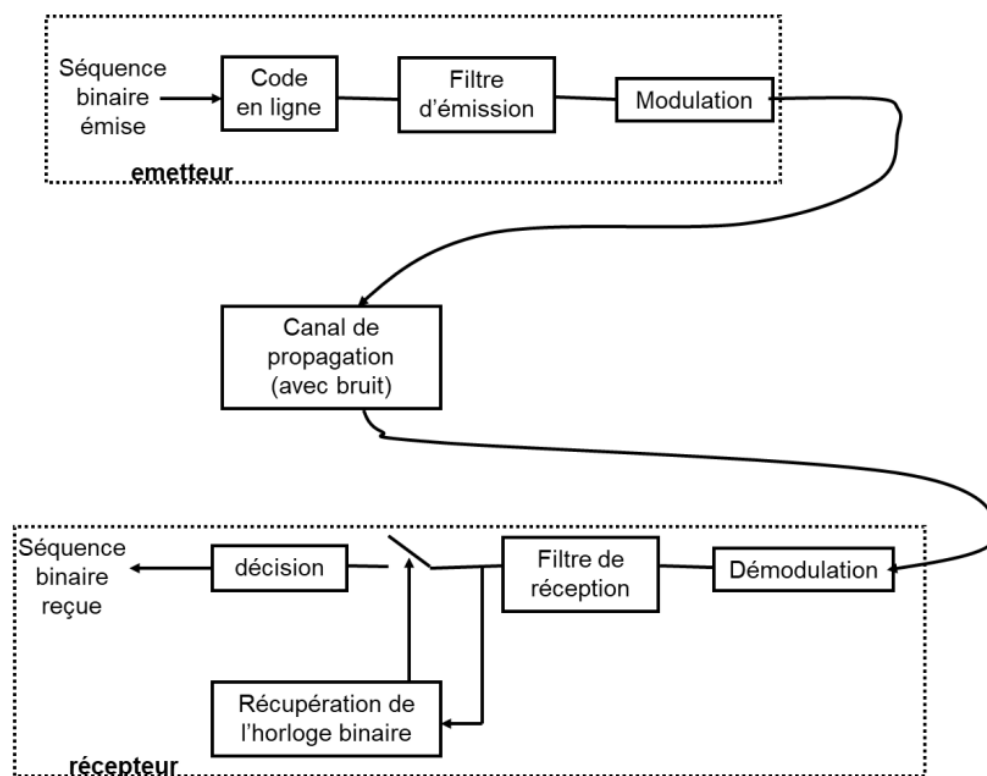


FIGURE 2.1 – CHAÎNE DE TRANSMISSION NUMÉRIQUE

2.1 L'émetteur

2.1.1 Séquence binaire émis

Un groupe de k bits forme un symbole appartenant à un ensemble finit de symboles, appelé alphabet, formé de $M = 2^k$ symboles. Un système utilisant un alphabet de dimension M est dite système M -aire. On suppose que les données sont émises périodiquement.

On note T_b la durée d'un bit et T la durée d'un symbole. Dans ce cas, la durée d'un symbole T est égale à k fois la durée d'un bit, $T = k.T_b$.

Le débit des symboles est appelé débit baud, ou vitesse de modulation.

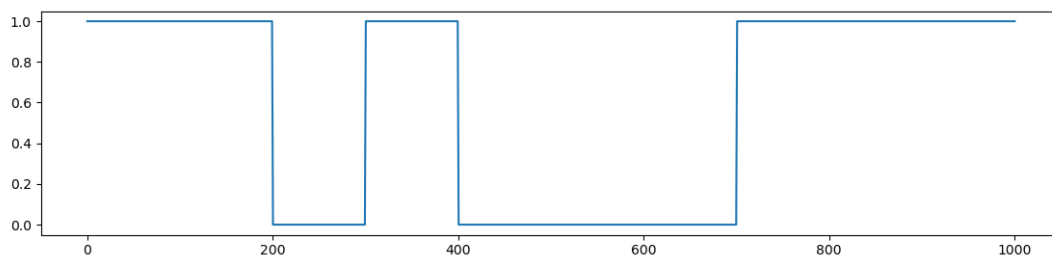


FIGURE 2.2 – Data viens du CAN

2.1.2 Codeur en ligne

Un code en ligne est un codage destiné à être utilisé dans les systèmes de communication pour transmettre des données.

Pour le transport de données numériques, le codage en ligne est souvent utilisé. Il consiste à représenter le signal numérisé à transporter, par un autre signal qui présente des variations d'amplitude régulièrement espacées dans le temps, celui-ci étant adapté aux propriétés physiques spécifiques des canaux de transmissions (et des équipements récepteurs). Le codage du signal est utilisé pour représenter les 1 et les 0 d'un signal numérique sur le lien, ce processus est appelé codage en ligne. Après le codage en ligne, le signal peut être directement émis sur le canal de transmission, sous la forme de variations de la tension ou du courant. Les types de codage en ligne les plus utilisés sont NRZ, RZ, Manchester, Miller et le codage HDBn.

Type de codage

***NRZ (Non Retour à Zero) :**

Dans le code en ligne NRZ (Non Retour à Zero), le bit 1 est représenté par un état significatif (par exemple, une tension clairement positive), et le bit 0 par un autre état significatif (par exemple, une tension clairement négative). Il n'existe pas d'état intermédiaire.

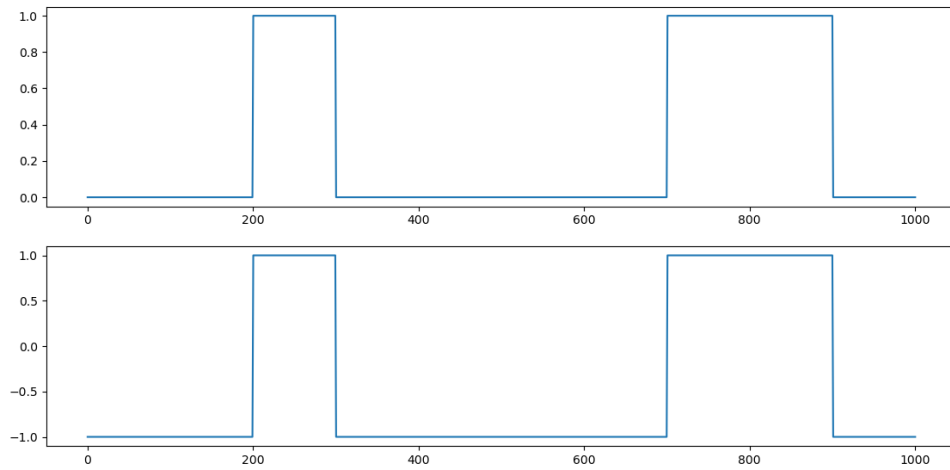


FIGURE 2.3 – Sortie de codeur NRZ

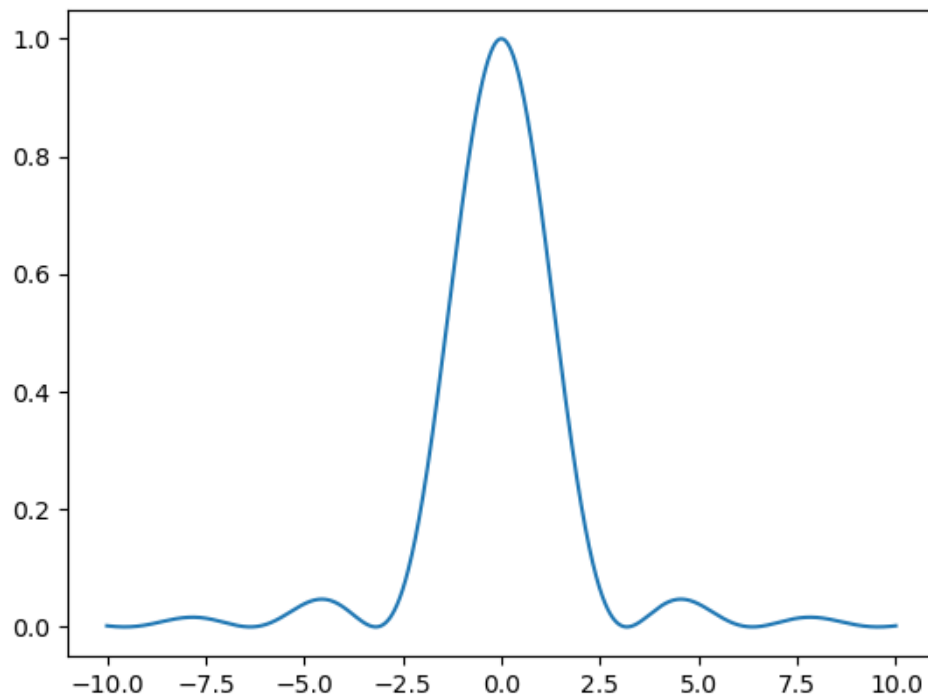


FIGURE 2.4 – DSP de NRZ

***RZ (Retour à Zero) :**

RZ (retour-à-zéro en français) est une méthode de codage en ligne à 2 niveaux utilisée dans la télécommunication dans lequel le signal retourne à la valeur zéro après chaque pulse, même s'il y a une succession de deux zéros ou de uns binaires.

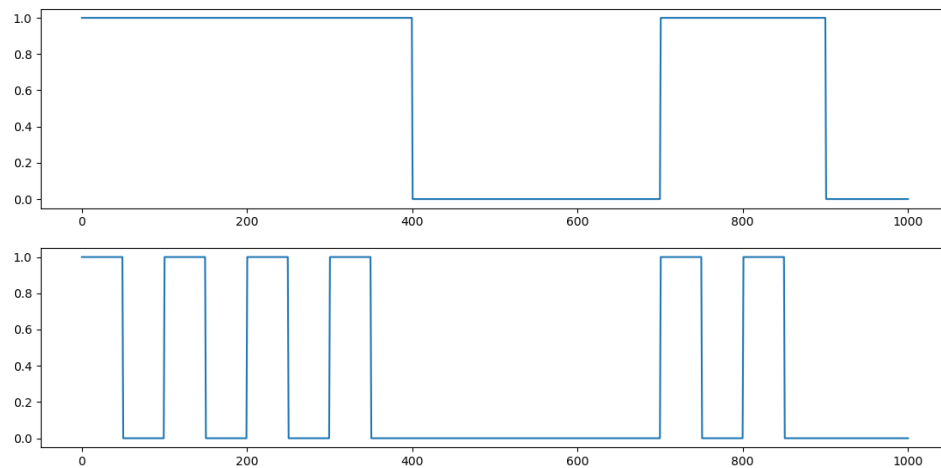


FIGURE 2.5 – Sortie de codeur RZ

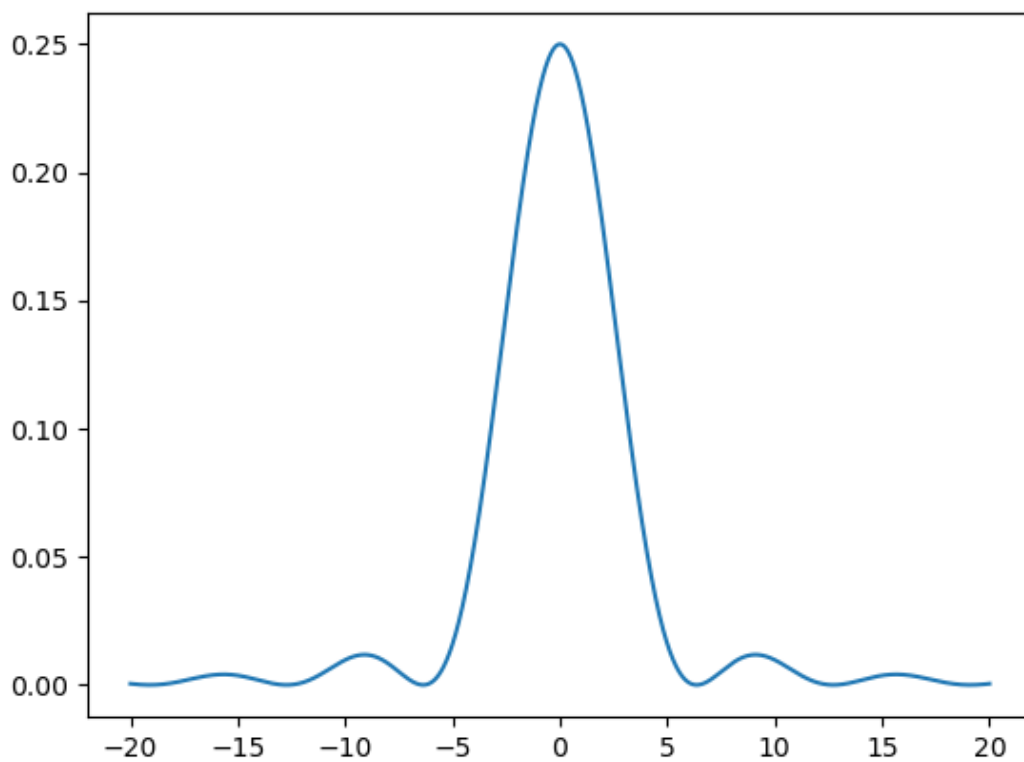


FIGURE 2.6 – DSP de RZ

*Miller :

Le codage Miller est une méthode de codage d'une information numérique pour une transmission en bande de base.

Signal intermédiaire identique au codage Manchester, puis suppression d'une transition sur deux. Le codage peut être réalisé de la manière suivante :

- transition (front montant ou descendant) au milieu du bit "1"
- pas de transition au milieu du bit "0"

-une transition en fin de bit "0" si celui-ci est suivi d'un autre "0"

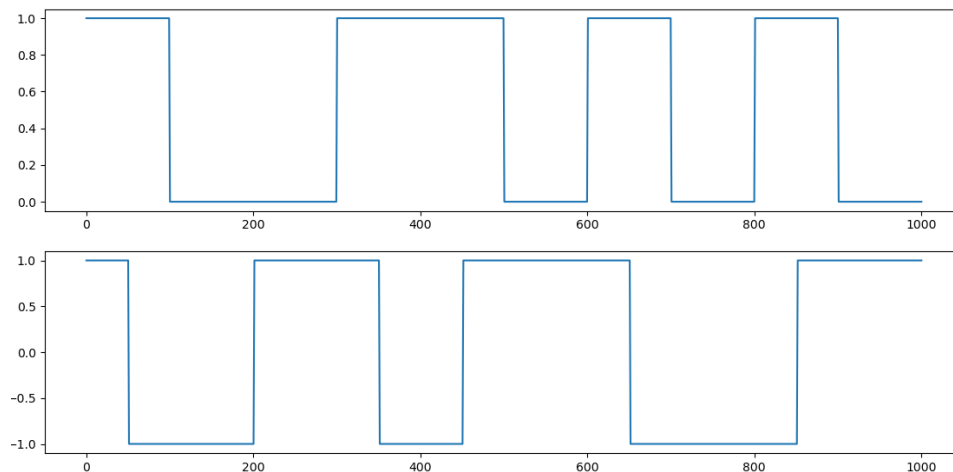


FIGURE 2.7 – Sortie de codeur Miller

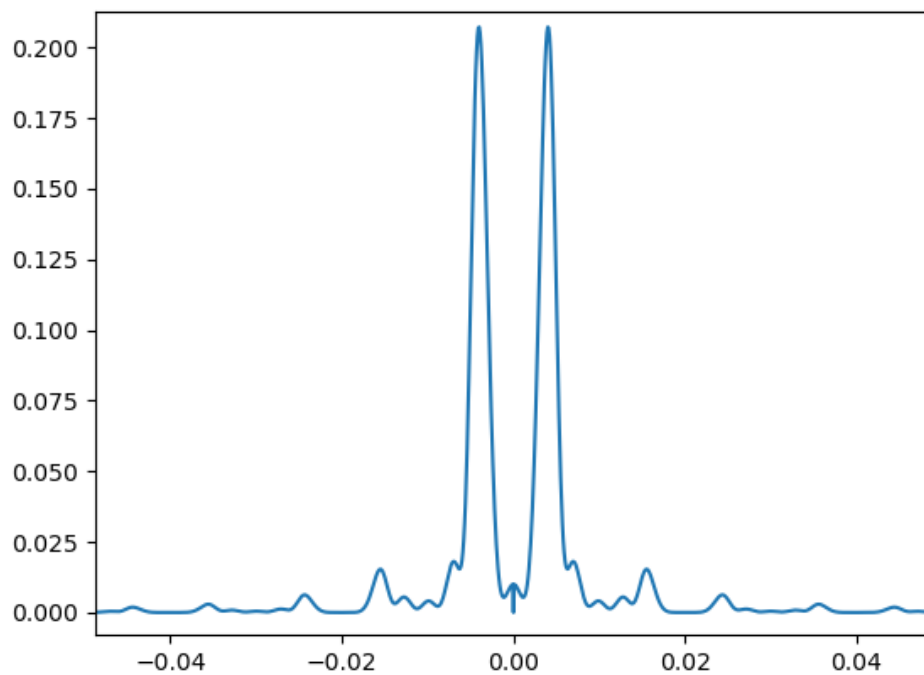


FIGURE 2.8 – DSP de Miller

***Manchester :**

Le codage Manchester est un codage synchrone, ce qui signifie que, outre les données à transmettre, les signaux transmis intègrent également l'horloge de synchronisation nécessaire à leur décodage.

Les transitions du signal codé transmettent à la fois la valeur logique du bit (0 ou 1) et l'instant de son échantillonnage. Une transition intermédiaire est ajoutée lorsque des bits de même valeur se suivent.

Toutefois le schéma du code diffère entre le côté émetteur et le côté récepteur, même si leur référence commune est le signal de synchronisation :

Du côté émetteur, le signal doit être stabilisé à la valeur du bit à transmettre au moment du top de l'horloge qui provoque la transition de synchronisation ; du côté récepteur, la transition marque le top d'échantillonnage de la valeur du bit : celui-ci est égal à la valeur qui précédait cet instant ou l'inverse de la nouvelle valeur qui suit la transition.

Si un codage simple peut s'effectuer en plaçant temporellement la synchronisation au centre du bit comme illustré ci-dessous, ce qui donne un signal « haut » durant 50% du temps, des émetteurs alimentés par piles peuvent réduire leur durée d'émission (état ON) en raccourcissant les périodes "hautes" avant ou après la transition de synchronisation, voire en ajoutant des transitions intercalaires.

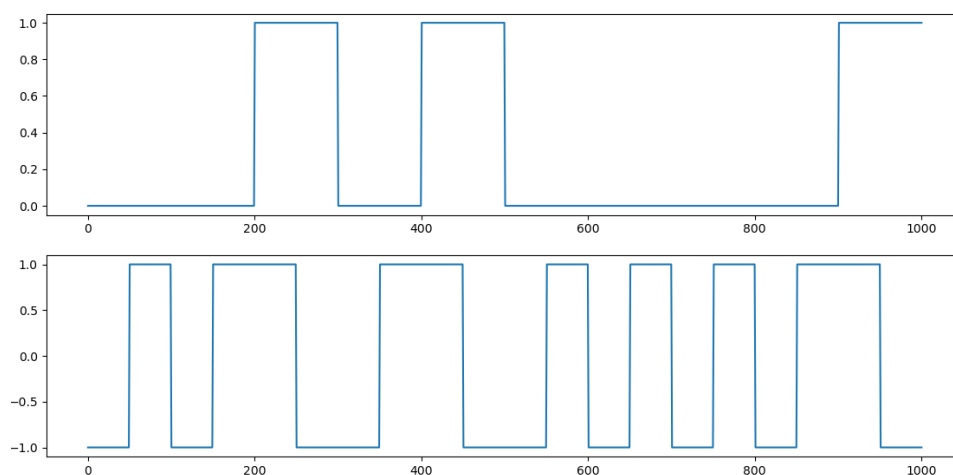


FIGURE 2.9 – Sortie de codeur Manchester

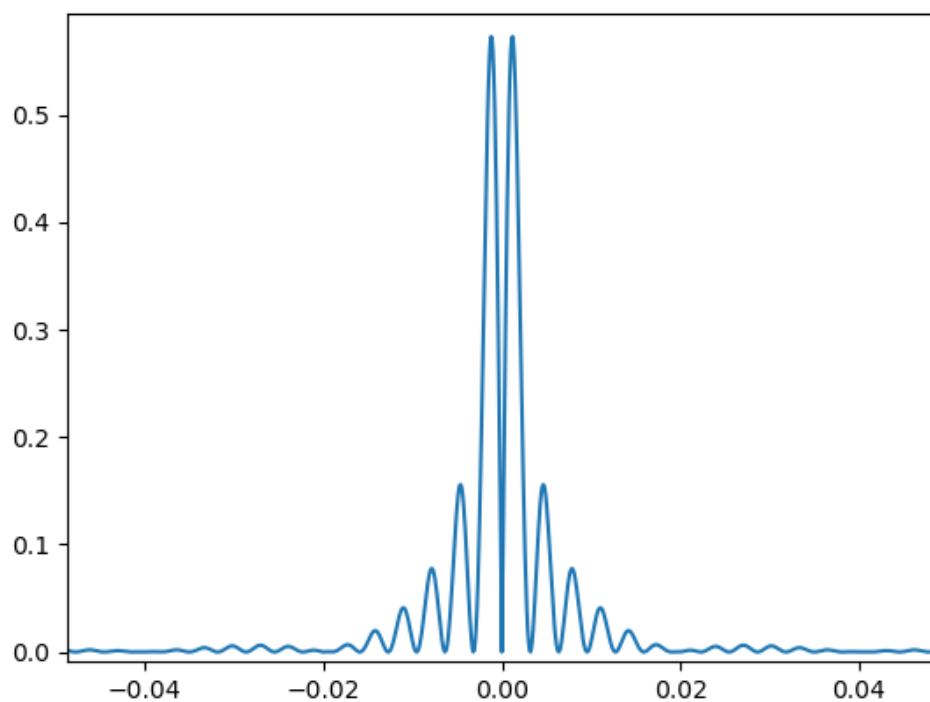


FIGURE 2.10 – DSP de Manchester

***HDBn (Haute Densité Binaire d'ordre n) :**

Ce code est une adaptation du code AMI pour éviter les pertes d'horloge. On établit une violation d'alternance volontaire pour les suites de quatre zéros :

“0000” → B00V avec B = -impulsions émises

V = violation d'alternance

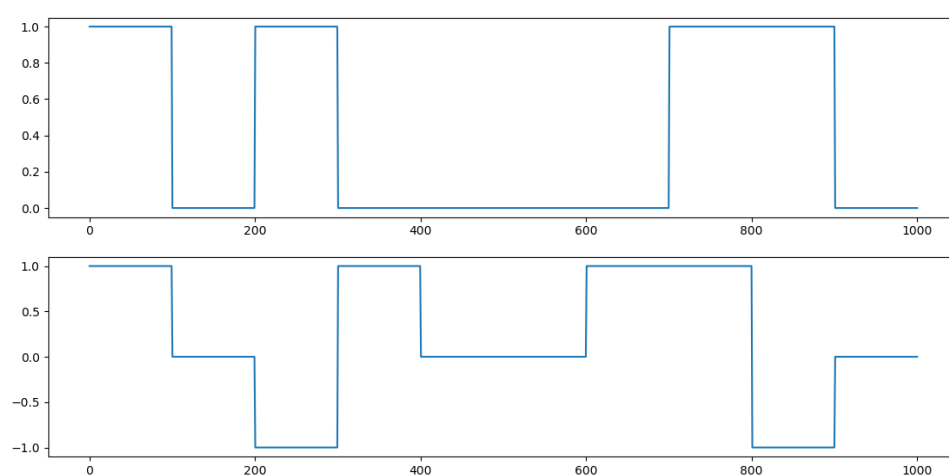


FIGURE 2.11 – Sortie de codeur HDBn

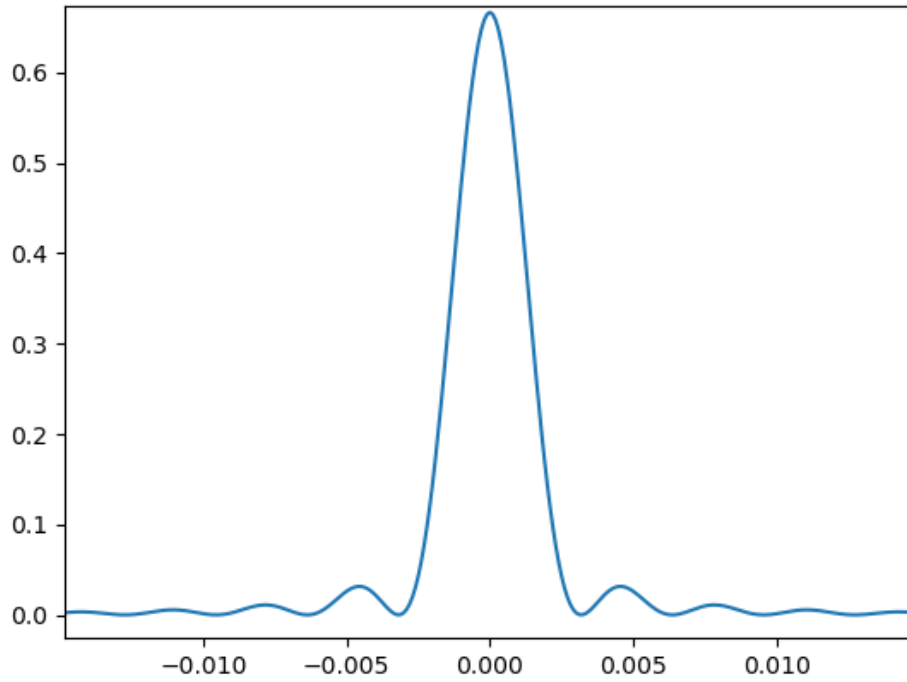


FIGURE 2.12 – DSP de HDBn

2.1.3 Filtre d'émission

Filtre blanchissant

Le filtre blanchissant convertis le signal d'entrée a des impulsions.

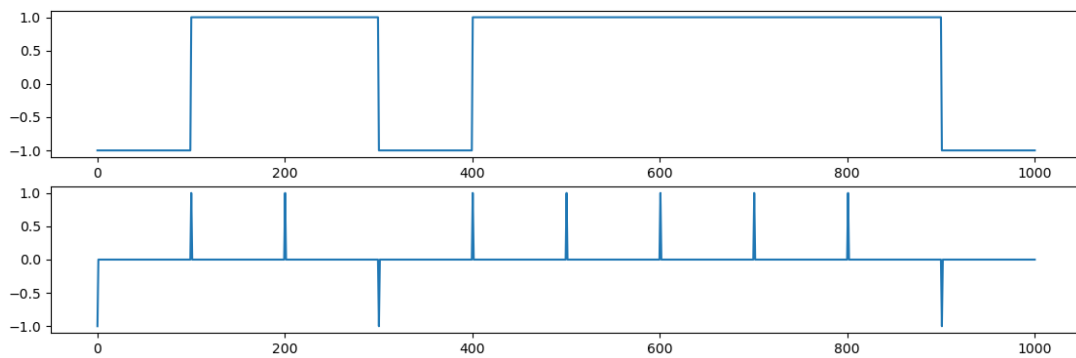


FIGURE 2.13 – Filtre blanchissant

Alors le filtre blanchissant sera la fonction réciproque de $G(f)$ notée

$$G_1(f) = \frac{1}{T_b} \frac{\pi f T_b}{\sin(\pi f T_b)} \Rightarrow G_1(f) \times G(f) = 1$$

Filtre de Nyquist

*Interférence entre symboles (IES)

Chevauchement entre symboles successifs conduisant à une erreur binaire. Ce phénomène se produit si l'amplitude de l'impulsion soumise à échantillonnage en réception dépend à l'instant de décision de symboles voisins.

Les retards des canaux et le phénomène de multi-trajet conduit à faire apparaître de l'IES.

Le phénomène d'IES a pour conséquence de réduire la marge de tolérance entre les niveaux échantillonnés en réception. En présence de bruit, il entraîne une augmentation de la probabilité d'erreur de décision.

Pour annuler l'IES, l'effet des symboles précédents doit s'annuler à l'instant d'échantillonnage = condition de Nyquist dans le temps. Pour ceci la bande passante du canal de propagation doit vérifier :

Critère de Nyquist temporel :

Pour permettre une transmission sans IES, la bande passante du canal de propagation doit vérifier :

$$B \geq \frac{1}{2T_s}$$

Pour le cas particulier : $B = \frac{1}{2T_s}$

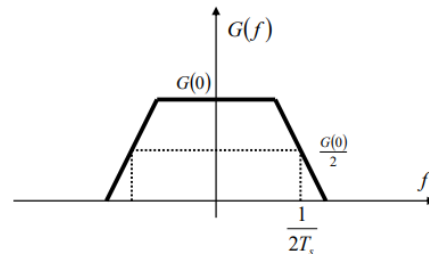
$$\begin{aligned} g(kT_s) &= 0 \Leftrightarrow k \neq 0 \\ g(0) &= 1 \end{aligned}$$

la réponse impulsionnelle $g(t)$ du canal vérifie :

Critère de Nyquist fréquentiel :

Le spectre de la réponse impulsionnelle admet le point : $\left(\frac{1}{2T_s}, \frac{G(0)}{2} \right)$

comme centre de symétrie impaire



Filtre utilisé dans les normes telecoms : filtre en cosinus surélevé

FIGURE 2.14 – Critère de Nyquist temporel et fréquentiel

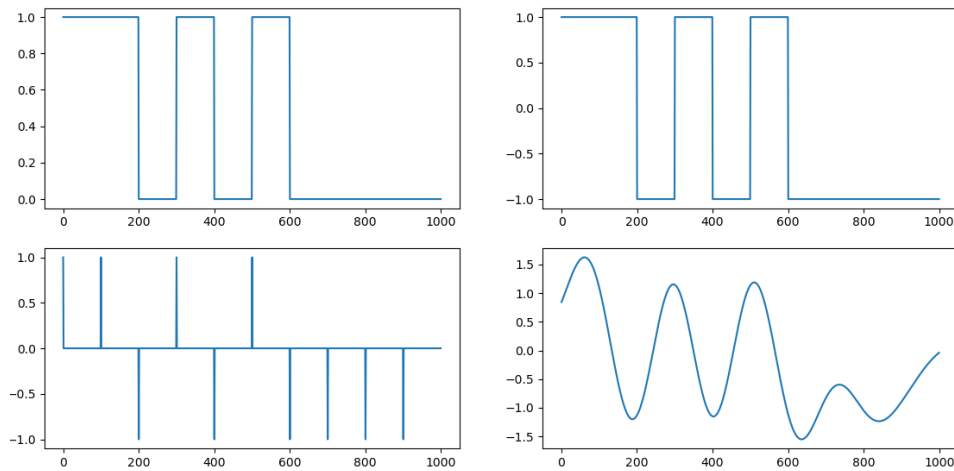


FIGURE 2.15 – Chaîne de transmission (data, codeur NRZ, filtre blanchissant et filtre de Nyquist)

Diagramme de l'œil

Le diagramme de l'œil permet de contrôler visuellement la quantité d'IES.

Superposition des tracés d'un signal aléatoire reçu sur un multiple de la durée du symbole.

Les performances du canal de transmission sont lues à travers les ouvertures verticales et horizontales de l'œil.

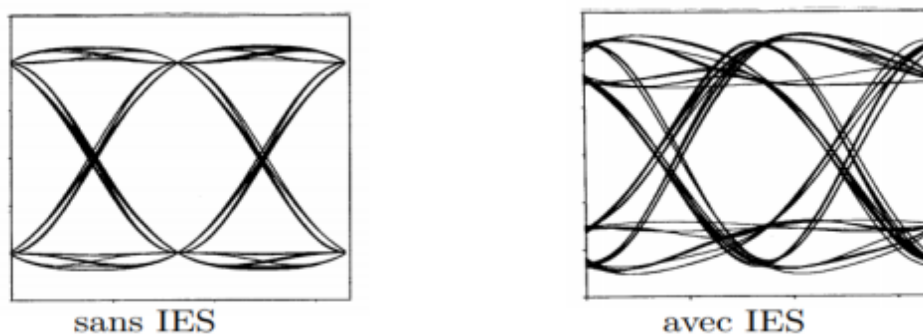


FIGURE 2.16 – Diagramme de l'œil

2.1.4 Modulation Numérique

En modulation numérique, la modulation est appliquée à une porteuse en modifiant l'amplitude, la fréquence ou la phase de cette dernière. Ce qui caractérise une modulation numérique est que le signal modulant (le message) est de nature numérique. Un signal résultant d'une modulation numérique est un signal RF. La modulation numérique en phase se nomme phase-shift keying (PSK) ou modulation par changement de phase. Dans le cas le plus simple, la porteuse est modulée par un signal binaire. On parle alors de 'binary phase-shift keying' (BPSK).

Type de modulation

*ASK (Amplitude Shift Keying)

La modulation par déplacement d'amplitude (MDA), appelée en anglais amplitude-shift keying (ASK), est un type de modulation numérique dans lequel l'amplitude du signal modulé varie.

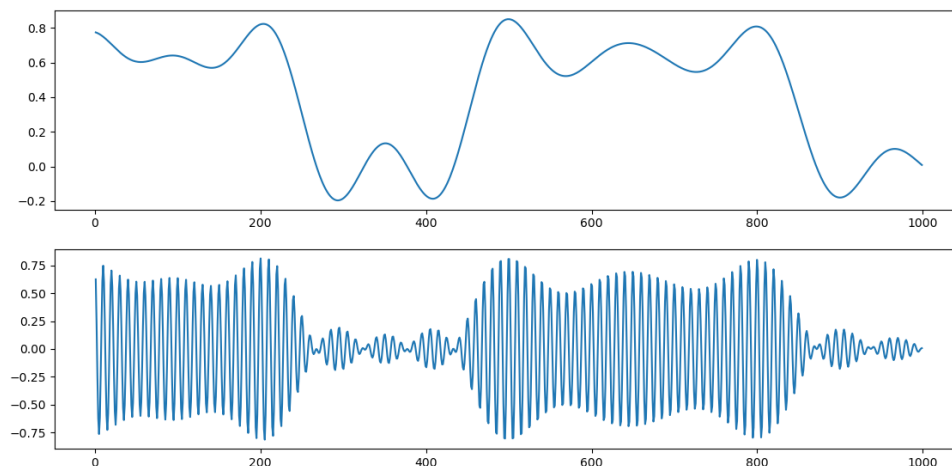


FIGURE 2.17 – Modulation ASK

Type de modulation

*PSK (Phase Shift Keying)

Le phase-shift keying (ou PSK, soit « modulation par changement de phase¹ ») désigne une famille de formes de modulations numériques qui ont toutes pour principe de véhiculer de l'information binaire via la phase d'un signal de référence (porteuse), et exclusivement par ce biais

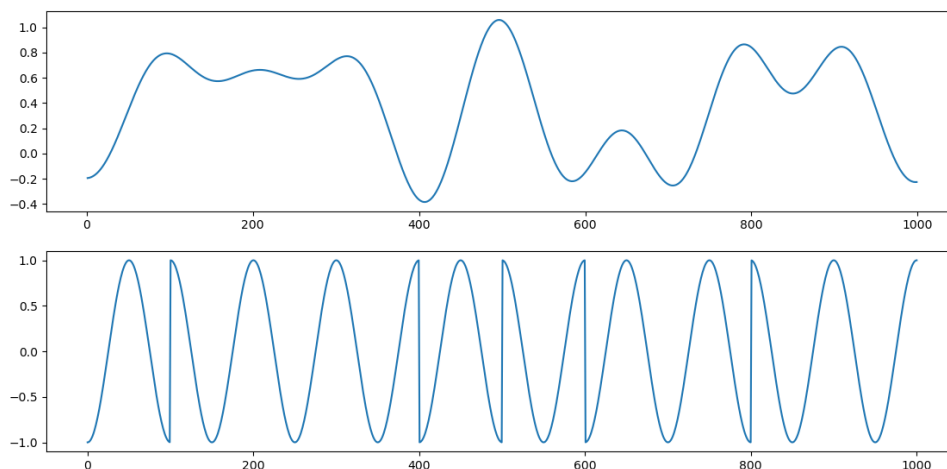


FIGURE 2.18 – Modulation PSK

Type de modulation

*FSK (Frequency Shift Keying)

La modulation par déplacement de fréquence (MDF)¹, plus connue sous sa dénomination anglophone frequency-shift keying (FSK) est un mode de modulation de fréquence numérique dans lequel la fréquence du signal modulé varie entre des fréquences prédéterminées.

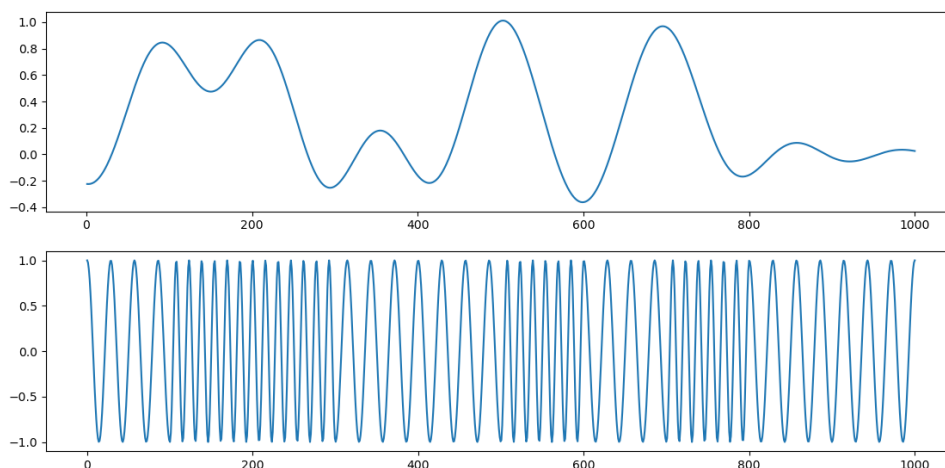


FIGURE 2.19 – Modulation FSK

2.2 Canal de transmission

Avant de concevoir la chaîne de transmission, en particulier la sélection La forme d'onde, les propriétés et les caractéristiques du canal utilisé doivent être étudiées. La puissance de bruit, le type et la stationnarité du canal sont ses paramètres Une connaissance préalable est essentielle pour une sélection efficace des formes d'onde. cette Une fois que vous comprenez la forme d'onde, vous pouvez évaluer la capacité du canal utilisation. Informations relatives à l'état du point de vue de l'opérateur La propagation est essentielle pour la première évaluation de la capacité du système et la qualité et la nature des services qu'il sera en mesure de fournir. Le canal de transmission représente la connexion entre l'émetteur et le récepteur et peut être Selon le type de données dont le transfert est autorisé, elles ont des propriétés différentes. Cette Un canal de transmission est caractérisé par sa capacité et sa bande passante.

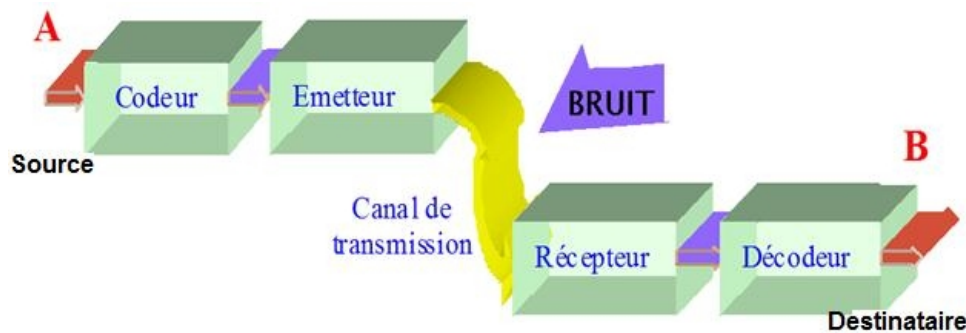


FIGURE 2.20 – canal de transmission

Les sources de perturbations sont variées et dépendent principalement de l'environnement est le diagramme des canaux de transmission (1.6). Les principaux types de bruit sont : Bruit galactique entre 20 MHz et 200 MHz dû au rayonnement de différents Énergie spatiale ; bruit atmosphérique jusqu'à 20 MHz Foudre d'orage, bruit industriel, bruit de la ville, correspondance micro-coupée Courte interruption du signal, saut de phase et scintillement Changements de phase soudains ou lents causés par l'alimentation électrique ; cela Diaphonie lors de l'exécution de plusieurs liaisons sur un seul câble.

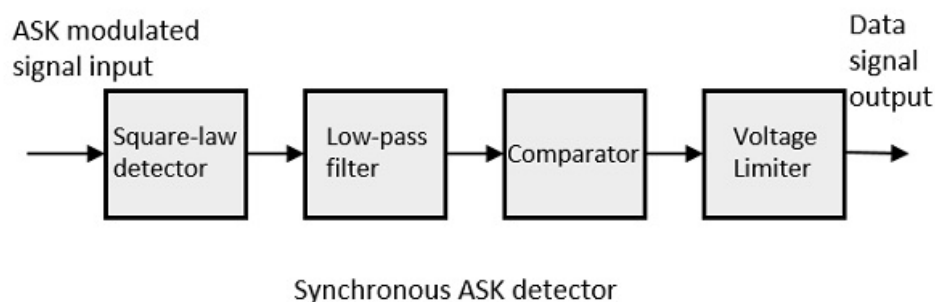
2.3 Récepteur

2.3.1 Démodulation

Démoduler : reconstituer le signal modulant $m(t)$ à partir du signal modulé reçu $S_{mod}(t)$ et de la porteuse $P(t)$.

2.3.2 Démodulateur ASK :

Le détecteur ASK synchrone se compose d'un détecteur de loi carrée, d'un filtre passe-bas, d'un comparateur et d'un limiteur de tension. Voici le schéma fonctionnel :



Le signal d'entrée modulé ASK est transmis au détecteur quadratique. Un détecteur quadratique est un détecteur dont la tension de sortie est proportionnelle au carré de la

tension d'entrée modulée en amplitude. Le filtre passe-bas minimise les hautes fréquences. Le comparateur et le limiteur de tension aident à obtenir une sortie numérique propre.

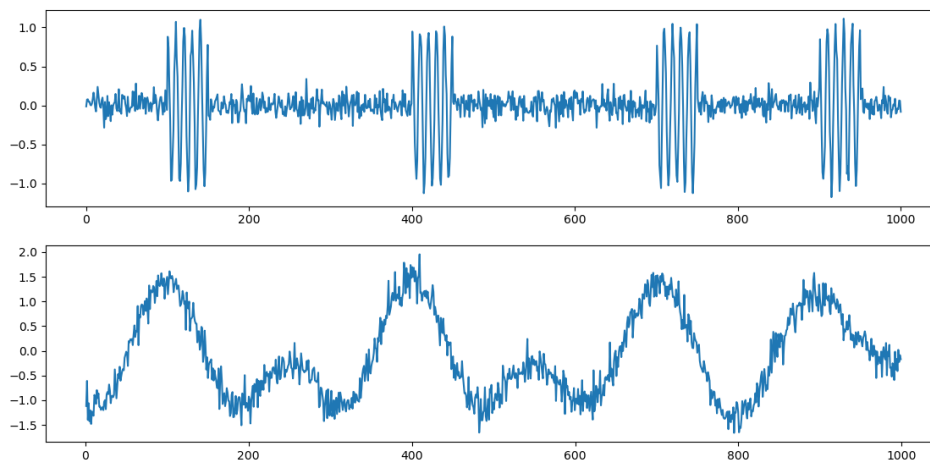


FIGURE 2.21 – ASK Demodulation

2.3.3 Démodulateur BPSK :

Le signal BPSK reçu est soit $+A \cos(2\pi f t)$ lorsque le binaire est au niveau logique 1, soit $-A \cos(2\pi f t)$ lorsque le niveau binaire est 0.

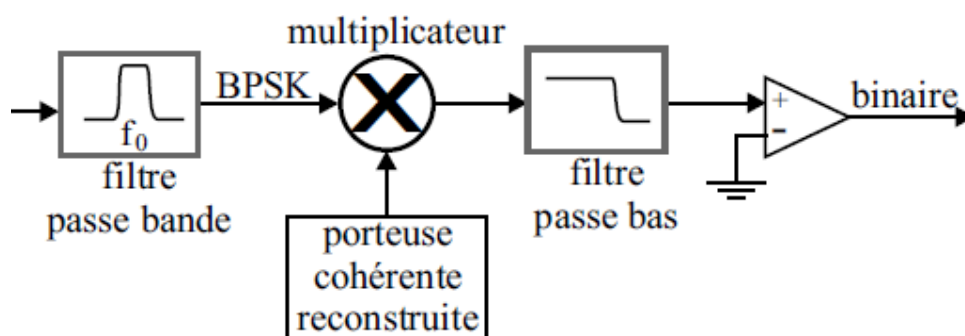


FIGURE 2.22 – BPSK Demodulation

2.3.4 Démodulateur QPSK :

Le démodulateur QPSK utilise deux circuits de démodulation de produit avec oscillateur local, deux filtres passe-bande, deux circuits intégrateurs et un convertisseur parallèle-série 2 bits. Voici le schéma :

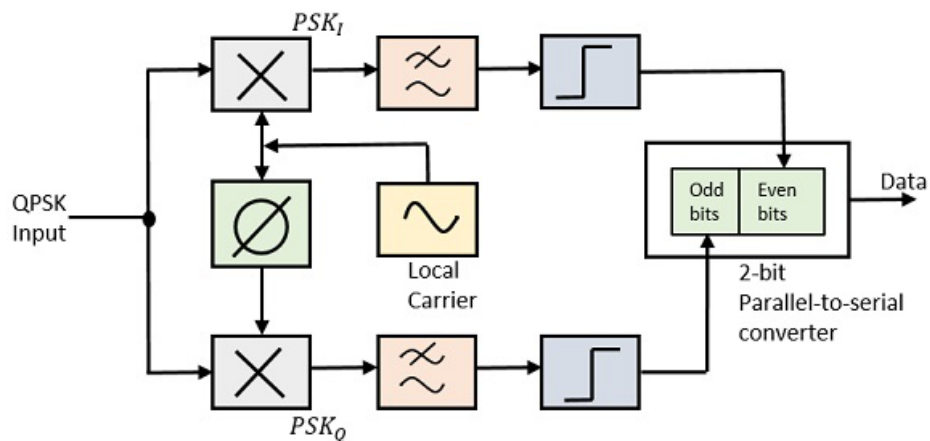


FIGURE 2.23 – QPSK Demodulation

Les deux détecteurs de produit à l'entrée du démodulateur démodulent simultanément les deux signaux BPSK. La paire de bits est récupérée ici à partir des données d'origine. Ces signaux, après traitement, sont transmis au convertisseur parallèle-série.

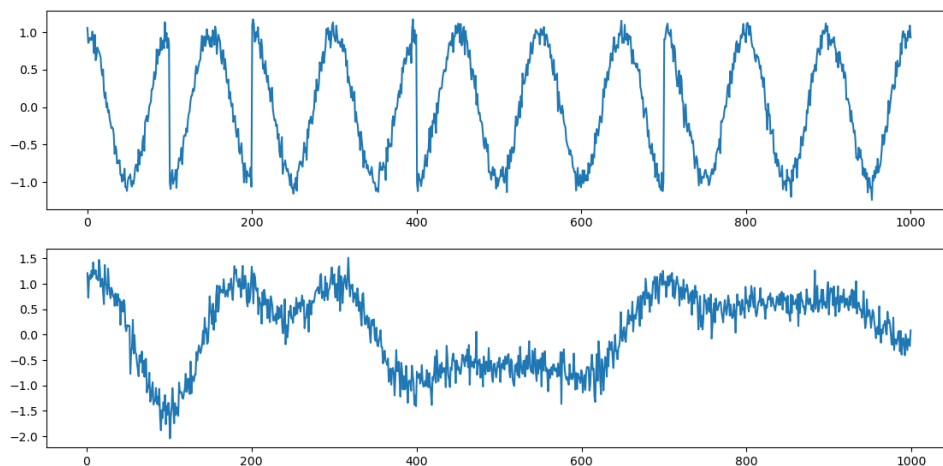


FIGURE 2.24 – PSK Demodulation

Démodulateur FSK :

La tension crête de la sortie des deux filtres est entrée dans le comparateur qui décide laquelle des deux fréquences est la plus importante. Si l'amplitude de la fréquence 2 est la plus grande, alors c'est un 1.

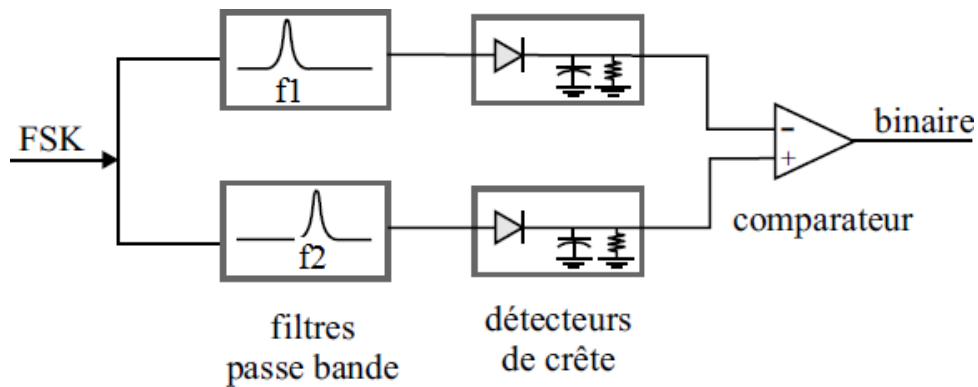


FIGURE 2.25 –

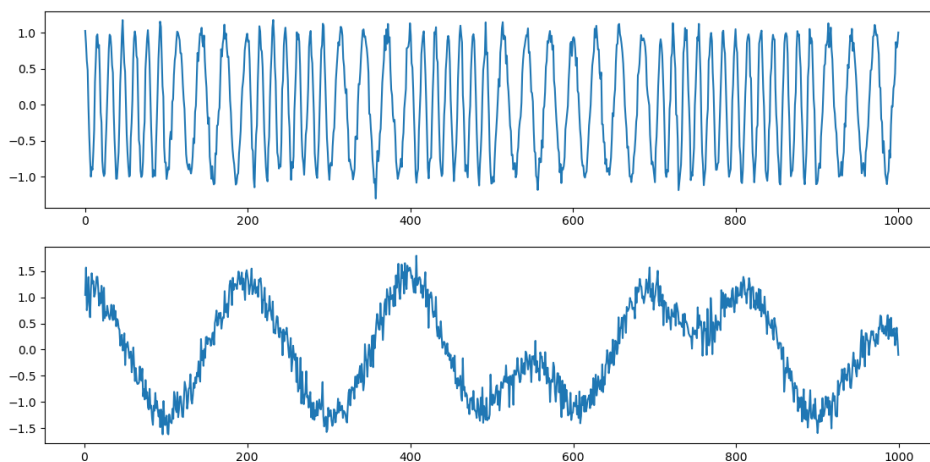


FIGURE 2.26 – FSK Demodulation

2.3.5 Filtre de réception

Le signal démodulé est un signal continu, mais le récepteur va devoir réaliser un échantillonnage afin de déterminer les éléments binaires transmis. Cependant avant l'échantillonnage, on montre qu'il faut réaliser un filtrage adapté à l'émetteur pour une réception optimale des symboles transmis. Le système de réception est très simple dans ce cas, car le signal reçu à un instant donné après ce filtrage correspond directement à un unique symbole, et celui-ci peut alors être décodé. Par contre lorsqu'un filtre d'émission est utilisé la réponse de ce filtre est généralement plus longue que , et le signal reçu à un instant ne dépend plus d'un seul symbole émis, mais également des autres symboles. Ce phénomène est appelé interférence entre symboles (IES).

IES :

Pour annuler cette interférence, il faut qu'à l'instant d'échantillonnage on ne prélève que le symbole émis, et annuler l'influence due aux autres symboles. C'est-à-dire qu'il

faut que la réponse impulsionnelle de la chaîne de transmission complète () vérifie :Le critère de Nyquist.

Le critère de Nyquist :

Le critère de Nyquist permet de déterminer l'expression du filtre de réception pour qu'il soit adapté au filtre d'émission et à la forme d'onde. De plus on peut montrer qu'il existe une répartition optimale entre les deux filtres d'émission et de réception, que l'on nomme demi-Nyquist, ou racine de Nyquist. Pour une discussion plus détaillée sur le filtre de réception et la racine de Nyquist.

2.3.6 Prise de décision :

L'étape suivante consiste à déterminer les bits correspondant au symbole reçu (D_k) après le filtre de réception. Ce symbole peut être différent du symbole qui avait été envoyé (C_k) à cause de perturbations introduites par le canal. La détection par maximum de vraisemblance est le critère optimal permettant de déterminer le symbole qui a été envoyé avec la plus grande probabilité. Pour cela on sélectionne le point de la constellation le plus proche (au sens de la distance euclidienne) du symbole reçu, et les bits qui sont associés à ce point de la constellation sont les bits qui ont été émis avec la plus grande vraisemblance. Le plan complexe est ainsi partitionné en zones de décision, chacune correspondant à un symbole de la constellation, et donc à un ensemble de bits particulier. Sur une constellation particulière, on peut représenter les limites de ces zones par des traits pointillés (on suppose que tous les symboles sont équiprobables) :

2.3.7 Taux d'erreurs binaire TEB

Une mesure de l'intégrité des données faisant référence au rapport du nombre de bits reçus en erreur sur le nombre total de bits transmis. Le rapport BER est souvent exprimé sous la forme d'une puissance de dix négative.

Chapitre 3

Software utiliser pour le simulation

3.1 Langage de programmation

3.1.1 Python

Python est le langage de programmation open source le plus employé par les informaticiens. Ce langage s'est propulsé en tête de la gestion d'infrastructure, d'analyse de données ou dans le domaine du développement de logiciels. En effet, parmi ses qualités, Python permet notamment aux développeurs de se concentrer sur ce qu'ils font plutôt que sur la manière dont ils le font. Il a libéré les développeurs des contraintes de formes qui occupaient leur temps avec les langages plus anciens. Ainsi, développer du code avec Python est plus rapide qu'avec d'autres langages.

Les principales utilisations de Python par les développeurs sont :

- *la programmation d'applications.
- *la création de services web.
- *la génération de code.
- *la métaprogrammation.

Techniquement, ce langage servira surtout pour le scripting et l'automatisation (interaction avec les navigateurs web).



FIGURE 3.1 – Python

3.1.2 Microsoft Visual Studio code

Microsoft Visual Studio est une suite de logiciels de développement pour Windows conçu par Microsoft. La dernière version s'appelle Visual Studio 2010. Visual Studio est un ensemble complet d'outils de développement permettant de générer des applications Web ASP.NET, des Services Web XML, des applications bureautiques et des applications mobiles. Visual Basic, Visual C++, Visual C et Visual J utilisent tous le même environnement de développement intégré (IDE, Integrated Development Environment), qui leur permet de partager des outils et facilite la création de solutions faisant appel à plusieurs langages. Par ailleurs, ces langages permettent de mieux tirer parti des fonctionnalités du Framework .NET, qui fournit un accès à des technologies clés simplifiant le développement d'applications Web ASP et de Services Web XML grâce à Visual Web Developer.

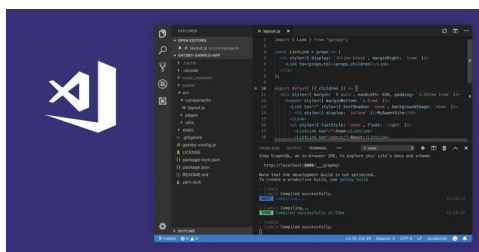


FIGURE 3.2 – Microsoft Visual Studio code

3.1.3 Numpy

NumPy est une bibliothèque pour langage de programmation Python, destinée à manipuler des matrices ou tableaux multidimensionnels ainsi que des fonctions mathématiques opérant sur ces tableaux.

Plus précisément, cette bibliothèque logicielle libre et open source fournit de multiples fonctions permettant notamment de créer directement un tableau depuis un fichier ou au contraire de sauvegarder un tableau dans un fichier, et manipuler des vecteurs, matrices et polynômes.

NumPy est la base de SciPy, regroupement de bibliothèques Python autour du calcul scientifique.



FIGURE 3.3 – Numpy

3.1.4 PyQt5

PyQt est un module libre qui permet de lier le langage Python avec la bibliothèque Qt distribué sous deux licences : une commerciale et la GNU GPL. Il permet ainsi de créer des interfaces graphiques en Python. Une extension de Qt Creator (utilitaire graphique de création d'interfaces Qt) permet de générer le code Python d'interfaces graphiques.



FIGURE 3.4 – PyQt5

3.1.5 Matplotlib

Matplotlib est une bibliothèque du langage de programmation Python destinée à tracer et visualiser des données sous formes de graphiques⁵. Elle peut être combinée avec les bibliothèques python de calcul scientifique NumPy et SciPy⁶. Elle fournit également une API orientée objet, permettant d'intégrer des graphiques dans des applications, utilisant des outils d'interface graphique polyvalents tels que Tkinter, wxPython, Qt ou GTK.

Matplotlib est distribuée librement et gratuitement sous une licence de style BSD4. Sa version stable actuelle (la 2.0.1 en 2017, la 3.5.0 en novembre 2021) est compatible avec la version 3 de Python.



FIGURE 3.5 – Matplotlib

Chapitre 4

Experimentation et discussions des résultats

4.1 La page d'accueil

Au lancement de l'application, une fenêtre d'accueil s'affiche.



FIGURE 4.1 – La page d'accueil

4.2 La page de chaine

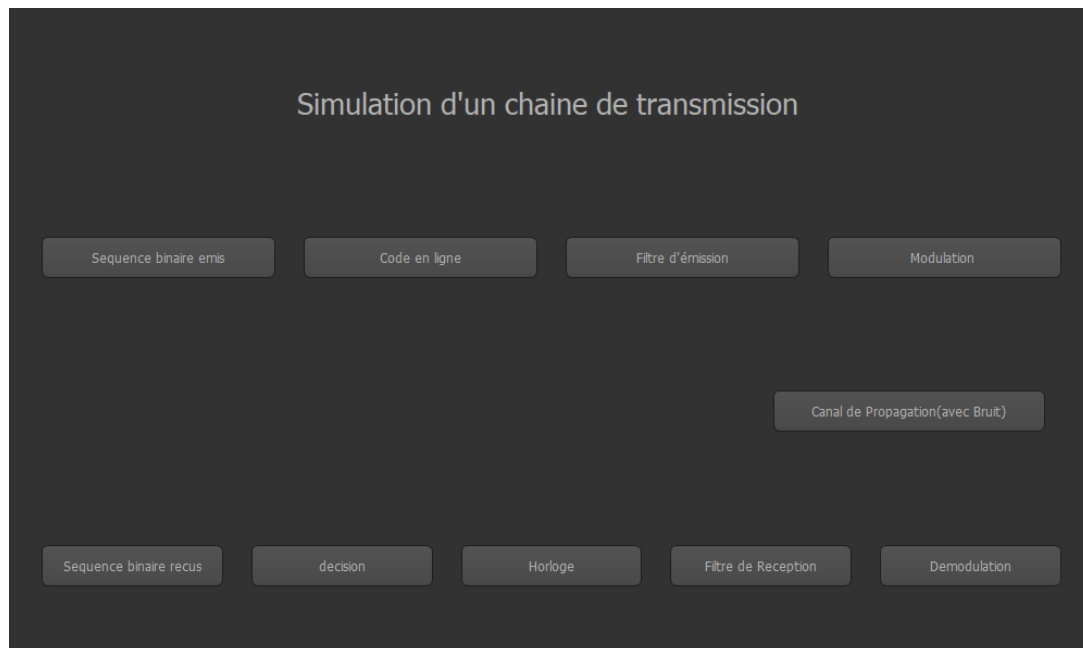


FIGURE 4.2 – Page de chaine

4.3 Séquence binaire emise

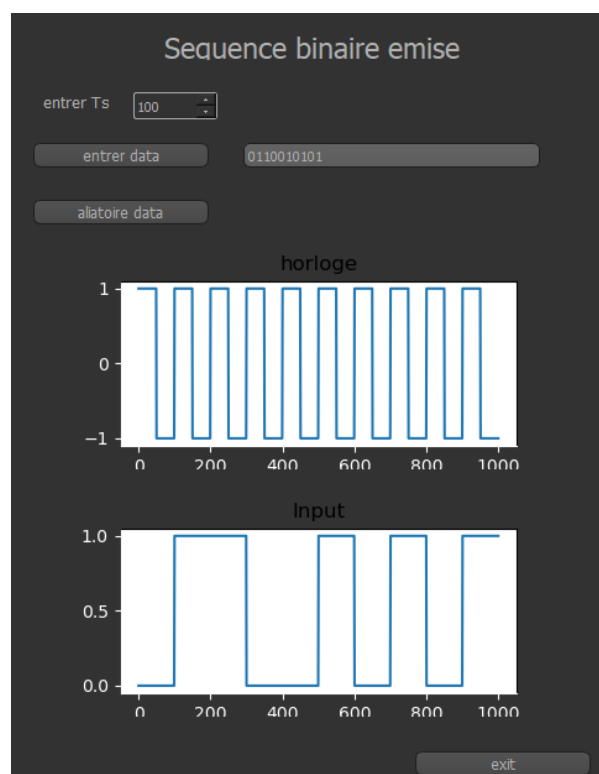


FIGURE 4.3 – Séquence binaire emise

4.4 Code en ligne

4.4.1 *Coudeur RZ

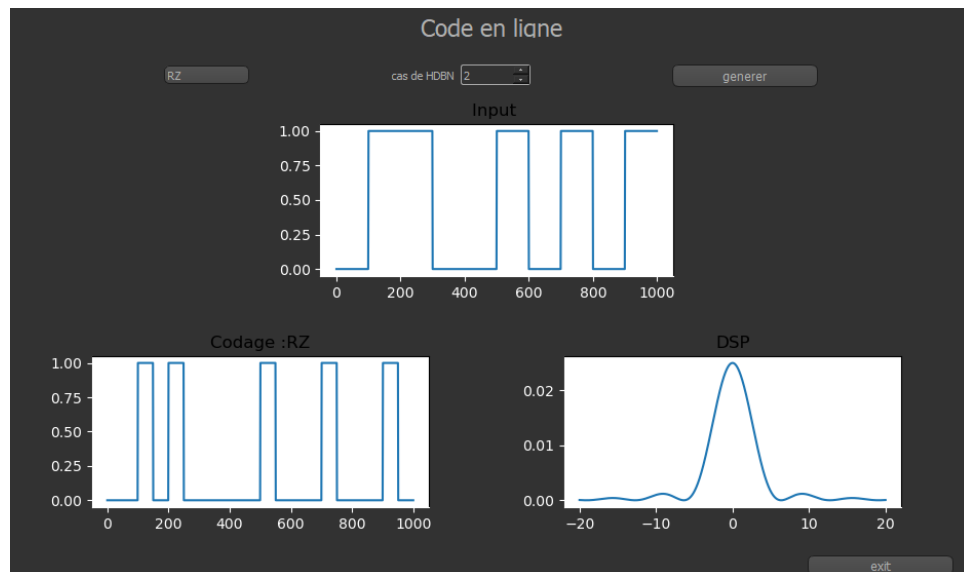


FIGURE 4.4 – Codeur en ligne RZ

4.4.2 *Codeur NRZ

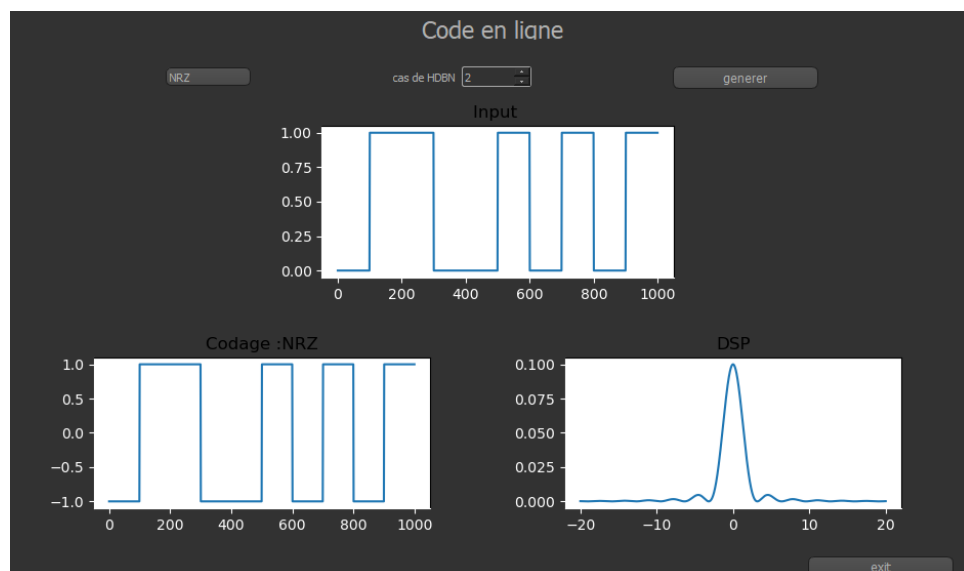


FIGURE 4.5 – Codeur en ligne NRZ

4.4.3 *Codeur Miller

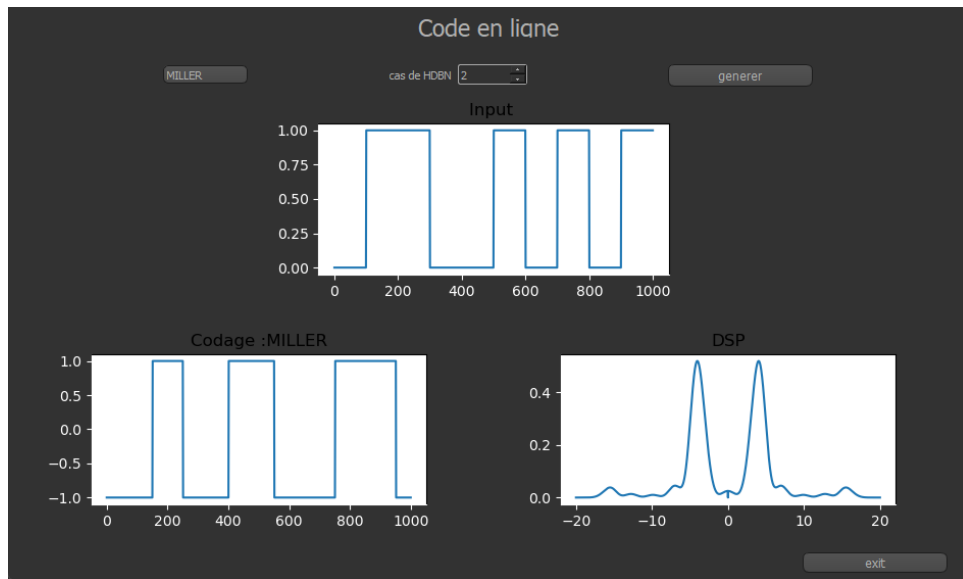


FIGURE 4.6 – codeur en ligne Miller

4.4.4 *Codeur Manchester

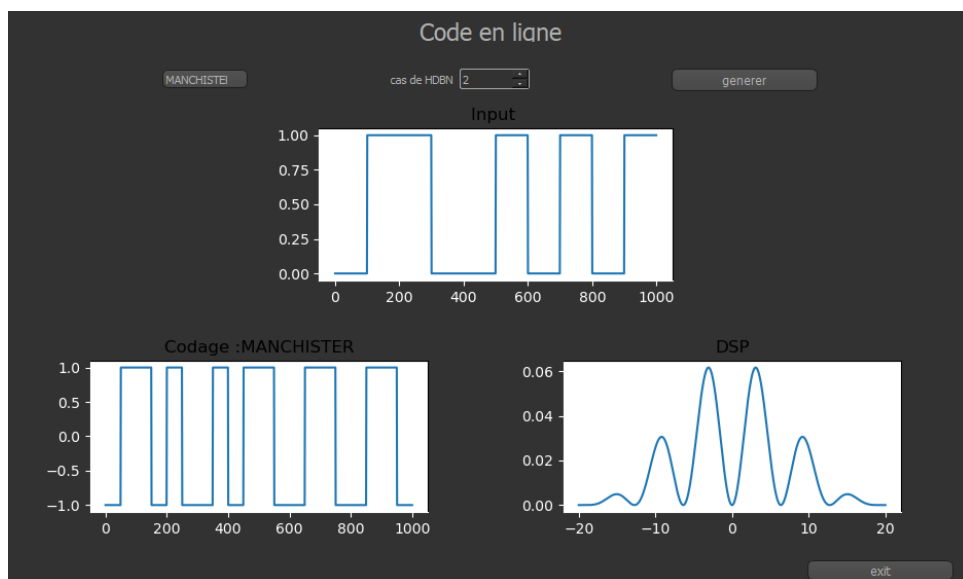


FIGURE 4.7 – Codeur en ligne Manchester

4.4.5 *Coudeur HDBN cas de N=3

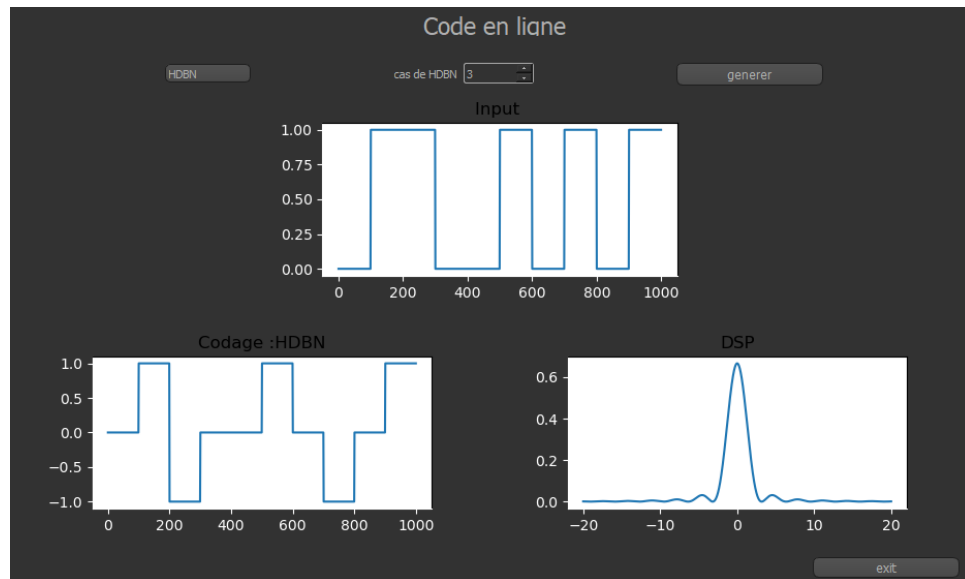


FIGURE 4.8 – Codeur en ligne HDB3

4.5 Filtre d'émission

4.5.1 Filtre d'émission avec le coefficient de retombé $\alpha=0$

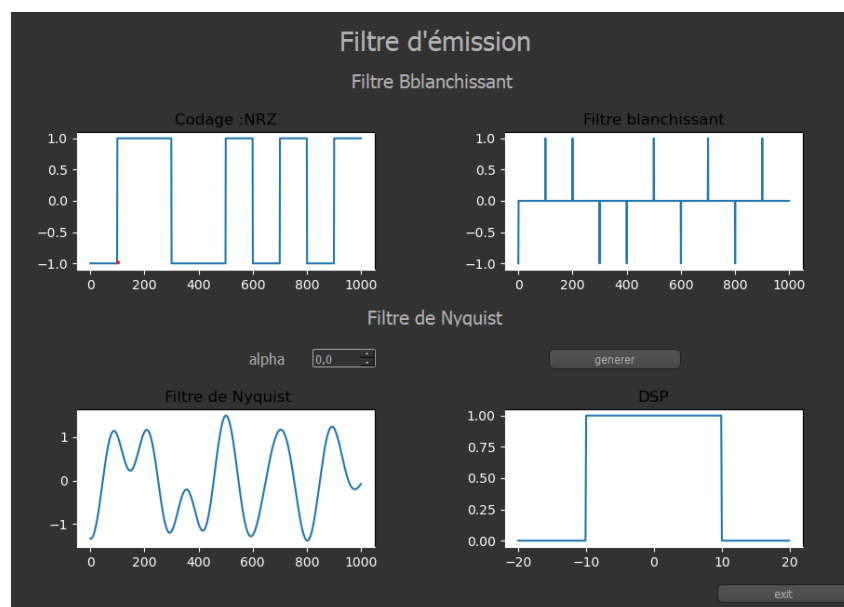


FIGURE 4.9 – Filtre d'émission $\alpha=0$

4.5.2 Filtre d'émission avec le coefficient de retombé $\alpha=0.5$

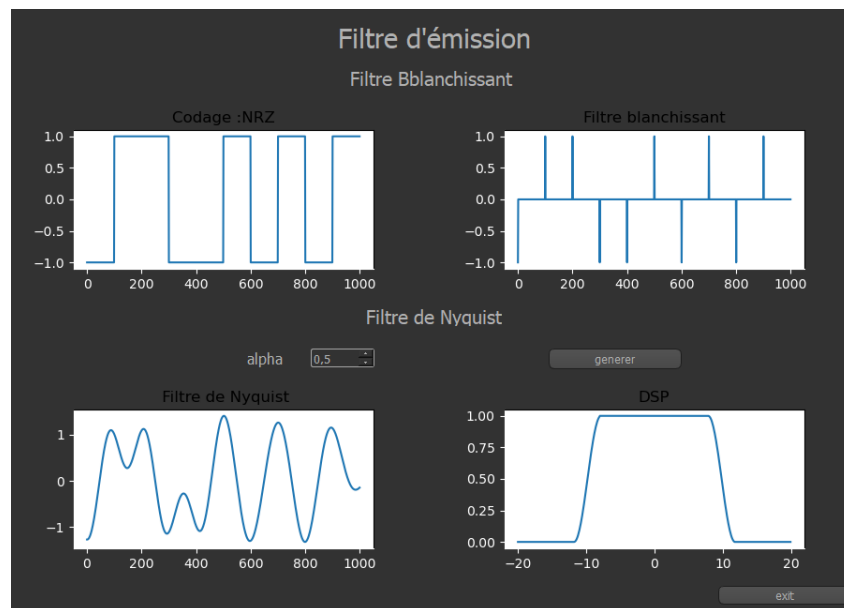


FIGURE 4.10 – Filtre d'émission $\alpha=0.5$

4.5.3 Filtre d'émission avec le coefficient de retombé $\alpha=1$

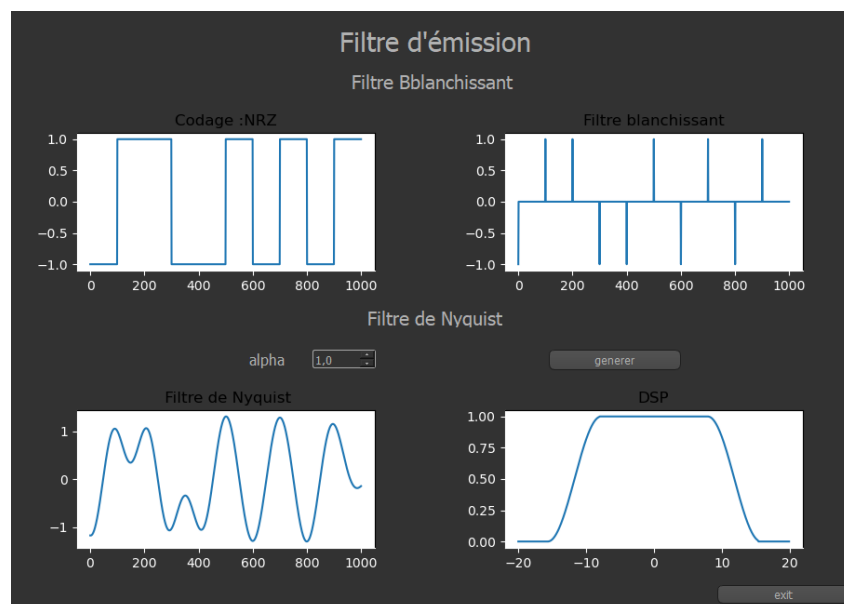


FIGURE 4.11 – Filtre d'émission $\alpha=1$

4.6 modulation

4.6.1 *Modulation ASK

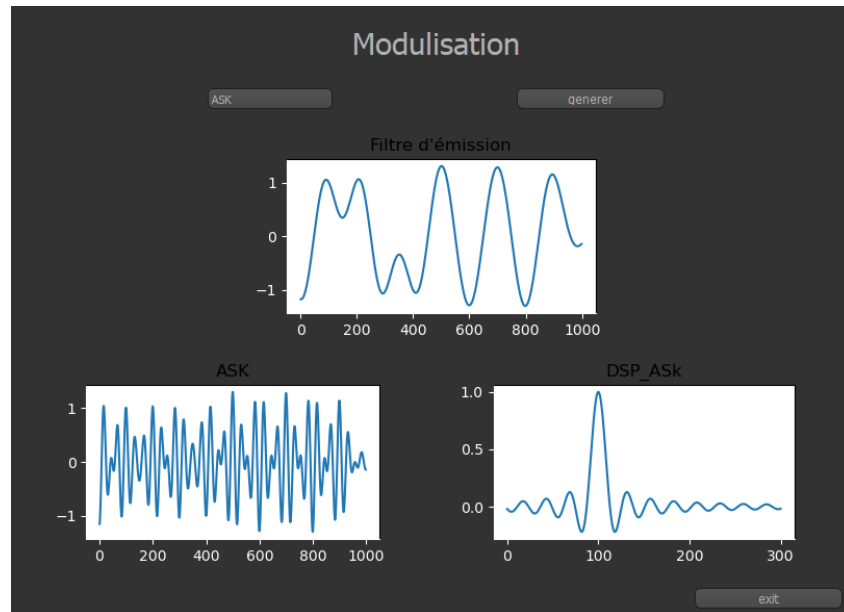


FIGURE 4.12 – Modulation ASK

4.6.2 *Modulation PSK

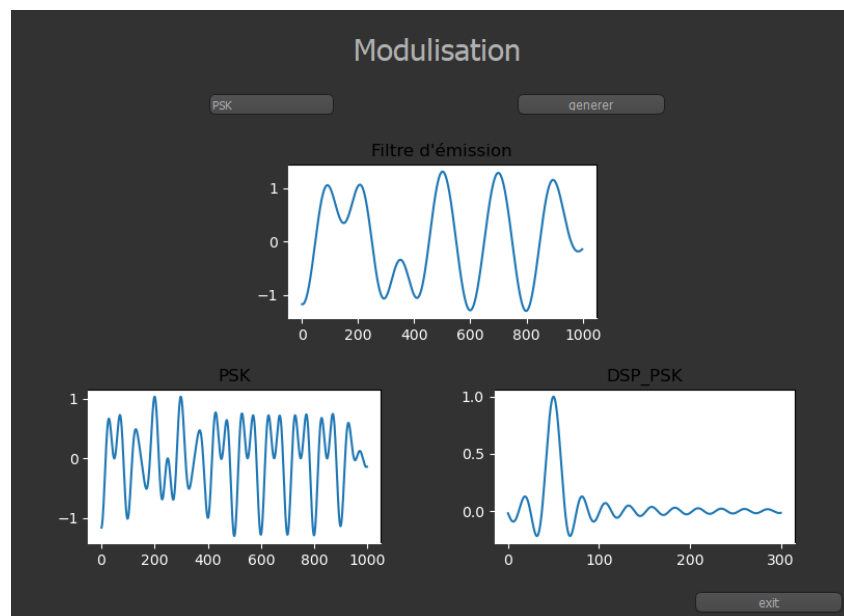


FIGURE 4.13 – Modulation PSK

4.6.3 *Modulation FSK

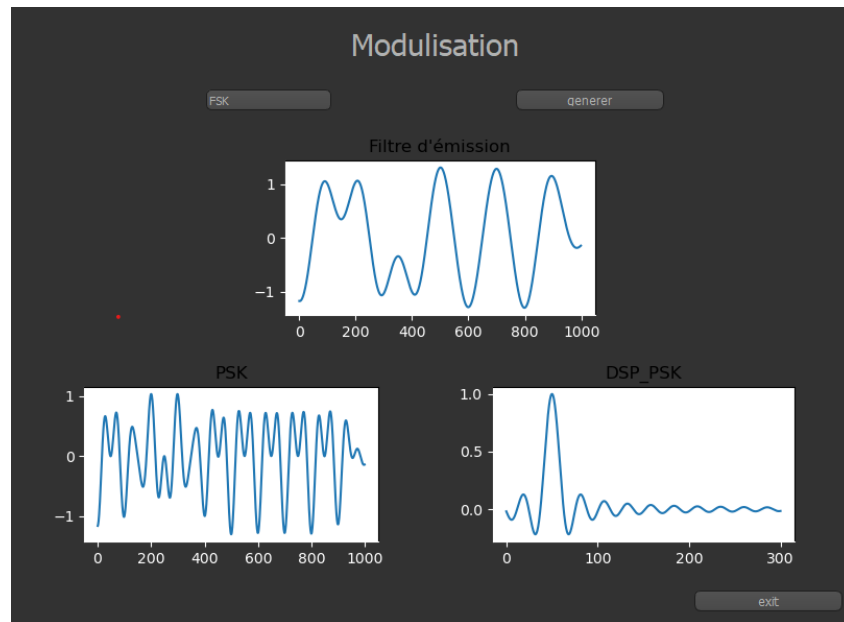


FIGURE 4.14 – Modulation FSK

4.7 Canal de Propagation(avec Bruit)

4.7.1 Bruit avec varinace=0.1

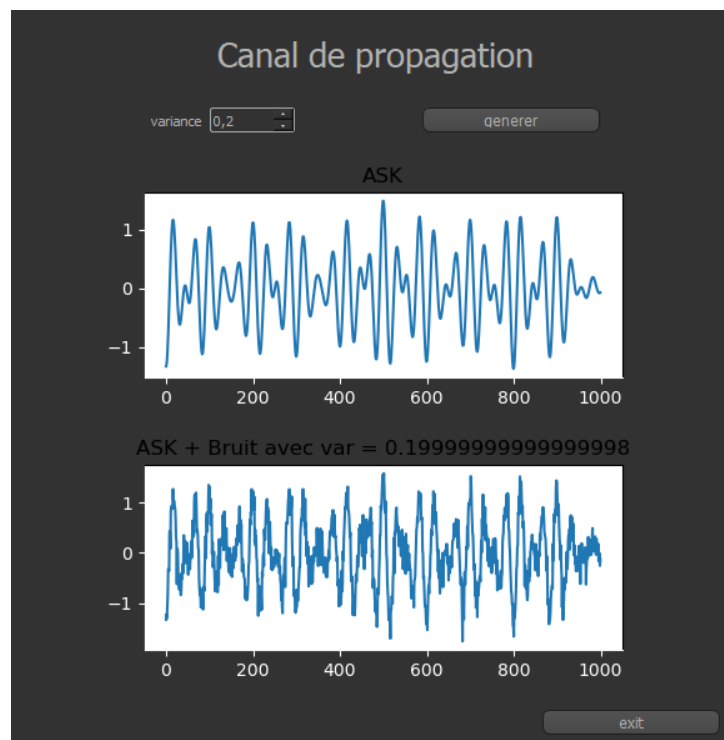


FIGURE 4.15 – Canal de Propagation(avec Bruit) varinace=0.1

4.7.2 Bruit avec varinace=0.3

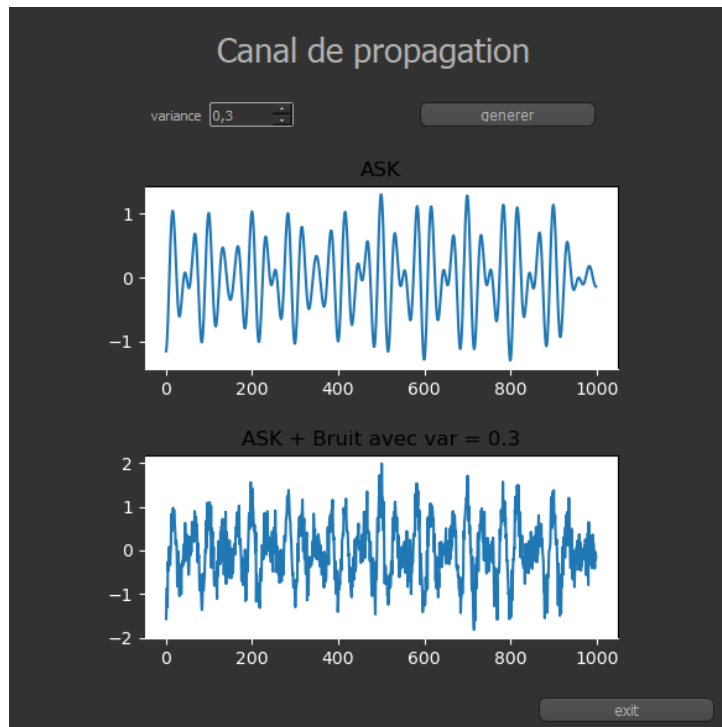


FIGURE 4.16 – Canal de Propagation(avec Bruit) varinace=0.3

4.7.3 Bruit avec varinace=1

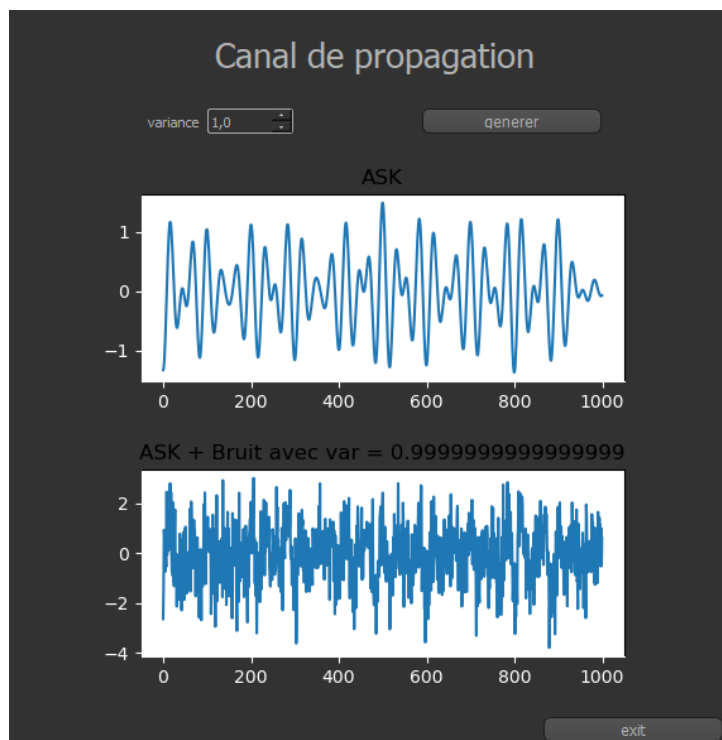


FIGURE 4.17 – Canal de Propagation(avec Bruit) varinace=1

4.8 Demodulation

4.8.1 Demodulation adapter au modulation

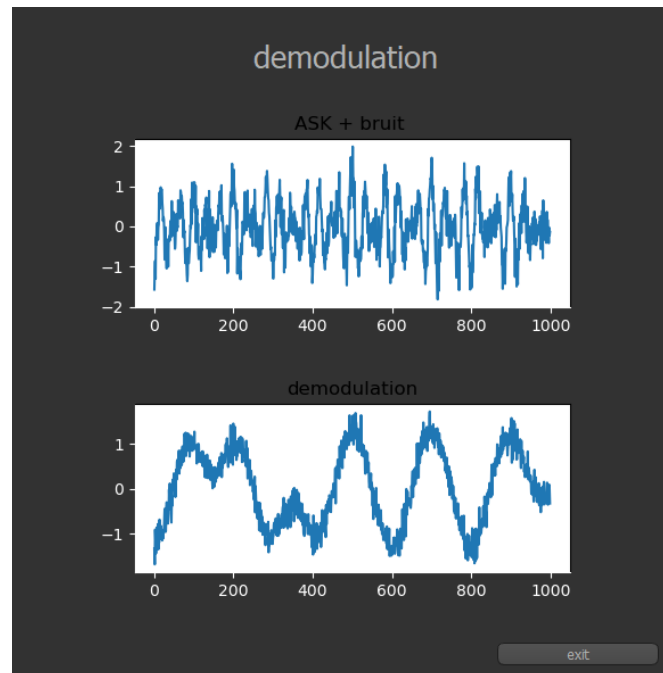


FIGURE 4.18 – Demodulation

4.9 Filtre de Reception

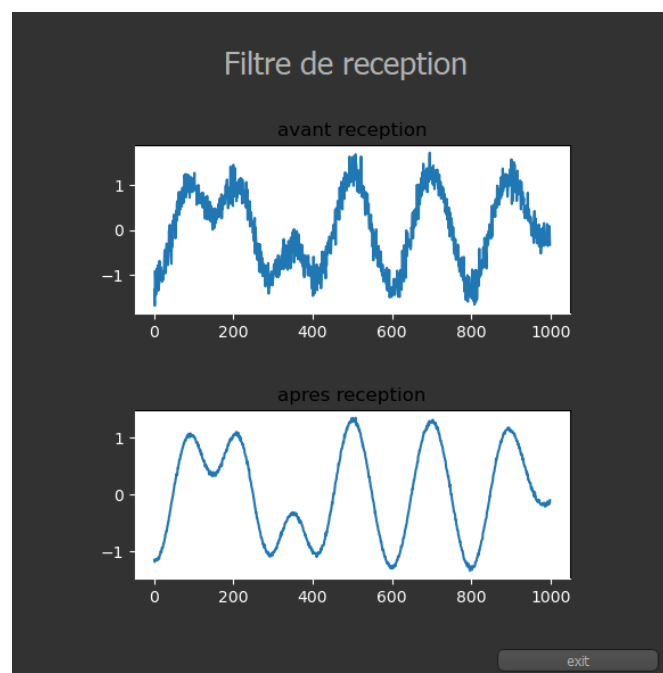


FIGURE 4.19 – Filtre de Reception

4.10 décision

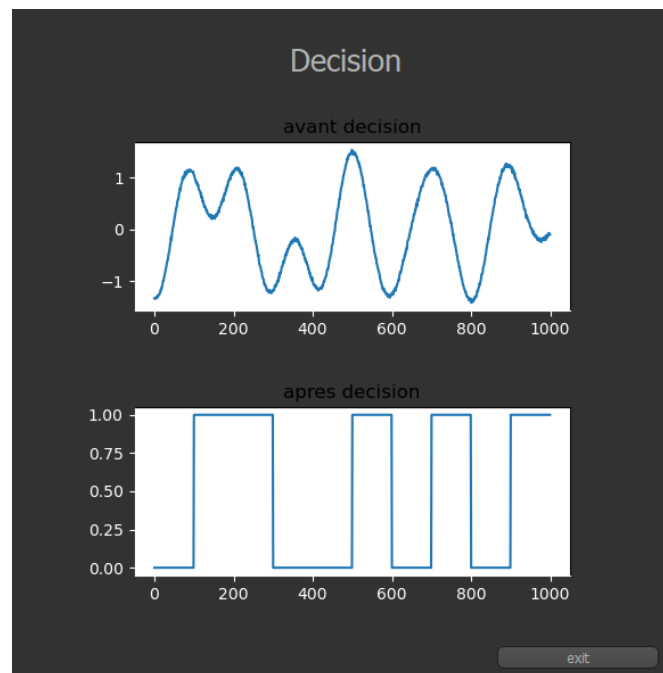


FIGURE 4.20 – décision

4.11 Horloge

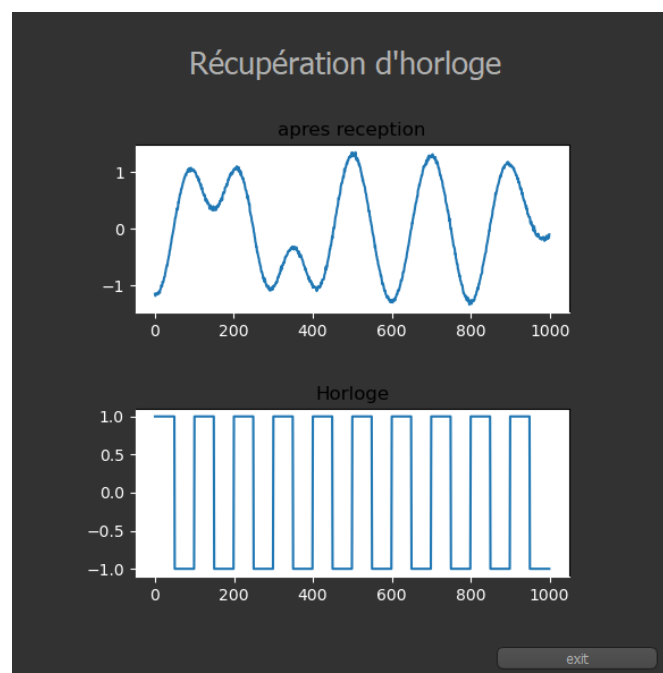


FIGURE 4.21 – Horloge

4.12 Séquence binaire reçue



FIGURE 4.22 – Séquence binaire recue

4.12.1 Conclusion :

Dans ce projet, nous avons parlé de la chaîne de communication numérique et de ses différentes étapes.

Tout d'abord, nous commençons par des différents types d'encodeurs en ligne, puis nous avons discuté filtre d'émission et comment fonctionne la modulation et ses types. nous avons également vu le cote de reception qui contient les processus inverse du l'émission : demodulation, filtre de reception et decodage. Nous avons donc créé une chaîne de communication numérique à l'aide de langage de programmation python.

Enfin Nous avons pu connecter ce modèle avec une interface graphique qui nous a permis de configurer Différents blocs de la chaîne et visualisation du DSP et du taux d'erreur binaire.

Ce projet nous a vraiment aidé à développer nos connaissances en communication numerique et nous avons également renforcé nos compétences en programmation sur Python et de comprendre comment traiter les outils de ce langage (Numpy,Pyqt5...).