
Telecommunication

modulation numérique

Travaux Pratiques (MatLab & Simulink)

able des matières

1	Modulation/démodulation en bande de base	6
1.1	Avant-propos	6
1.2	Modulation des signaux numériques.....	7
	Codage en ligne – Réalisation du modèle "codage en ligne"	
1.3	Démodulation : cas d'un canal de transmission bruité	13
	Méthode du corrélateur – Réalisation de la démodulation en bande de base	
2	Modulation/démodulation sur fréquence porteuse	15
2.1	Avant-propos	15
2.2	Etude d'une transmission à travers un canal à bande infinie : modulation/démodulation	16
2.3	Transmission de symboles de n bits	17
2.4	Représentation complexe du signal	18
3	Chaîne de communication	20
3.1	Avant-propos	20
3.2	Emission/réception QAM-M (ou PSK-M) sur fréquence porteuse.....	21
3.3	Emission/réception QAM-64 (ou PSK-64) en bande de base	22

ntroduction

La transmission des signaux numériques Chaîne de transmission

➤ La chaîne de transmission numérique

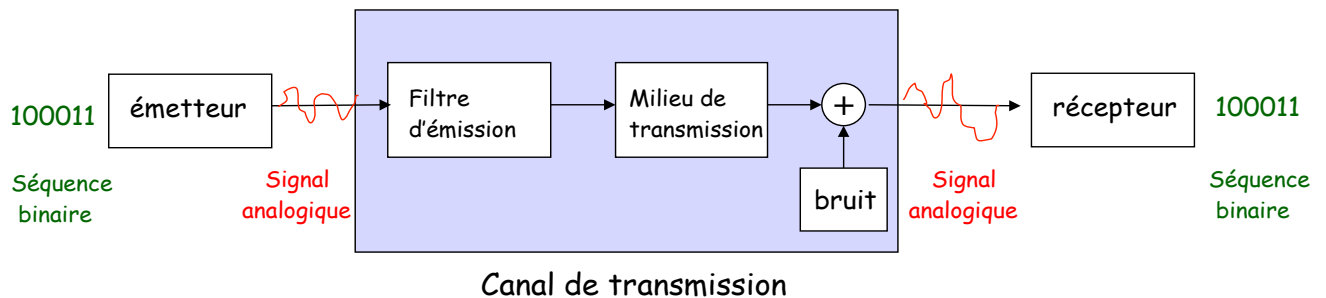


FIGURE 1 – Chaîne de communication numérique

La simulation de la chaîne de communication sera complète, et comprendra :

– TP 1

- la simulation d'une séquence binaire : tirage aléatoire de 1 et 0,
- le codage de symboles 1 et 0 : ~~transformation~~ ^{TUT R&T 1^{ère} année} symboles en un signal, électrique, pas exemple. Cette opération, qui est effectuée au niveau de l'émetteur, s'appelle **codage en ligne** ou **codage en bande de base** (NRZ, RZ, Manchester, etc..).
- le décodage de ce signal, permet de décider si le signal reçu est un 1 ou un 0. Les performances du décodeur seront mesurées en fonction du bruit lié à la transmission (bruit de canal). Cette opération est réalisée au niveau du récepteur.

– TP 2

Pour s'adapter à la bande passante du signal on utilise des techniques appelées **modulation sur fréquence porteuse**. On étudiera particulièrement la modulation QAM-2 qui sera généralisée aux modulation QAM-4. La démodulation (décodage) de ces signaux sera étudiée ainsi que l'influence de la largeur de bande du canal et du bruit sur la réception.

– TP 3

Une chaîne de transmission WiMAX (modulation QAM-64, débit 54 Mbits/s, fréquence porteuse 5 GHz) sera réalisée. Ses performances, en fonction de la largeur de bande du canal et du bruit de transmission, seront évaluées.

Nous utiliserons Simulink pour simuler ces transmissions numériques.

Qu'est-ce qu'est Simulink ?

Simulink est une plate-forme de simulation qui utilise le noyau MatLab.

Ses principales caractéristiques et utilisations sont :

- un environnement complet pour la modélisation, la simulation et l'implantation de systèmes dynamiques intégrés,
- la conception et le test de systèmes linéaires ou non linéaires, à temps discrets, à temps continu hybride ou multi cadence,
- le développement d'applications pour le contrôle, les DSP, les communications et les systèmes,
- une architecture ouverte qui permet l'intégration de modèles provenant d'autres environnements.

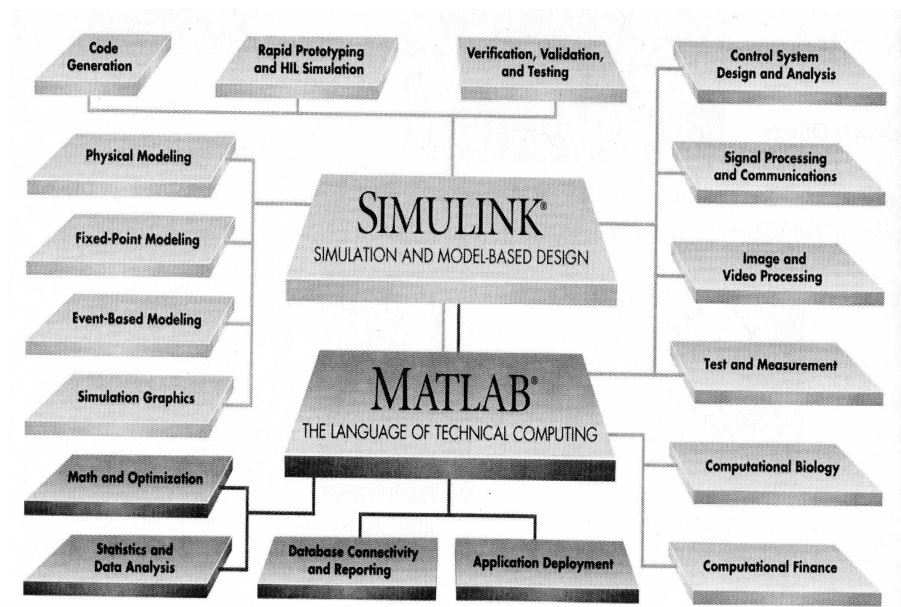


FIGURE 2 – Simulink dans l'environnement MatLab

L'environnement graphique, constitué de blocs, permet de réaliser facilement des plateformes de simulation. Ce logiciel est largement utilisé dans l'industrie, en particulier dans les secteurs automobile, aéronautique et de communication numérique.

Configuration de Simulink

Lancer MatLab et taper sous la fenêtre de commande *Simulink* :

» Simulink

Vous pouvez aussi utiliser l'icône Simulink qui est dans la barre de menu MatLab.

Pour créer un nouveau modèle, aller dans *File* puis *New* puis *Model*. Penser à enregistrer chaque nouveau modèle dans votre répertoire de travail. Vous allez assembler des blocs : la recherche d'un nouveau bloc se fait en utilisant la fonction recherche de bloc (*find block*) dans la fenêtre *Simulink Library Browser*.

Nous voulons simuler une chaîne de communication numérique. Nous devons donc configurer le modèle pour travailler sur des séquences numériques.

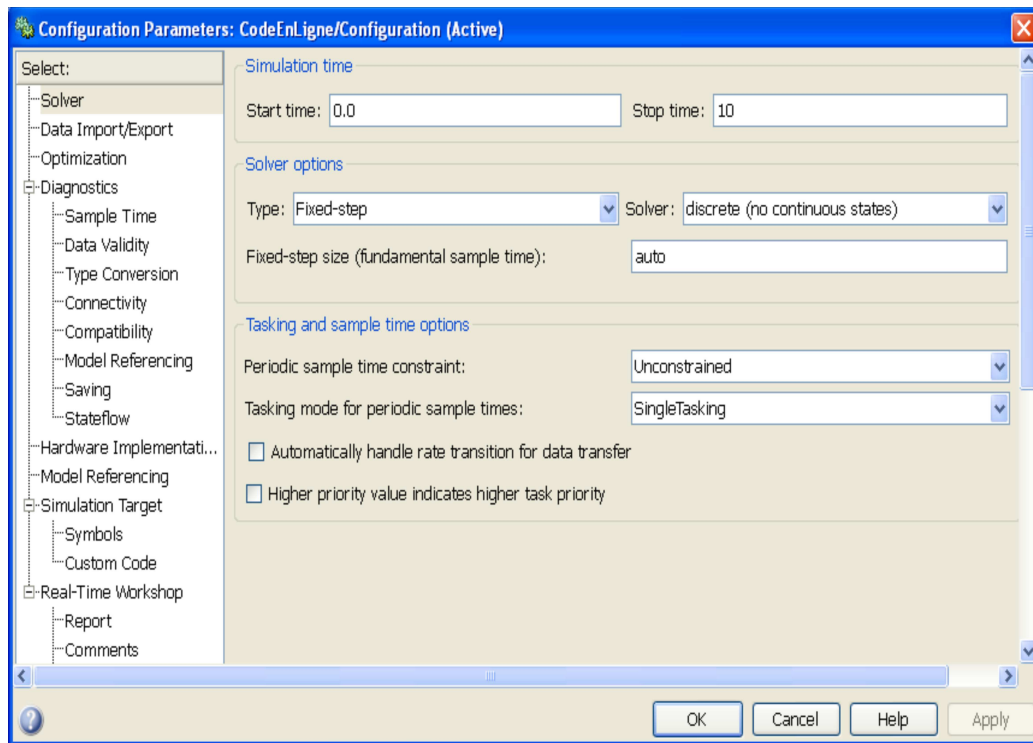


FIGURE 3 – Configuration d'un modèle Simulink

Pour tout le déroulement de ces TP et **pour tous** les modèles que vous allez créer, Simulink doit être configuré de la façon suivante (voir figure 3) :

Aller dans *Simulation* puis *Configuration Parameters* de chaque nouveau modèle et régler :

- *Solver* puis le champ *type* : *Fixed-step*
- *Solver* puis le champ *solver* : *Discrete*
- *Solver* puis le champ *Tasking mode for periodic sample times* : *Single Tasking*
- *Optimization* puis *Signal ans parameters* : cocher *Inline Parameters*
- *Diagnostic/Sample time* puis le champ *single task rate transition* : *None*

Simulink est maintenant configuré pour travailler sur des signaux numériques et simuler une chaîne de communication.

TP 1

odulation/démodulation en bande de base

1.1 Avant-propos

- Une séquence binaire composée d'une suite de symboles "0" et "1" (0 1 1 0 1 1 0 0) qui n'a pas de signification physique. Pour transmettre l'information qu'elle représente, il faut représenter les "0" et "1" par un signal physique $s_0(t)$ et $s_1(t)$, appelé **forme d'onde du codage** : on appelle cette opération **codage en ligne**.
- Le signal représentant la séquence binaire peut être électrique, optique, électromagnétique selon la nature du canal de communication utilisé pour la transmission (câble coaxial, fibre optique, canal hertzien, ...). Il peut être modulé (ou non) par une fréquence porteuse afin de s'adapter à la bande passante du canal de transmission. Il sera donc filtré par le canal, atténué et généralement du bruit perturbera la transmission.
- Le signal est alors reçu par un récepteur et démodulé pour reconstituer la séquence binaire émise. La **démodulation numérique** consiste à prendre une décision concernant le symbole représenté par le signal au temps t_0 pendant la durée T_s : est-ce un "0" ou un "1" qui a été émis ?

Le problème est tout à fait différent de celui de la transmission des signaux analogiques. Lors de la transmission de signaux analogiques, on cherche à avoir le meilleur rapport S/B à la réception pour reconstituer le signal analogique émis. Dans le cas de la transmission de signaux numériques, il faut reconstituer la séquence binaire émise et donc prendre la bonne décision : est-ce un "1" ou un "0" ? La performance de la chaîne de communication numérique se mesure en taux d'erreurs binaires (nb d'erreurs / nb de bits transmis).

1.2 Modulation des signaux numériques

1.2.1 Codage en ligne

Les symboles "0" et "1" sont représentés par un signal physique, qui peut être une tension, un courant, dont l'unité est le V, le mV, A, mA, ...

On étudiera les codages NRZ, RZ et Manchester pour représenter la séquence binaire.

Le signal transmis par un canal de communication est filtré (bande passante limitée) et atténué. Pour assurer une bonne transmission, on utilise des répéteurs qui amplifient le signal mais qui ont de mauvaises performances en basses fréquences.

Un code est d'autant plus intéressant que sa largeur de bande est faible. Lorsqu'on utilise des répéteurs, le code ne doit pas avoir de composante continue (énergie à fréquence nulle). Dans certain cas, l'information de la cadence d'émission des symboles est présente dans le code ; cette information peut être intéressante pour une bonne synchronisation en réception.

Nous étudierons quelques codes en ligne et leur intérêt pour la transmission des signaux. La figure ci-dessous rappelle les différents codes que l'on peut utiliser pour représenter les symboles 0 et 1.

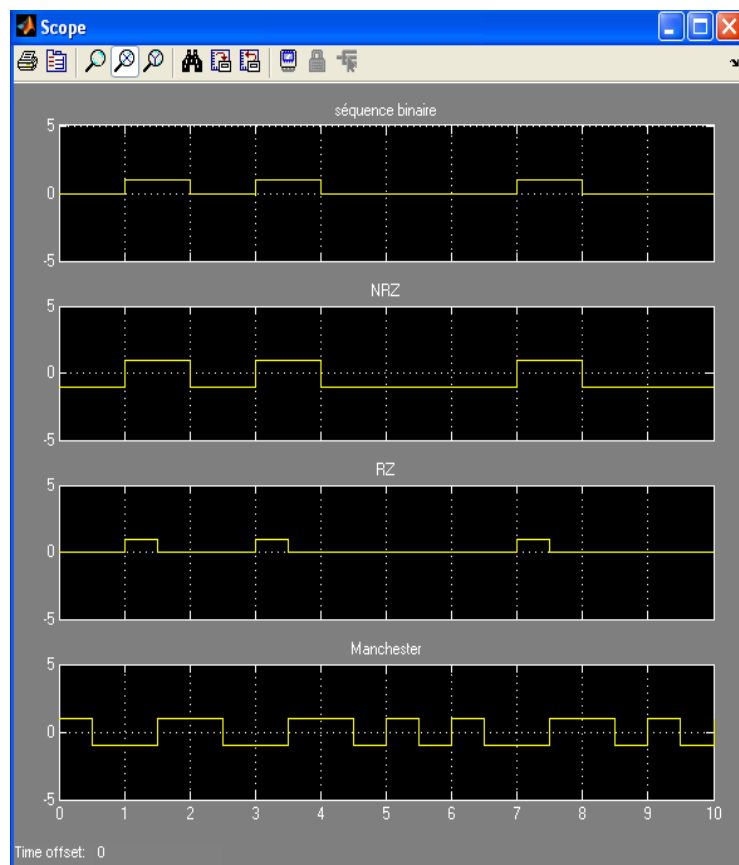


FIGURE 1.1 – Les différents codes utilisés pour représenter les symboles

1.2.2 Réalisation du modèle "codage en ligne"

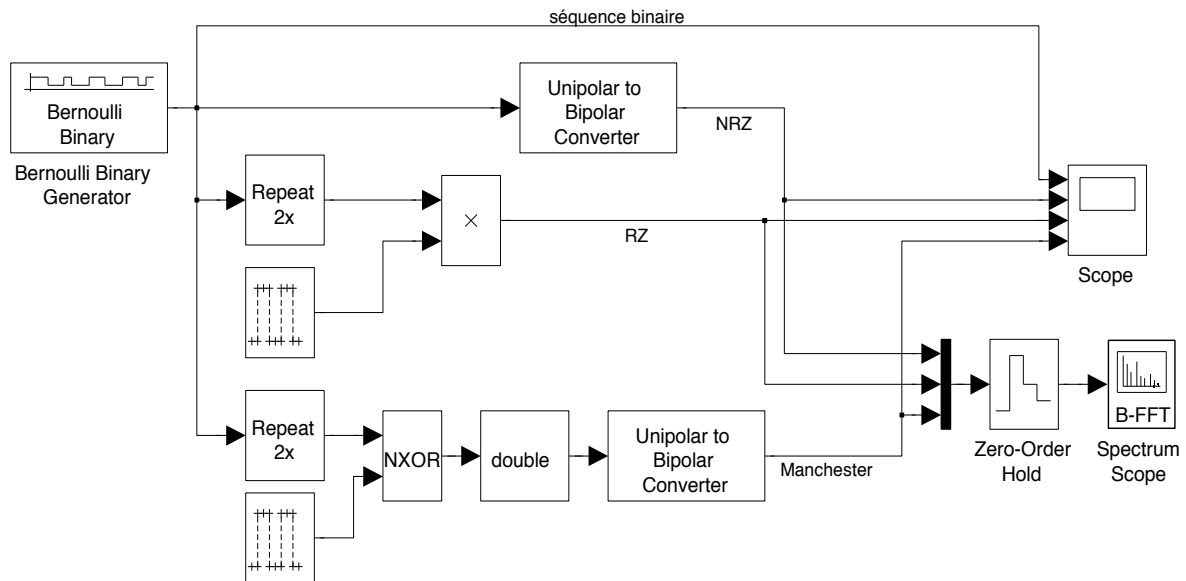


FIGURE 1.2 – Les différents codage en Bande de Base : modèle Simulink

Le codage NRZ

Le codage NRZ transforme :

- le symbole 0 en un signal de $-1V$ pendant la durée T_s d'émission du symbole,
- le symbole 1 en un signal de $+1V$ pendant T_s .

Exercice 1 Codage NRZ

Réaliser le modèle Simulink NRZ : 1^{ière} ligne de la figure 1.2.

- Séquence binaire de symboles :

Cette séquence de 1 et 0 est générée par un générateur de séquences aléatoires.

Le bloc *Bernoulli Binary Generator* que l'on trouve dans le *Communication System Toolbox/Comm Sources/Random Data Sources* permet de générer aléatoirement des 1 et des 0 qui apparaissent tous les T_s secondes. Pour cela, fixer :

★ *Probability of zeros* = 0.5 (autant de zéros que de un)

★ *seed* = 25. Le *seed* initialise la séquence aléatoire.

★ *sample time* = 1. Les 1 et les 0 apparaîtront toutes les $T_s = 1$ seconde.

- Le signal NRZ :

On utilise le bloc *Unipolar to Bipolar Converter* du *Communication System Toolbox/Utility Blocs*. Ce bloc prend M valeurs de même polarité (positive ou négative) et les transforme en $M/2$ valeurs de polarité positive et $M/2$ valeurs de polarité négative.

M est bien évidemment paire : il représente le nombre de symboles utilisés pour représenter la séquence binaire. Si les symboles sont constitué d'un bit, $M = 2$; si les symboles sont constitués de n bits, $M = 2^n$. Lorsqu'on a deux symboles, "0" et "1", $M = 2$. Ce bloc permet donc de générer un signal analogique de durée $T_s = 1s$ qui vaut $+1V$, si le symbole d'entrée vaut 1, ou $-1V$, si le symbole d'entrée vaut 0.

Il faudra donc fixer dans ce bloc :

- M-ary number = 2,
- Polarity : positive,
- Output data type : Inherit via internal rule.
- Visualisation des signaux :
 - La séquence de symboles et le signal NRZ seront observés en **fonction du temps** avec un scope (régler le *stop time* = 10 pour observer 10 symboles).
 - Le spectre du signal NRZ sera observé en **fonction de la fréquence** avec un analyseur de spectre :
 - ★ on utilise un échantillonneur bloqueur *Zero-Order Hold* dont la période d'échantillonnage T_e est fixée de façon à observer les signaux entre $0Hz$ et $+5Hz$. Il faut donc que la fréquence d'échantillonnage $F_e = 10Hz$ et donc, la période d'échantillonnage, $T_e = 0.1s$.
 - ★ l'analyseur de spectre, *Scope FFT*, sera configuré avec (voir figure 1.3) :
 - Scope Properties :
 - Spectrum units : Watts/Hertz
 - Spectrum type : one-side ([0 ... $F_s/2$])
 - Buffer input coché
 - Buffer size : $128*4$
 - Buffer overlap : 64
 - Window : Hann
 - Treat Mx1 and unoriented sample based signals as : M channels
 - Window sampling : periodic
 - Number of spectral averages : 200
 - Axis Properties :
 - décocher la case *Inherit sample time from input*
 - Sample time of original time series : identique à T_e
 - Frequency display offset (Hz) : 0
 - Frequency display limits : Auto
 - Y-axis label : Magnitude-squared, dB

Pour observer la représentation spectrale, il faut que le signal soit suffisamment long : fixer le *stoptime* égal à *inf* et arrêter la simulation quand la représentation n'évolue plus.

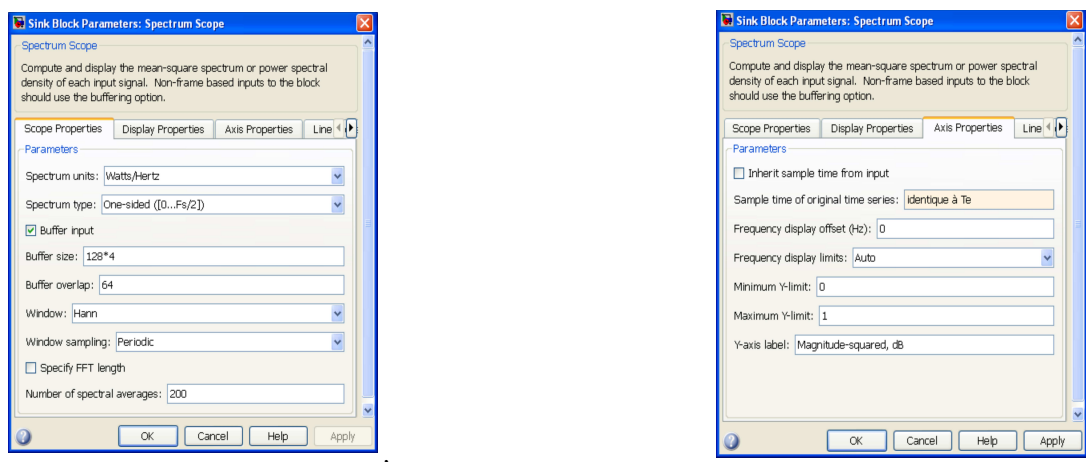


FIGURE 1.3 – Réglage de l'analyseur de spectre

Observation des signaux :

1. Représenter le codage NRZ de la séquence binaire en fonction du temps (entre 0 et 10s).
2. Représenter le spectre du signal. Faire varier le temps T_s (1s, 2s, 4s) et tracer les représentations spectrales. Quelles remarques faites vous sur cette représentation spectrale. Dans quelle type de liaison sera utilisé le codage NRZ ?

Le codage RZ

Le codage RZ transforme :

- le symbole 0 en un signal de 0V pendant une durée T
- le symbole 1 en un signal qui présentera une transition descendante à $T_s/2$: il vaut +1V pendant $T_s/2$ puis 0V pendant $T_s/2$.

Exercice 2 Codage RZ

Réaliser le modèle Simulink RZ : 2^{ème} ligne de la figure 1.2.

- Compléter le modèle précédent et utiliser le bloc *Bernouilli Binary Generator*
- Placer une horloge "Générateur de Pulse". On trouve le Bloc *Pulse Generator* dans la librairie *Simulink/Sources*. Configurer ce générateur de façon à avoir un signal de période $T_s/2$ (T_s étant la cadence d'envoi des symboles, fixée dans le bloc *Bernouilli Binary Generator*. Si $T_s = 1$, fixer la fréquence d'échantillonnage de ce bloc à 0.5 s, dont l'amplitude varie entre 0 et 1
 - pulse type : sample time (on travaille avec des signaux échantillonnés)
 - time : use simulation time
 - amplitude : 1
 - période (number of samples) : 2
 - pulse width : 1
 - phase delay : 0
 - sample time : 0.5
- Multiplier la séquence aléatoire par ce Générateur de Pulses. Le Générateur de Bernouilli produit 1 échantillon toutes les secondes et le Générateur de Pulses en produit 2. Insérer le bloc *Repeat* que l'on trouve dans la librairie *DSP System Toolbox/Signal Operations* pour que les signaux aient la même cadence.
- Utiliser le bloc *Product* pour multiplier le signal en sortie des blocs *Repeat* et *pulse Generator*.

Observation des signaux

- Rajouter une entrée au scope pour observer la séquence binaire, le codage NRZ et le codage RZ
 - Utiliser le multiplexeur *Mux* que l'on trouve dans la librairie *Simulink/Signal Routing* pour comparer les signaux sur le scope FFT (fig 1.2).
Pour distinguer les deux courbes, vous pouvez attribuer des couleurs à chaque courbe.
1. Représenter en fonction du temps (10s) la séquence binaire et le signal en sortie du multiplieur, qui est le codage RZ de la séquence binaire.
 2. Observer ce signal en fonction de la fréquence (*stoptime = inf*). Observations ? Faire varier T_s (1s, 2s, 4s) et vérifier la formule de Bennett. Attention, lorsque T_s change, n'oubliez pas de changer la période du *Pulse Generator* qui vaut $T_s/2$.
 3. Quel est l'intérêt du codage RZ par rapport au codage NRZ ? Son désavantage ?

Le code Manchester

Le code Manchester transforme

- le symbole 0 en un signal de durée T_s qui présente une transition descendante à $T_s/2$: il vaut +1V pendant $T_s/2$ puis -1V pendant $T_s/2$.
- le symbole 1 en un signal de durée T_s qui présente une transition montante à $T_s/2$: il vaut -1V pendant $T_s/2$ puis +1V pendant $T_s/2$.

Exercice 3 Code Manchester

Réaliser le modèle Simulink Manchester : 3^{ième} ligne de la figure 1.2.

Pour cela, télécharger et configurer le schéma Simulink Manchester. .

Observation des signaux

- Rajouter une entrée au scope pour observer la séquence binaire, le codage NRZ, le codage RZ et le codage Manchester.
 - Utiliser un scope FFT pour observer la représentation spectrale du signal obtenu par codage NRZ, codage RZ et codage Manchester.
 - Lancer la simulation
 - fixer le *stoptime* égal à 12 et observer les représentations temporelles
 - fixer le *stoptime* à *inf* pour observer les représentations fréquentielles.
1. Rappeler la table de vérité du NXOR (OU exclusif). A l'aide d'un chronogramme, expliquer le codage Manchester.
 2. Représenter les signaux observés en temps et en fréquence.
 3. Quel est l'avantage du code Manchester ? pourquoi ce type de code est adapté pour la transmission à longue distance (Ethernet, etc.) ?

1.3 Démodulation : cas d'un canal de transmission bruité

Nous considérons dans cette simulation que le canal de transmission a une **bande passante infinie** et que la seule perturbation que subit le signal est l'ajout de bruit. Nous allons étudier l'influence de ce bruit sur la réception du signal en mesurant un taux d'erreur. Pour cela on utilisera une séquence binaire représentée par un signal codé Manchester auquel on additionne un bruit blanc gaussien pour simuler le signal reçu.

1.3.1 Méthode du corrélateur

On suppose connu le codage utilisé à l'émission pour transmettre la séquence binaire (codage RZ, NRZ, Manchester, ...), et donc les formes d'onde $s_1(t)$ et $s_0(t)$ représentant les symboles "1" ou "0". Le débit binaire est connu (ou estimé à la réception) ce qui permet d'estimer la durée d'émission T_s de chaque symbole.

La méthode du corrélateur s'utilise généralement pour des codages tels que $s_1(t) = -s_0(t)$. Pour reconstituer la séquence de symboles émise, on multiplie le signal reçu $r(t)$ par la forme d'onde $s_1(t)$ et on intègre pendant la durée T_s d'émission du symbole : $A = \int_{t_0}^{t_0+T_s} r(t)s_1(t)dt$.

Le signe de l'intégrale est estimé par un comparateur à seuil zéro (organe de décision) : s'il est positif ($A > 0$), on décidera que le symbole reçu est un "1" ; s'il est négatif ($A < 0$), ce sera un "0".

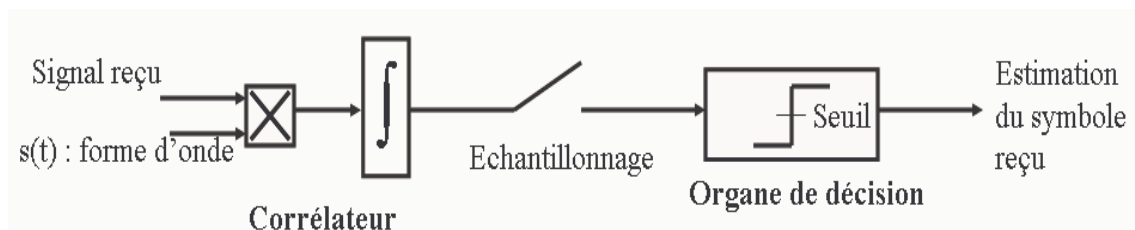


FIGURE 1.4 – Principe démodulation des signaux numériques

1.3.2 Réalisation de la démodulation en bande de base

Télécharger le modèle **figure15**. Le bloc *Manchester* utilisé pour l'émission est celui que vous avez déjà réalisé. Ce bloc est configuré pour émettre des signaux avec un débit binaire de 1bit/s en code Manchester. Vous pouvez double-cliquer sur ce bloc pour voir comment il est constitué.

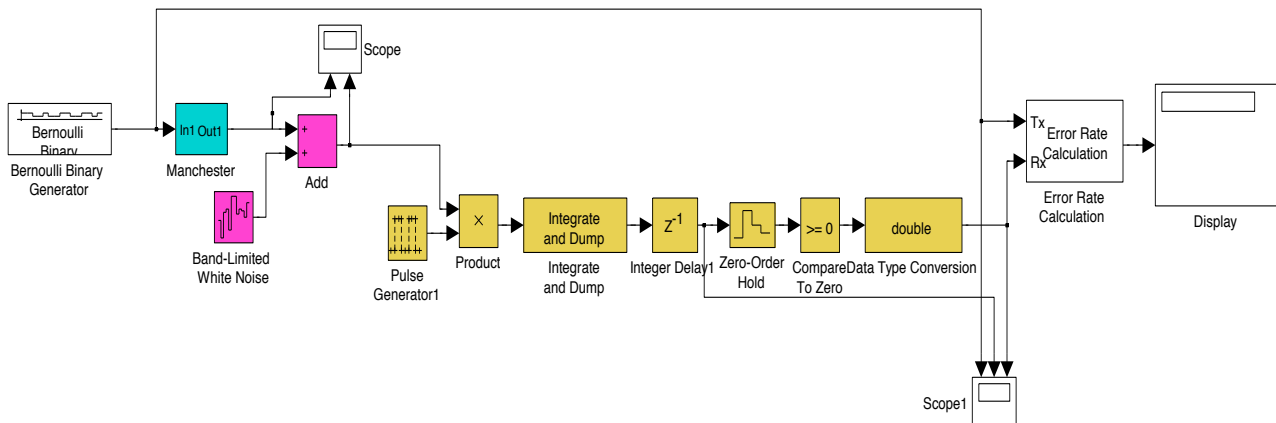


FIGURE 1.5 – Démodulation d'un signal codé Manchester bruité

Configurer la partie démodulation du modèle afin d'obtenir les caractéristiques suivantes :

- **Canal bruité**

La puissance du bruit dans le canal est fixée à 10^{-2} . Le bruit est réalisé par le bloc *Band Limited White Noise*.

- **Démodulation**

Pour réaliser la démodulation, utiliser un corrélateur :

- Le générateur d'impulsions doit être identique à celui utilisé pour l'émission.
- L'intégrateur (bloc *Integrate and Dump*) permet de sommer les échantillons du signal pendant le temps symbole $T_s = 1s$. Cocher la case *Output intermediate value* pour visualiser l'intégration pendant la durée du symbole.

Exercice 4 Démodulation : canal bruité

1. Visualisation des signaux reçus

Visualiser la séquence binaire transmise et la séquence reçue (*stop time = 12*). Déterminer la valeur du retard entre la séquence émise et la séquence reçue.

Utiliser les formats *Port/Signals Displays* pour visualiser les caractéristiques des signaux du modèle. Le *Sample time color* permet de visualiser les différentes fréquences d'échantillonnage des signaux. Le *Port Data Types* permet de connaître le format des signaux.

Le bloc *Data Type Conversion* permet de convertir les données sortant du comparateur en un format compatible avec les autres blocs utilisés dans le modèle.

2. Mesure du taux d'erreur

Pour analyser de longues séquences (10 000 points) et calculer un taux d'erreur, on utilise un compteur d'erreur, le bloc *Error Rate Calculator* qu'il vous faudra configurer.

Faire la simulation avec *stop time = 1e4*. Vérifier qu'il n'y a pas d'erreur en réception car le bruit est très faible.

Mesurer le taux d'erreur pour une puissance du bruit de $1e-3$, $1e-2$, $1e-1$ et 1 . Commenter.

TP 2

odulation/démodulation sur fréquence porteuse

2.1 Avant-propos

Au cours du TP précédent, nous avons vu comment émettre et recevoir une séquence binaire composée de symboles d'un bit "0" et "1".

Pour améliorer le débit binaire de la transmission on a intérêt à considérer des symboles composés de plusieurs bits (augmentation de l'efficacité spectrale).

De plus, pour s'adapter aux caractéristiques du canal, l'information sera portée par la fréquence F_p , la fréquence porteuse.

Par exemple, la séquence binaire 1011100111110001, peut être vue comme une séquence de 16 symboles de longueur 1 bit, ou comme une séquence de 8 symboles de longueur 2 bits valant respectivement $s_1 \rightarrow "00"$, $s_2 \rightarrow "01"$, $s_3 \rightarrow "10"$, $s_4 \rightarrow "11"$: pendant le même temps d'émission du symbole T_s , on transmet alors 2 bit ; le débit binaire est 2 fois plus grand. Le prix à payer sera la complexité du système pour réaliser la modulation/démodulation, donc une augmentation du taux d'erreur binaire.

La modulation d'une séquence binaire composée de symboles de longueur n peut être réalisée par une modulation ASK (Amplitude Shift Keying), PSK (Phase Shift Keying), QAM (Quadrature Amplitude Modulation) ou FSK (Frequency Shift Keying) qui n'est pas abordée dans le cours. Elle est caractérisée par la longueur des symboles : pour des symboles de longueur n bits, il y a $M = 2^n$ symboles différents. La modulation est appelée ASK-M, PSK-M, QAM-M ou FSK-M.

Par exemple, pour des symboles de longueur 2 bits, ce sera une modulation QAM-4, car il y a $2^2 = 4$ symboles différents. Lorsqu'on utilisera des symboles de longueur 6 bits, ce sera une modulation QAM-64.

La modulation QAM-64 (6 bits par symbole) est largement utilisée dans les modems, et dans d'autres formes de communications numériques haut débit. On utilise en France une modulation QAM-64 pour l'émission de la TNT. La modulation QAM-256 est fréquemment utilisée pour la télévision numérique par câble et dans les modems.

2.2 Etude d'une transmission à travers un canal à bande infinie : modulation/démodulation

On suppose que le canal de transmission est bruité, de **bande passante infinie**.

Exercice 1 Modulation de type ASK-2 sur fréquence porteuse

On s'intéresse dans un premier temps à la modulation d'un train binaire ($M = 2$) avec un codage en ligne Manchester.

Télécharger le modèle **figure21**.

Le bloc *Manchester* réalise une modulation de type Manchester (voir TP1). Double cliquer sur ce bloc pour voir les éléments qui le composent.

Configurer le modèle sachant que :

- ★ le débit binaire est de 1bit/s.
- ★ la probabilité d'apparition des zéros est de 0,5 (autant de zéros que de 1).
- ★ la fréquence porteuse de la modulation est de 5Hz.

Observer les signaux, en fonction du temps et de la fréquence. Commenter.

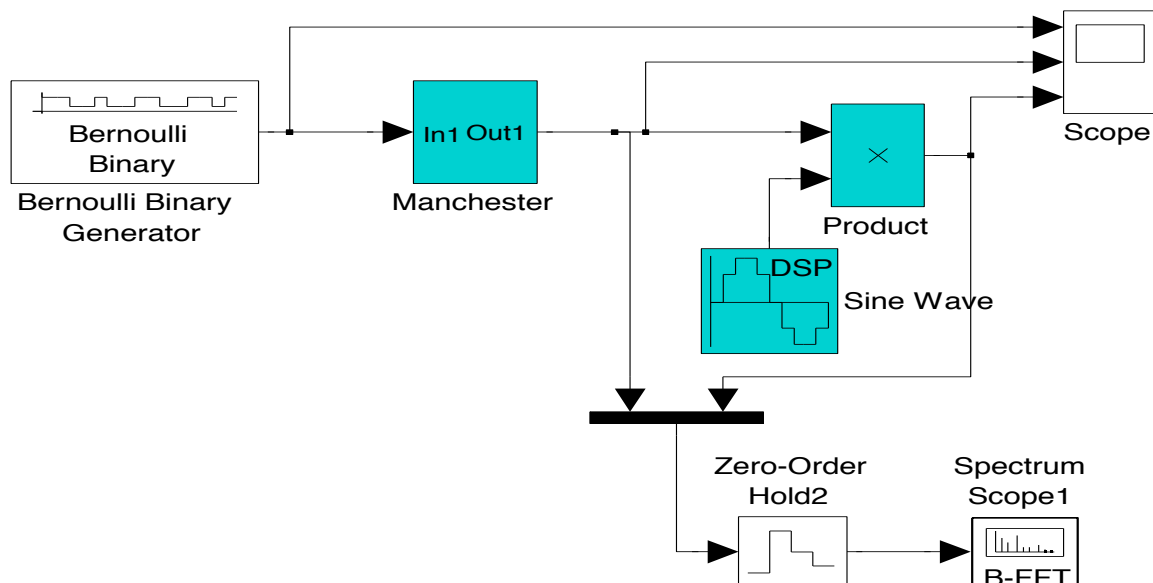


FIGURE 2.1 – Modulation ASK-2 sur fréquence porteuse.

Exercice 2 Démodulation ASK-2 sur fréquence porteuse

Télécharger et configurer le modèle **figure22**.

Sur ce modèle, le bloc modulation représente le modèle réalisé précédemment.

La démodulation consiste à multiplier le signal modulé par la même porteuse en phase et à filtrer passe-bas (le filtre est déjà configuré). L'intégration du signal pendant le temps symbole T_s , associée à un comparateur, permet de prendre une décision sur la valeur du symbole émis (0 ou 1).

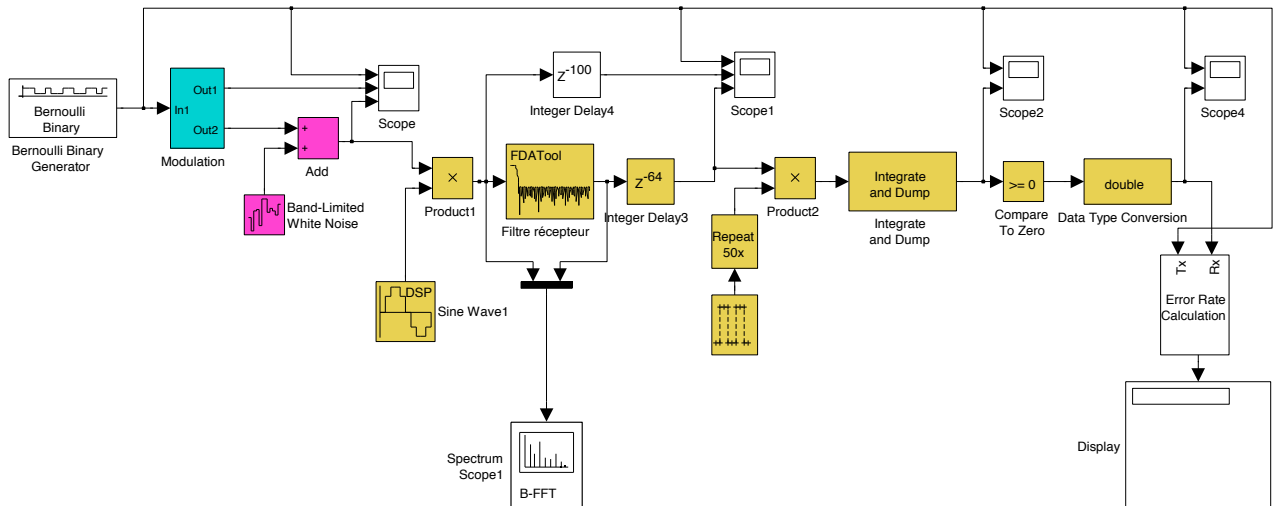


FIGURE 2.2 – Modulation/démodulation ASK-2 sur fréquence porteuse.

1. Observer et commenter les représentations temporelle et fréquentielle des signaux.
2. Quel est le rôle du bloc *Integrate Delay* placé après le filtre ?
3. Déphaser l'oscillateur du démodulateur entre 0 et $\frac{\pi}{2}$. Observer le signal en sortie de l'intégrateur ? Déduire un résultat mathématique important sur l'orthogonalité des fonctions sinusoïdales.

2.3 Transmission de symboles de n bits

Exercice 3 Modulation/démodulation en quadrature : QAM-4

Supposons que l'on dispose de deux trains binaires différents. En exploitant la propriété d'orthogonalité des fonctions sinusoïdales, il est possible de démoduler 2 trains binaires différents. Les symboles seront alors composés de 2 bits.

Télécharger le modèle **figure25**.

Configurer le modèle. Choisir 2 valeurs de *seed* différents pour les 2 générateurs. Le bloc *Bit to Integer* permet de convertir 2 bits en un entier (compris entre 0 et 4). Penser à déphaser de $\frac{\pi}{2}$ l'un des 2 oscillateurs du modulateur pour générer des cosinusoides ou des sinusoides (le faire également pour la partie démodulation).

1. Expliquer pourquoi il faut 2 seed différents dans les générateurs de Bernoulli.
2. Ecrire l'équation du signal en sortie du modulateur.
3. Ecrire l'équation du signal en sortie du démodulateur.
4. Mesurer la valeur du bruit pour laquelle on commence à avoir des erreurs.
5. Changer le signe de la deuxième entrée de l'additionneur du modulateur et observer l'évolution du taux d'erreur.

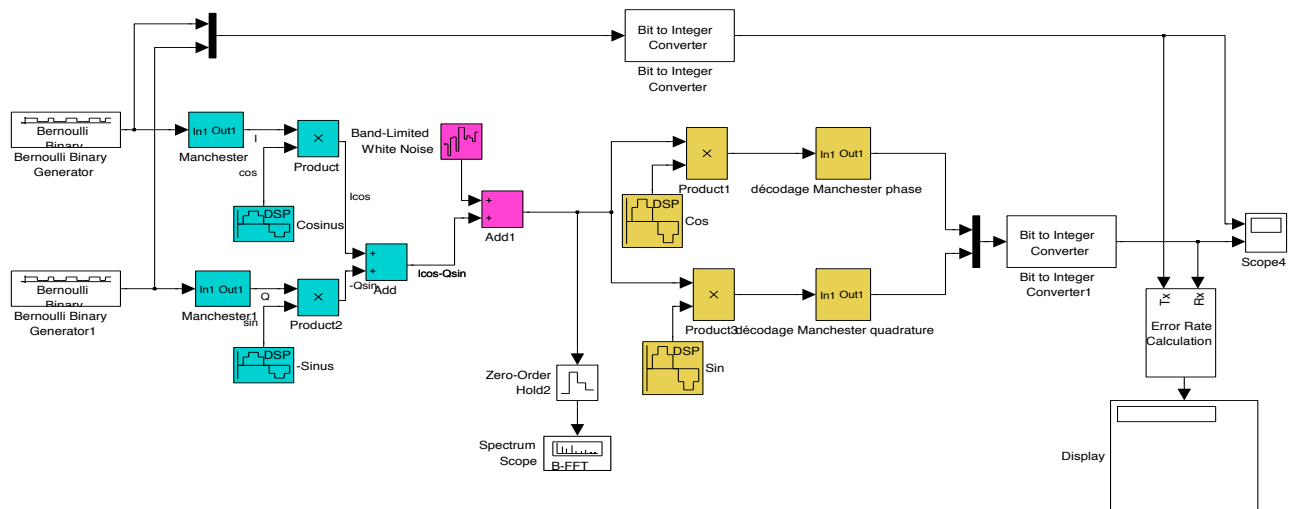


FIGURE 2.3 – Modulation/démodulation de phase en quadrature de 2 trains binaires.

2.4 Représentation complexe du signal

Exercice 4 Modulation de phase en quadrature

On définit l'enveloppe complexe d'un signal par $\alpha(t) = I(t) + jQ(t)$, où $I(t)$ désigne la composante en phase et $Q(t)$ celle en quadrature.

Télécharger le modèle **figure26** qui réalise la modulation/démodulation QAM-4 en utilisant une représentation complexe du signal.

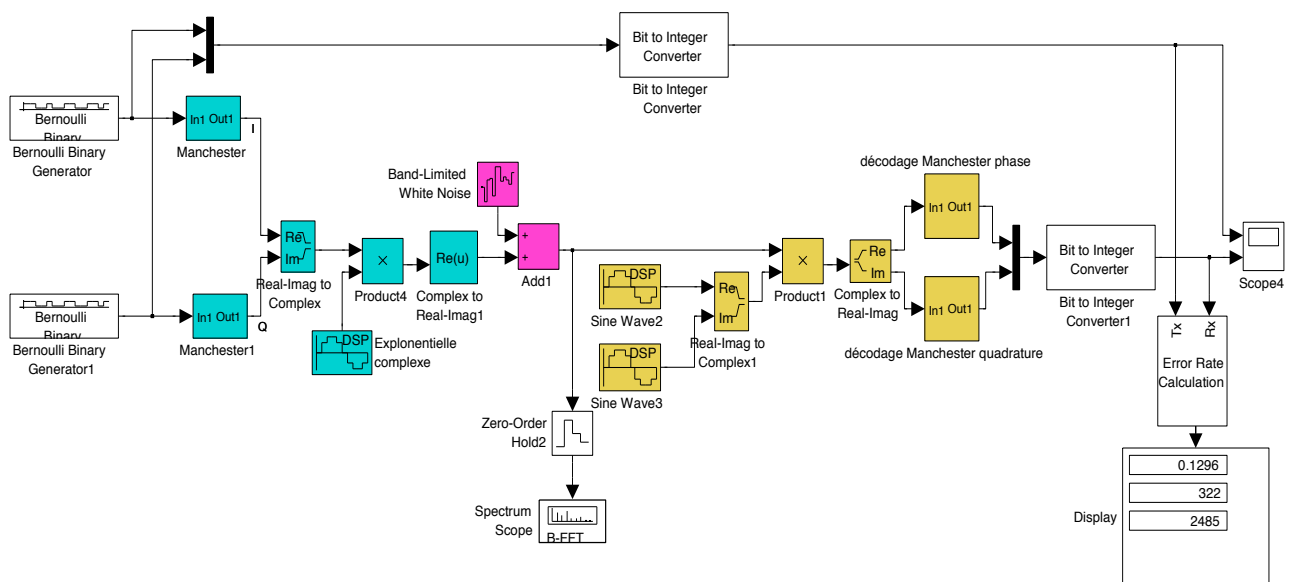


FIGURE 2.4 – Comparaison d'une modulation/démodulation de phase par représentations réel et complexe du signal.

1. Comparer le taux d'erreur mesuré sur ce modèle avec celui mesuré sur le **figure25** (pour les mêmes réglages).
2. Conclure sur l'intérêt d'une représentation complexe pour la simulation d'une modulation de type QAM.

TP 3

chaîne de communication

3.1 Avant-propos

L'objectif de ce TP est d'étudier l'impact des principales perturbations rencontrées lors d'une transmission Wi-Fi réelle (canal à bande limitée, bruit, ...) en terme de taux d'erreur en réception.

IEEE 802.11 est un ensemble de normes concernant les réseaux sans fil édictées sous l'égide du comité de standardisation IEEE 802. Ces normes constituent un tout cohérent servant de base de travail aux constructeurs développant des équipements et les services chargés de l'implémentation des infrastructures réseaux à liaison filaire et sans fil.

TABLE 3.1 – 802.11x

Protocole	Date	Fréquence	Taux transfert (Typ)	Taux transfert (Max)	Portée (Intérieur)	Portée (Extérieur)
Legacy	1997	2.4-2.5 GHz	1 Mbit/s	2 Mbit/s	?	?
802.11a	1999	5.15-5.35 5.47-5.725 5.725-5.875 GHz	25 Mbit/s	54 Mbit/s	25 m	75 m
802.11b	1999	2.4-2.5 GHz	6.5 Mbit/s	11 Mbit/s	35 m	100 m
802.11g	2003	2.4-2.5 GHz	25 Mbit/s	54 Mbit/s	25 m	75 m
802.11n	2009	2.4 GHz ou 5 GHz	200 Mbit/s	54 Mbit/s	50 m	125 m
802.11y	2008	3.7 GHz	23 Mbit/s	54 Mbit/s	50 m	5000 m

3.2 Emission/réception QAM-M (ou PSK-M) sur fréquence porteuse

Exercice 1 Emission sur fréquence porteuse

Télécharger le modèle **figure31** qui permet de simuler une transmission respectant la norme Wi-Fi IEEE 802.11 avec un débit de 54Mbits/s sur fréquence porteuse de $F_0 = 5.10^9\text{ Hz}$.

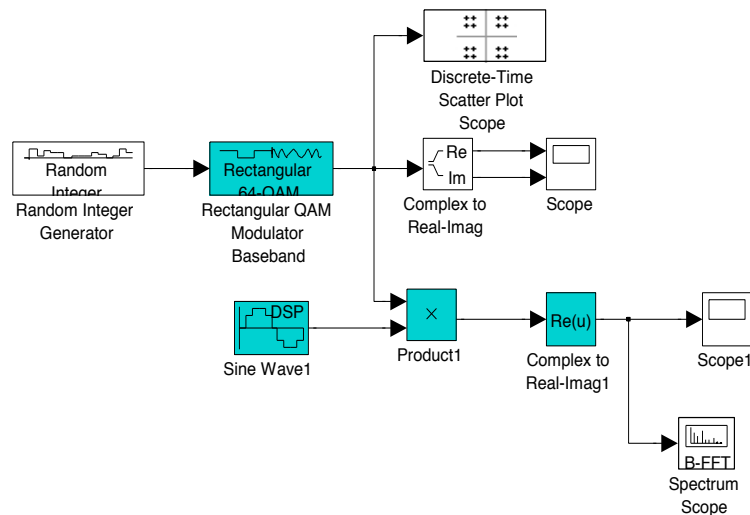


FIGURE 3.1 – Modulation QAM-64 sur fréquence porteuse.

1. Sachant que la bande passante du canal $B = 20\text{MHz}$ ($B \geq \frac{2}{T_S} = \frac{2}{n * T_B}$), quelle est la longueur optimale des symboles (nombre de bits) ?
2. Quelle doit être la fréquence d'échantillonnage de la porteuse ?
3. Configurer le modèle permettant de simuler une transmission Wi-Fi.
4. Que pensez-vous du temps de calcul pour la simulation de cette transmission ?

Exercice 2 Réception sur fréquence porteuse

Télécharger le modèle **figure32** qui est déjà configuré.

1. Faire varier le rapport signal/bruit dans le canal de transmission (bloc *AWGN Chanel*) et observer les constellations et le taux d'erreur binaire. Commentaires.
2. Quelle est la valeur minimale entière du rapport S/B pour laquelle il n'y a pas d'erreur de transmission ?
3. Recommencer la simulation en utilisant une modulation PSK-n (blocs *General QAM* en modulation et démodulation). Commentaires.

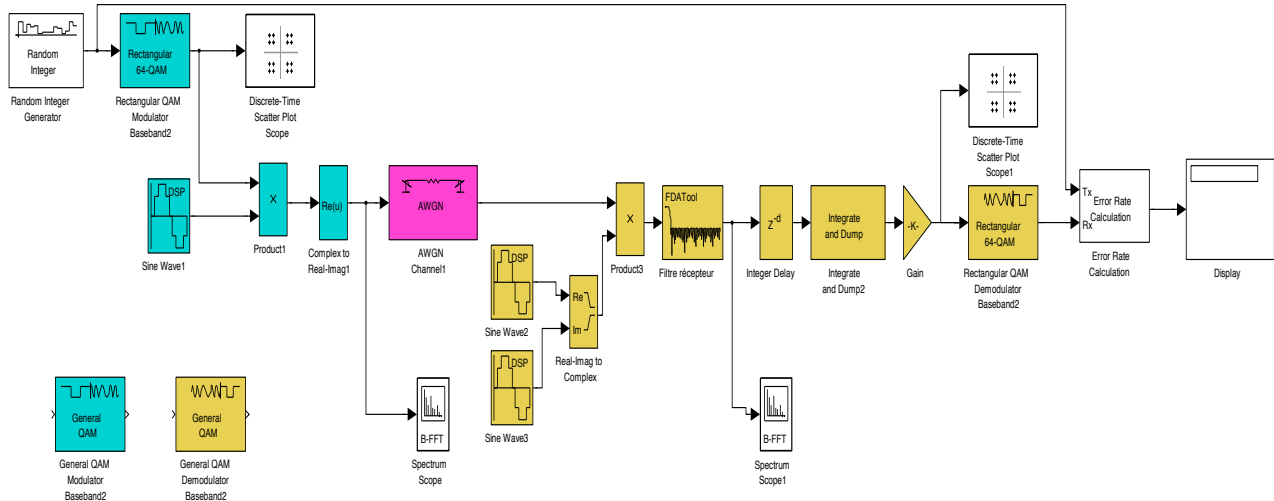


FIGURE 3.2 – Transmission Wi-Fi IEEE 802.11 sur fréquence porteuse

3.3 Emission/réception QAM-64 (ou PSK-64) en bande de base

Exercice 3 Modulation sur fréquence porteuse

Télécharger le modèle *figure33* qui est une simulation d'une transmission Wi-Fi IEEE 802.11 en bande de base.

Faire varier les paramètres du modèle et mesurer les taux d'erreur. Conclusions. -

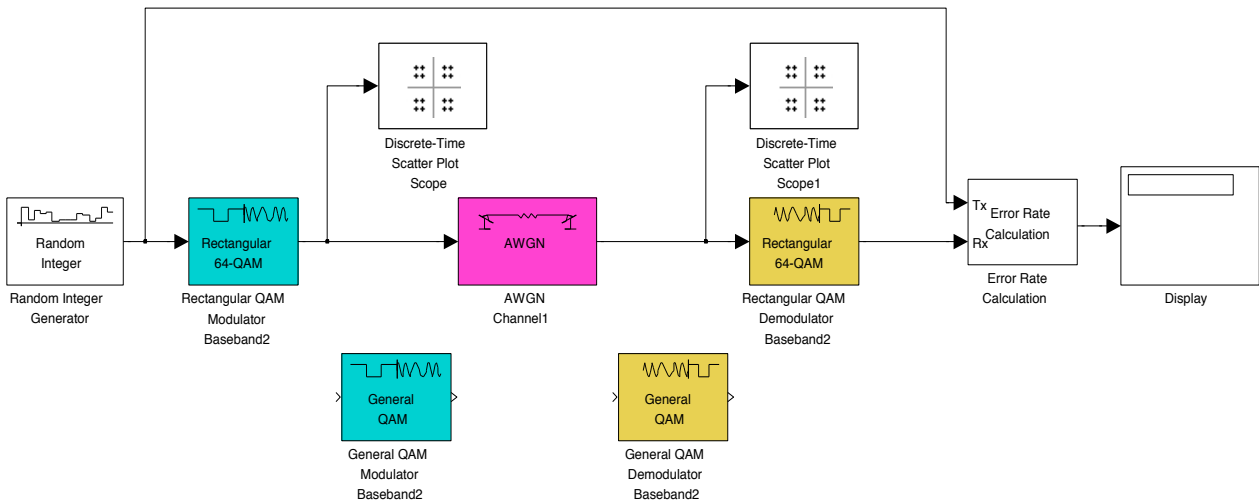


FIGURE 3.3 – Transmission Wi-Fi IEEE 802.11 en bande de base