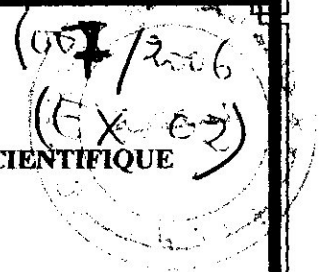
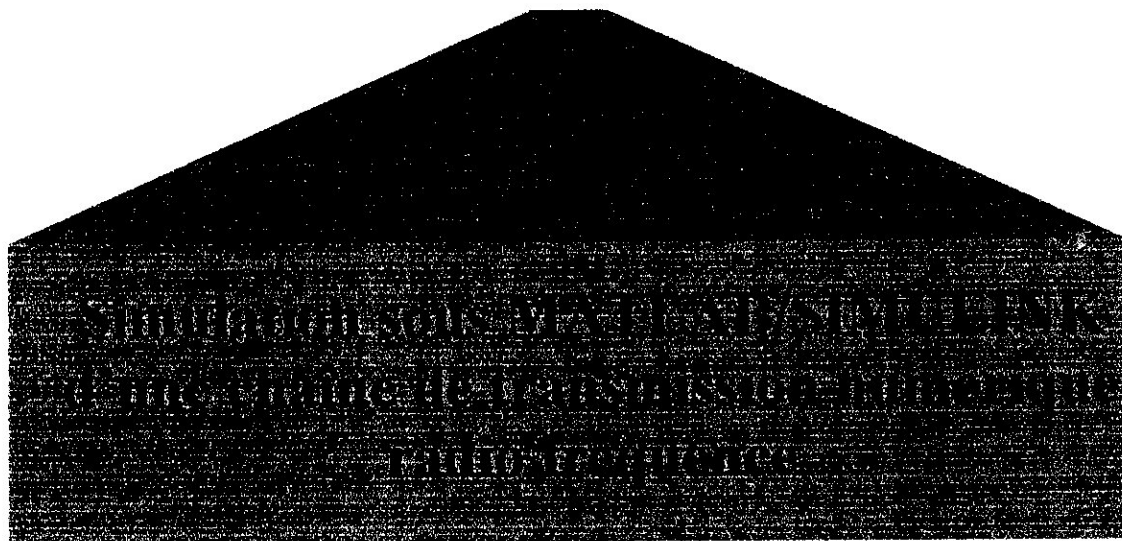


REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE



Université Saad Dahleb de Blida
Faculté des sciences de l'ingénieur
Département d'aéronautique

Projet de fin d'études
En vue de l'obtention du diplôme d'ingénieur d'état en
Aéronautique (option installations)



Réalisé par

EL AISSA Hamed

encadré par :

Mr. HELAL Mahmoud

PROMOTION 2005/2006

أهلاً وسهلاً
عُماًشياً

إلى القلعة الصامدة في وجه مؤامرات الحاقدين ...
إلى الصخرة التي تكسرت عليها سكاكين الغدر و
الخيانة.....
إلى الينبوع الذي سقا ربيع شبابي.....

إلى وطني الحبيبسوريا

حمد العيسى

البلدة في 2006/07/25

شكر

أقدم بالشكر لكل فرد كان له دور في حياتي وأخص بالذكر:
الإنسان الذي كد وجد لأكبر وأستزيد من العلم أبي الغالي.
الشمعة التي احترقت لتضيء الطريق أمامي أمي الحنون.
الدعامة التي ساندتني معنويا طوال سنوات الدراسة عائلتي
الكريمة.

العين التي سهرت على راحتي واليد التي بسطت لي بساط
الطمأنينة و السعادة زوجتي.

مصابيح النور التي فتحت عيني على آفاق جديدة في ميادين
العلم أساتذتي الأجلاء.

شكر خاص للأستاذ محمود هلال على الجهود الجبارة التي
بذلها معي لانجاز هذا العمل.

و الى كل من مد لي يد العون من قريب او بعيد.

حمد العيسى

خلاصة

إن الهدف من هذه المذكرة تسليط الضوء على تقنية تعديل فعالة في النقل الرقمي المسماة QAM 16، باستعمال أداة المحاكاة في لغة البرمجة مطلب. من أهم خصائص هذه التقنية المعدل الثنائي المرتفع بالمقارنة مع تقنيات مستعملة حاليا مثل PSK، مع نفس التردد للرموز. يمكن استعمال هذه التقنية في مختلف الميادين، في نقل المعطيات أو الصوت.

Résumé

L'objectif de ce mémoire est une description d'une technique de modulation efficace en transmission numérique dite QAM 16, en utilisant l'outil de Matlab « Simulink ».

La propriété la plus importante est le débit binaire élevé en comparaison avec des autres techniques utilisées comme PSK, avec la même fréquence de symbole.

Cette technique peut être utilisée dans plusieurs domaines, pour la transmission des données ou de voix.

Abstract

The objective of this memory is a description of an efficient modulation technique in digital transmission so-called QAM 16, by using the tool of Matlab " Simulink ".

The most important property is the binary debit raised in comparison with another used techniques as PSK, with the same frequency of symbol.

This technique can be used in several domains, for the transmission of the data or voice.

TABLE DES MATIERES

	Pages
INTRODUCTION GENERALE	1
I. Modulation numérique	
I.1 Introduction	2
I.2 C'est quoi la modulation numérique et pourquoi ?	3
I.3 Définitions et appellations	4
I.4 Principe des modulations numériques	5
I.5 Modulation par déplacement d'amplitude (MDA)	7
I.5.1 Modulation par tout ou rien	8
I.5.2 Modulation à "M ETATS"	9
I.6 Modulation par déplacement de phase (MDP)	11
I.7 Modulation d'amplitude sur deux porteuses en quadrature (MAQ)	17
I.8 Modulation par déplacement de Fréquence (MDF).	22
I.9 Applications	23
I.10 Conclusion	24
II. Présentation du logiciel MATLAB/SIMULINK	
II.1 Introduction	25
II.2 La structure du SIMULINK	25
II.3 Description des blocs de Simulink	29
II.3.1 La bibliothèque « Simulink »	29
II.3.1.1 Bloc Sine wave (générateur du sinus)	29
II.3.1.2 Bloc Constant (Constant)	30
II.3.1.3 Bloc product (Produit)	30
II.3.1.4 Bloc Sum (somme)	31
II.3.1.5 Quantizer (Quantificateur)	32
II.3.1.6 Bloc Mux (multiplexeur)	32
II.3.1.7 Demux (Demultiplexeur)	33
II.3.1.8 Bloc Scope (Oscilloscope)	34
II.3.1.9 Bloc To Workspace.	36
II.3.1.10 Display (display)	37
II.3.1.11 Zero-order hold (Bloqueur d'ordre zero)	37
II.3.1.12 Bloc Terminator (Termineur)	38
II.3.1.13 Bloc transport delay	38
II.3.1.14 Sign (Signe)	39
II.3.1.15 Bloc de Saturation (Saturation)	40
II.3.1.16 Bloc Inport (Bloc d'entrée)	40
II.3.1.17 Bloc Outport (Bloc de sortie)	41
II.3.2 La bibliothèque « communications »	42
II.3.2.1 Random integer generator (Générateur des nombres entiers aléatoires)	42
II.3.2.2 Bit to integer converter (Convertisseur bits _entier).	42
II.3.2.3 Integer to bit converter (Convertisseur entier-bits)	43
II.3.2.4 AWGN (Additive White Gaussian Noise) channel (canal AWGN).	44
II.3.2.5 PLL (boucle à verrouillage de phase).	45
II.3.2.6 Error rate calculation (Calcul du Taux de l'erreur)	46

II.3.3 La bibliothèque « DSP » Digital Signal Processing (traitement numérique du signal)	47
II.3.3.1 Analog filter design (Dessin du Filtre analogique)	47
II.3.3.2 Simple and hold (échantillonneur- bloqueur).	48
II. 4 Création des schémas blocs	49
II. 5 Ajustement des paramètres de la simulation	50
II.6 conclusion	51

III. Simulation sous MATLAB/SIMULINK d'une chaîne de transmission numérique radio fréquence

III.1 Introduction	52
III.2 Présentation de la chaîne de transmission	52
III.2.1 Emetteur	52
III.2.2. Récepteur	53
III.3 Gamme de fréquences	54
III.4 Description du travail sous MATLAB	54
III.4.1 Emetteur	55
III.4.1.1 Source d'information	55
III.4.1.2 Codeur I/Q.	55
III.4.1.3 Modulateur des signaux I et Q	57
III.4.2 Le canal de transmission	58
III.4.3 Récepteur	61
III.4.3.1 Récupération de la fréquence porteuse	61
III.4.3.2 Démodulation du signal reçu	62
III.4.3.3 Récupération de la fréquence symbole	63
III.4.3.4 Reconstruction des signaux I et Q.	64
III.4.3.5 Décodeur I/Q	65
III.4.4 Visualisation	66
III.4.5 Calculateur d'erreurs	67
III.4.6 Modèle complet de la chaîne de transmission	67
III.5 Conclusion	69

IV. Mode d'emploi de la simulation

IV.1 Introduction	70
IV.2 L'ouverture du modèle	70
IV.3 Le démarrage de la simulation	73
IV.4 Les visualisations	74
IV.5 Conclusion	80

CONCLUSION GENERALE	82
---------------------	----

ABREVIATIONS

BIBLIOGRAPHIE

Introduction générale

De nos jours, les systèmes de télécommunication numérique avancés sont de plus en plus présents dans notre quotidien et connaissent une évolution sans précédent. Les téléphones cellulaires, les modems haut débit et les réseaux de communication par satellites ne représentent que la pointe d'un iceberg que nous découvrirons dans la décennie de ce nouveau millénaire. La technologie sera prête à temps ou du moins, elle saura s'adapter aux nouvelles tendances du marché.

Bien entendu, ces nouvelles tendances seront liées au monde de l'informatique, de l'information et des communications. En effet, plusieurs groupes scientifiques internationaux travaillent sur les produits du futur qui seront ceux d'un demain très proche. De ces nombreux produits qui sont sur le point d'être mis sur le marché, notons simplement le vidéotéléphone mobile, l'Internet - téléphone mobile, les nano - ordinateurs portables avec liaison RF, le récepteur GPS intégré à ces différents produits, les véhicules intelligents, les centrales inertiels électroniques miniatures, etc. Tous ces futurs produits seront bientôt accessibles sur le marché mais combien d'autres encore restent encore à découvrir ? C'est à la nouvelle génération d'ingénieur que cette question s'adresse !

La plupart de ces dispositifs électroniques nécessitent un système de transmission numérique, c'est à dire un émetteur, un récepteur de signaux numériques et un canal de transmission permettant au module de communiquer avec son entourage.

D'une façon générale, un outil d'analyse basé sur une simulation logicielle réduit le temps de conception et les coûts de développement des projets connexes. Pour plusieurs projets, il s'avère indispensable dans un premier temps, de créer cet outil d'analyse pour ensuite vérifier, par exemple, l'amélioration du comportement du système face aux différents types de brouilleurs présents dans un environnement hostile.

L'objectif général de ce projet consiste à réaliser un système de transmission numérique basé sur la technique de modulation dite QAM 16 (Quadrature Amplitude Modulation avec 16 états).

Simulink et ses blocs sont employés pour démontrer des concepts du système de transmission numérique.

Le premier chapitre aura pour but de présenter les techniques de la modulation numérique les plus utilisées.

Le deuxième chapitre représente l'interface de la simulation (Matlab/Simulink) et le fonctionnement des différents schémas blocs pour modéliser un système.

Le troisième chapitre nous permettra d'aborder plus spécifiquement le problème de construction d'un modèle de la chaîne de transmission bloc par bloc et de choisir les paramètres de chaque bloc.

Le dernier chapitre nous permettra de tester le bon fonctionnement du modèle par la comparaison entre les signaux émis et reçu et proposer une première validation des résultats obtenus par le calculateur d'erreur.

En conclusion, nous allons faire une analyse des résultats obtenus, dégager l'intérêt d'un tel logiciel et faire des propositions pour la poursuite de cette étude.

Chapitre I

Modulation numérique

Références [1], [2], [3], [5]

I.1 Introduction	02
I.2 C'est quoi la modulation numérique et pourquoi ?	03
I.3 Définitions et appellations.	04
I.4 Principe des modulations numériques.	05
I.5 Modulation par déplacement d'amplitude (MDA).	07
I.6 Modulation par déplacement de phase (MDP).	11
I.7 Modulation d'amplitude sur deux porteuses en quadrature (MAQ).	17
I.8 Modulation par déplacement de Fréquence (MDF).	22
I.9 Applications	23
I.10 Conclusion.	24

I Modulation numérique

I.1 Introduction

Les systèmes de transmission numérique véhiculent de l'information entre une source et un destinataire en utilisant un support physique comme le câble, la fibre optique ou encore, la propagation sur un canal radioélectrique. Les signaux transportés peuvent être soit directement d'origine numérique, comme dans les réseaux de données, soit d'origine analogique (parole, image...) mais convertis sous une forme numérique. La tâche du système de transmission est d'acheminer l'information de la source vers le destinataire avec le plus de fiabilité possible.

Le schéma synoptique d'un système de transmission numérique est donné à la Figure I.1 où l'on se limite aux fonctions de base :

- La source émet un message numérique sous la forme d'une suite d'éléments binaires.
- Le codeur peut éventuellement supprimer des éléments binaires non significatifs (compression de données ou *codage de source*).
- La modulation a pour rôle d'adapter le spectre du signal au canal (milieu physique) sur lequel il sera émis.
- Enfin, du côté récepteur, les fonctions de démodulation et de décodage sont les inverses respectifs des fonctions de modulation et de codage situées du côté émetteur.

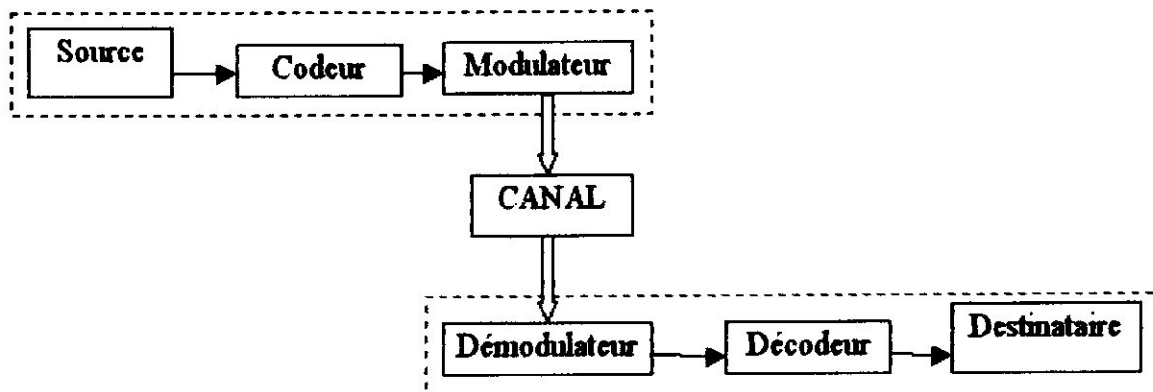


Figure I.1 : Schéma d'un système de transmission numérique

Les trois caractéristiques principales permettant de comparer entre elles les différentes techniques de transmission sont les suivantes:

- La *probabilité d'erreur* P_e par bit transmis permet d'évaluer la qualité d'un système de transmission. Elle est fonction de la technique de transmission utilisée, mais aussi du canal sur lequel le signal est transmis. Il est à noter que P_e est une valeur théorique dont une estimation non biaisée au sens statistique est le Taux d'Erreur par Bit TEB.
- L'*occupation spectrale* du signal émis doit être connue pour utiliser efficacement la bande passante du canal de transmission. On est contraint d'utiliser de plus en plus des modulations à grande efficacité spectrale.

- La complexité du récepteur dont la fonction est de restituer le signal émis est le troisième aspect important d'un système de transmission.

L2 C'est quoi la modulation numérique et pourquoi ?

La modulation a pour objectif

- d'adapter le signal à émettre au canal de transmission.
- La transmission de champs électromagnétique (E.M) dans l'espace est accomplie avec l'utilisation des antennes. Les dimensions de l'antenne dépendent de la longueur d'onde λ du signal émis (en général $\lambda/4$) : $\lambda = \frac{c}{f}$ Avec : c la vitesse de la lumière, et f la fréquence porteuse.

Exemple : si on émet un signal en bande de base de fréquence 3 KHz, on a besoin d'antenne de 25 Km de longueur. Mais si on utilise une porteuse de 900 MHz, une antenne de 8 cm est suffisante.

La modulation consiste à modifier un ou plusieurs paramètres d'une onde porteuse

$$S(t) = A \cos(\omega_0 t + \varphi_0)$$

Les paramètres modifiables sont :

- L'amplitude : A
- La fréquence : $f_0 = \frac{\omega_0}{2\pi}$
- La phase: φ_0

Dans les procédés de modulation binaire, l'information est transmise à l'aide d'un paramètre qui ne prend que deux valeurs possibles. Dans les procédés de modulation M-aire, l'information est transmise à l'aide d'un paramètre qui prend M valeurs. Ceci permet d'associer à un état de modulation un mot de n digits binaires. Le nombre d'états est donc $M = 2^n$. Ces n digits proviennent du découpage en paquets de n digits du train binaire issu du codeur.

Les types de modulation les plus fréquemment rencontrés sont les suivants :

- Modulation par Déplacement d'Amplitude MDA. (*Amplitude Shift Keying ASK*).
- Modulation par Déplacement de Phase MDP. (*Phase Shift Keying PSK*).
- Modulation d'amplitude de deux porteuses en quadrature MAQ. (*Quadrature Amplitude modulation QAM*).
- Modulation par Déplacement de Fréquence MDF. (*Frequency Shift Keying FSK*).

I.3 Définitions et appellations

La transmission de signaux numériques fait appel à quelques notions de base qui sont rappelées ci-après :

- **Le bit** est une valeur binaire (binary digit). Un bit égal à 0 ou 1.
- **Un symbole** est un élément d'un alphabet. Si M est la taille de l'alphabet, le symbole est alors dit M -aire. Lorsque $M=2$, le symbole est dit binaire. En groupant, sous forme d'un bloc, n symboles binaires indépendants, on obtient un alphabet de $M = 2^n$ symboles M -aires. Ainsi un symbole M -aire véhicule l'équivalent de $n = \log_2 M$ bits.
- **La rapidité de modulation R** se définit comme étant le nombre de changements d'états par seconde d'un ou de plusieurs paramètres modifiés simultanément. Un changement de phase du signal porteur, une excursion de fréquence ou une variation d'amplitude sont par définition des changements d'états.

La "rapidité de modulation $R = \frac{1}{T_s}$ s'exprime en "bauds".

Où T_s est la durée d'un symbole en seconde.

- **Le débit binaire D** se définit comme étant le nombre de bits transmis par seconde. Il sera égal ou supérieur à la rapidité de modulation selon qu'un changement d'état représentera un bit ou un groupement de bits.

Le "débit binaire $D = \frac{1}{T_b}$ s'exprime en "bits par seconde".

Où T_b est la durée d'un bit en seconde.

Pour un alphabet M -aire, on a la relation fondamentale : $T_s = nT_b$ soit $D = nR$.

Il y a égalité entre débit de source et rapidité de modulation uniquement dans le cas d'une source binaire (alphabet binaire).

La qualité d'une liaison est liée au *taux d'erreur par bit* :

$$T.E.B = \frac{\text{nombre de bits faux}}{\text{nombre de bits transmis}}$$

- **L'efficacité spectrale** d'une modulation se définit par le paramètre $\eta = \frac{D}{B}$ et s'exprime en "bit/seconde/Hz". La valeur D est le "débit binaire" et B est la largeur de la bande occupée par le signal modulé. Pour un signal utilisant des symboles M -aires, on aura :

$$\eta = \frac{D}{B} = \frac{nR}{B} = \frac{\log_2 M}{B} \cdot \frac{1}{T_s} = \frac{1}{T_s B} \log_2 M \text{ bit/sec/Hz.}$$

l'efficacité spectrale caractérise la capacité d'une modulation à passer un débit maximum dans une largeur de canal minimum.

I.4 Principe des modulations numériques

Le message à transmettre est issu d'une source binaire. Le signal modulant, obtenu après codage, est un signal en bande de base, éventuellement complexe, qui s'écrit sous la forme :

$$c(t) = \sum_k c_k(t) = \sum_k c_k \cdot g(t - kTs)$$

$$\text{avec : } c_k = a_k + j b_k \text{ et } c_k(t) = a_k(t) + j b_k(t).$$

Le canal de transmission étant un milieu continu, avant de pouvoir y transmettre les symboles c_k il faut obtenir un signal continu .

La fonction $g(t)$ est une forme d'onde qui est prise en considération dans l'intervalle $[0, Ts]$.

Une forme d'onde classique est tout simplement le rectangle de durée Ts .

Dans les modulations MDA, MDP et MAQ, la modulation transforme ce signal $c(t)$ en un signal modulé $m(t)$ tel que :

$$m(t) = \text{Re} \left[\sum_k c_k(t) \cdot e^{j(\omega_0 t + \varphi_0)} \right]$$

La fréquence $f_0 = \frac{\omega_0}{2\pi}$ et la phase φ_0 caractérisent la sinusoïde porteuse utilisée pour la modulation.

Si les $c_k(t) = a_k(t) + j b_k(t)$ sont réels ($b_k(t) = 0$), la modulation est dite *unidimensionnelle*, et s'ils sont complexes la modulation est dite *bidimensionnelle*.

Le signal modulé s'écrit aussi plus simplement :

$$m(t) = \sum_k a_k(t) \cdot \cos(\omega_0 t + \varphi_0) - \sum_k b_k(t) \cdot \sin(\omega_0 t + \varphi_0)$$

$$\text{en posant : } a(t) = \sum_k a_k(t) \text{ et } b(t) = \sum_k b_k(t)$$

$$\text{Ou encore : } m(t) = a(t) \cdot \cos(\omega_0 t + \varphi_0) - b(t) \cdot \sin(\omega_0 t + \varphi_0)$$

Le signal $a(t)$ module en amplitude la porteuse en phase $\cos(\omega_0 t + \varphi_0)$ et le signal $b(t)$ module en amplitude la porteuse en quadrature $\sin(\omega_0 t + \varphi_0)$.

Les deux signaux $a(t)$ et $b(t)$ sont aussi appelés "trains modulants".

Le schéma théorique du modulateur est représenté sur la Figure I.2.

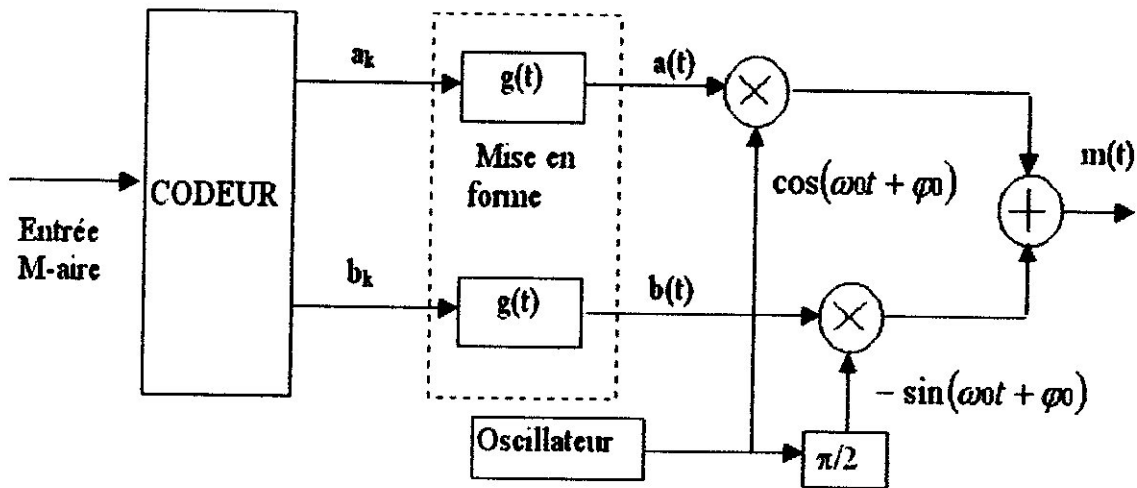


Figure I.2 : Forme générale du modulateur

A chaque symbole émis correspond un signal élémentaire de la forme :

$$m_k(t) = a_k \cdot g(t - kT_s) \cdot \cos(\omega_0 t + \varphi_0) - b_k \cdot g(t - kT_s) \cdot \sin(\omega_0 t + \varphi_0)$$

Le signal modulé $m(t)$ véhicule des informations distinctes à travers a_k et b_k qui sont deux signaux en bande de base appelés respectivement *composante en phase* (I en anglais) et *composante en quadrature* (Q en anglais).

Une représentation dans le plan complexe qui fait correspondre à chaque signal élémentaire un point $C_k = A_k + jB_k$ permet de différencier chaque type de modulation. L'ensemble de ces points associés aux symboles porte le nom de **constellation** (figure I.3).

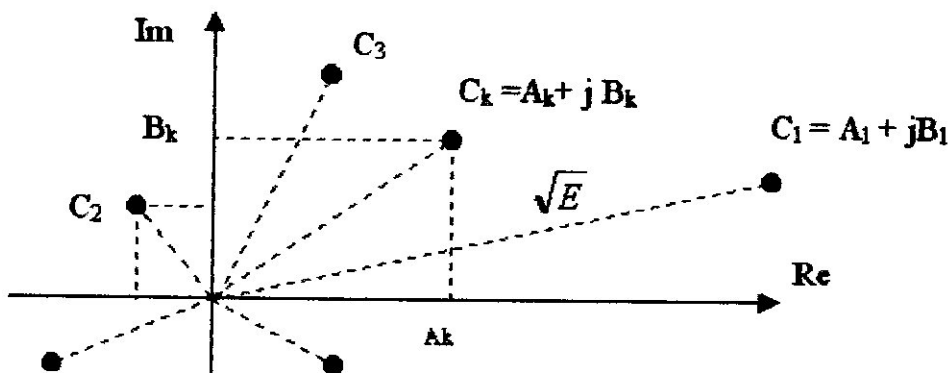


Figure I.3 : Définition d'une constellation numérique

Le choix de la répartition des points dépend des critères suivants :

- Pour pouvoir distinguer deux symboles, il faut respecter une distance minimale d_{\min} , entre les points représentatifs de ces symboles. Plus cette distance est grande et plus la probabilité d'erreur sera faible. La distance minimale entre tous les symboles est :

$$d_{\min} = \underset{i \neq j}{\text{Min}}(d_{ij}) \text{ avec } d_{ij} = |C_i - C_j| \text{ où } C_i \text{ et } C_j \text{ les distances symbole -origine}$$

- A chaque symbole émis correspond un signal élémentaires $m_k(t)$ et par là même une énergie nécessaire à la transmission de ce symbole. Dans la constellation, la distance entre un point et l'origine est proportionnelle à la racine carrée de l'énergie qu'il faut fournir pendant l'intervalle de temps $[kT_s, (k+1)T_s]$ pour émettre ce symbole. La puissance moyenne d'émission des symboles est assimilable à $\sum_i |C_i|^2$ et la puissance crête à $\text{Max}_i |C_i|^2$

Les deux critères évoqués ci-dessus sont antagonistes puisque l'on serait tenté d'une part d'éloigner les symboles au maximum pour diminuer la probabilité d'erreur et d'autre part, de les rapprocher de l'origine pour minimiser l'énergie nécessaire à la transmission.

Les critères de choix d'une modulation sont :

- La constellation qui suivant les applications mettra en évidence une faible énergie nécessaire à la transmission des symboles ou une faible probabilité d'erreur.
- L'occupation spectrale du signal modulé.
- La simplicité de réalisation (avec éventuellement une symétrie entre les points de la constellation).

1.5 Modulation par Déplacement d'amplitude (MDA)

Définition : la modulation MDA (appelée aussi ASK) est une modulation dans laquelle l'amplitude de la porteuse varie au rythme des symboles transmis.

Dans ce cas, la modulation ne s'effectue que sur la porteuse en phase $\cos(\omega_0 t + \varphi_0)$. Il n'y a pas de porteuse en quadrature. Cette modulation est parfois dite mono dimensionnelle. Le signal modulé dans une période T_s s'écrit alors:

$$m(t) = a_k \cdot g(t - kT_s) \cdot \cos(\omega_0 t + \varphi_0)$$

La forme de l'onde $g(t)$ est rectangulaire, de durée T_s et d'amplitude égale à 1 si t appartient à l'intervalle $[0, T_s]$ et égale à 0 ailleurs.

I.5.1 Modulation par tout ou rien

Un exemple de modulation d'amplitude est la modulation (binaire) par tout ou rien encore appelée par son abréviation anglaise : OOK pour "On Off Keying", le principe de cette modulation est montré à la figure I.4.

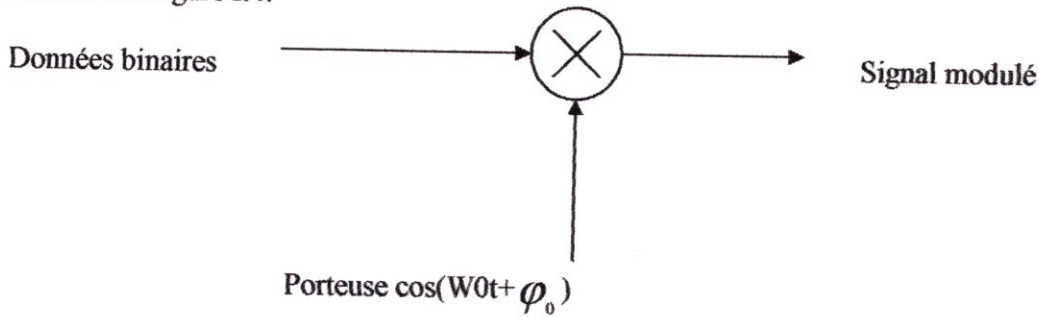


Figure I.4 : principe de la modulation tout ou rien

Dans ce cas, un seul bit est transmis par période T_s , et par conséquent $n=1$ et $M=2$. Le symbole a_k prend sa valeur dans l'alphabet $(0, a_0)$ comme montre la figure I.5.



Figure I.5: Constellation de la modulation d'amplitude par tout ou rien

On observe donc sur un chronogramme des extinctions de porteuse quand $a_k=0$ (Figure I.6).

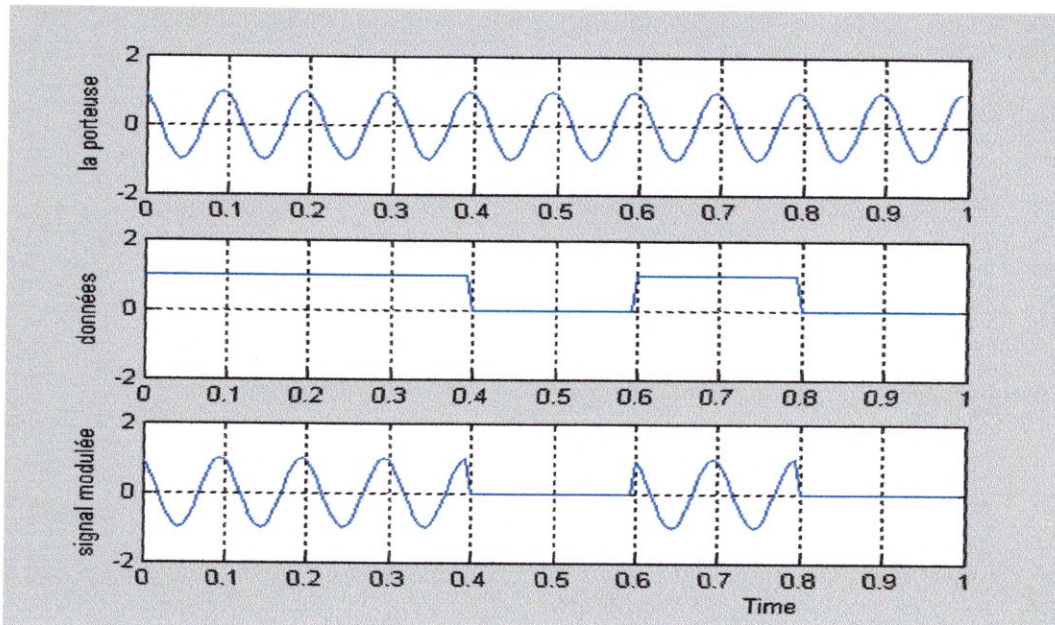


Figure I.6: Chronogramme de la modulation d'amplitude par tout ou rien

I.5.2 Modulation à "M ETATS"

On a toujours $M = 2^n$ amplitudes possibles du signal, mais ici les valeurs de l'alphabet sont telles que :

$$A_i = (2i - (M + 1)) \cdot a_0 \text{ avec } i = 1, 2 \dots M.$$

Suivant les valeurs de n on obtient le tableau suivant :

n	M	Valeurs de l'alphabet
1	2	$-1a_0, 1a_0$
2	4	$-3a_0, -1a_0, 1a_0, 3a_0$
3	8	$-7a_0, -5a_0, -3a_0, -1a_0, 1a_0, 3a_0, 5a_0, 7a_0$

Tableau I.1 : les valeurs possibles de modulation selon le nombre de bits

La constellation de la modulation à M états symétriques est donnée par la Figure I.7 pour M prenant comme valeurs 2, 4 et 8.

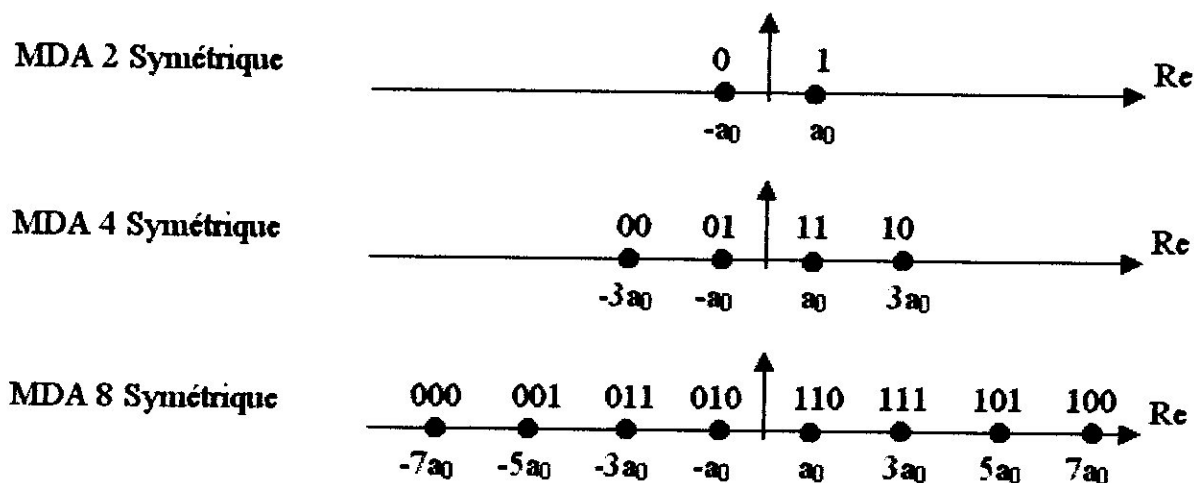


Figure I.7: Constellation de la modulation d'amplitude à M états

On remarquera que la disposition des symboles met en œuvre un code de Gray de telle sorte qu'un seul bit change lorsque l'on passe d'un point à un autre.

Le principe de la modulation d'amplitude à M états est montré à la figure I.8.

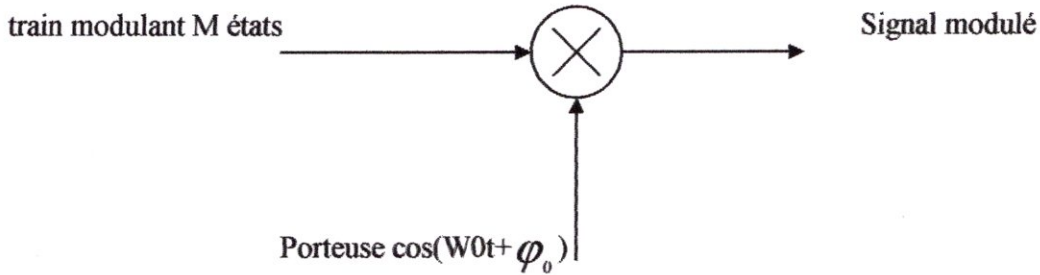


Figure I.8 : principe de la modulation d'amplitude à M états

Le chronogramme de cette modulation est donné à la figure I.9. On remarque que à chaque changement du signe il y a un déphasage de π .

Dans cet exemple, deux bits sont transmis simultanément dans chaque symbole.

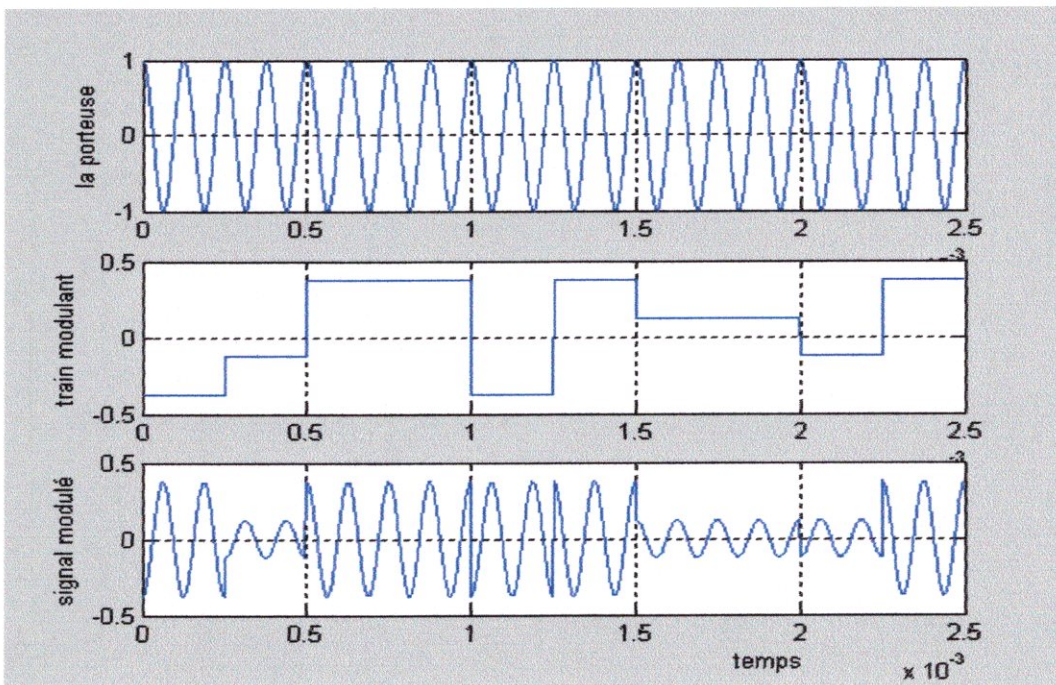


Figure I.9: Chronogramme de la modulation d'amplitude à M états

❖ La démodulation :

Le signal reçu dans une période T_s est $m(t) = a_k \cdot g(t - kT_s) \cdot \cos(\omega_0 t + \varphi_0)$

On multiplie ce signal par une sinusoïde sortant d'un oscillateur local $A_1 \cdot \cos(\omega_0 t + \varphi)$ on obtient

$$S_1(t) = m(t) = a_k \cdot g(t - kT_s) \cdot \cos(\omega_0 t + \varphi_0) \cdot A_1 \cdot \cos(\omega_0 t + \varphi)$$

En développant cette expression et en éliminant le terme en $\cos(2\omega_0 t)$ par filtrage, on obtient :

$$S_2(t) = \frac{A_1}{2} \sum_k (a_k \cdot g(t - kT_s) \cdot \cos(\varphi_0 - \varphi_k))$$

Donc, si le récepteur dispose d'un oscillateur local synchronisé en fréquence et en phase sur celui de l'émission, φ_0 sera proche de φ et, donc $\cos(\varphi_0 - \varphi)$ sera voisin de 1, et par conséquent $s_2(t) \approx \frac{A_1}{2} \sum_k a_k \cdot g(t - kT)$.

Ainsi, le signal $s_2(t)$ est à une homothétie près égal au train modulant $a(t) = \sum_k a_k \cdot g(t - kT)$ qui est lui même le signal porteur de l'information. Il reste encore au récepteur à récupérer le rythme, de période T_s , des symboles transmis, et échantillonner le signal $s_2(t)$ à une fréquence $1/T_s$.

Dans les procédures précédentes on utilise la relation suivante :

$$\cos x \cdot \cos y = \frac{1}{2} [\cos(x+y) + \cos(x-y)]$$

Le principe de démodulation est montré à la figure I.10.

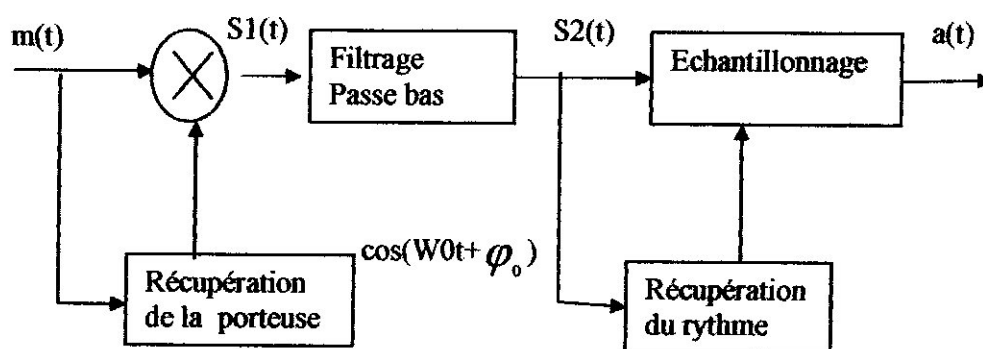


Figure I.10 : principe de démodulation d'amplitude à M états

I.6 Modulation par Déplacement de phase (MDP)

Définition : la modulation MDP (appelée aussi PSK) est une modulation dans laquelle la phase de la porteuse varie au rythme des symboles transmis.

Dans ce type de modulation on utilise la notion d'enveloppe complexe, ou les symboles c_k s'écrit :

$c_k = a_k + j b_k$, Dans ce cas, les symboles c_k sont répartis sur un cercle de rayon A , et par conséquent : $c_k = a_k + j b_k = \sqrt{a_k^2 + b_k^2} \cdot e^{j\varphi_k}$ Mais $\sqrt{a_k^2 + b_k^2} = A$ donc :

$$a_k = A \cos(\varphi_k) \quad \text{et} \quad b_k = A \sin(\varphi_k)$$

L'ensemble des phases possibles se traduit par les expressions suivantes :

$$\varphi_k = \frac{\pi}{M} + k \frac{2\pi}{M} \text{ Lorsque } M > 2; \quad \text{Et : } \varphi_k = 0 \text{ ou } \pi \text{ lorsque } M = 2.$$

Si on prend l'exemple où $M=4$ et $n=2$ (MDP-4), Le tableau I.2 précise les différentes valeurs en fonction du symbole à transmettre.

Bit pair	Bit impair	Symbole	φ_k	a_k	b_k
0	0	00	$\frac{\pi}{4}$	$A \cdot \sqrt{2}/2$	$A \cdot \sqrt{2}/2$
1	0	01	$\frac{3\pi}{4}$	$-A \cdot \sqrt{2}/2$	$A \cdot \sqrt{2}/2$
1	1	11	$\frac{5\pi}{4}$	$-A \cdot \sqrt{2}/2$	$-A \cdot \sqrt{2}/2$
0	1	10	$\frac{7\pi}{4}$	$A \cdot \sqrt{2}/2$	$-A \cdot \sqrt{2}/2$

Tableau I.2 : les valeurs de MDP-4 en fonction des symboles

On appelle "MDP-M" une modulation par déplacement de phase (MDP) correspondant à des symboles M-aires. Les Figures I.11 et I.12 montrent les différentes constellations de MDP pour $M=2, 4$ et 8 .

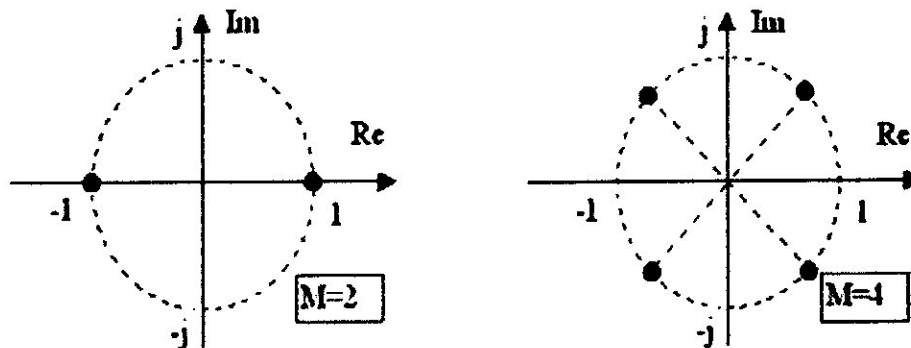


Figure I.11: Constellation des symboles en modulation de phase MDP-2,4

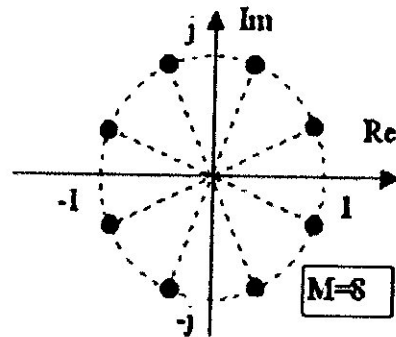


Figure I.12 : Constellation des symboles en modulation de phase MDP-8

❖ La modulation

Dans un période T_s le signal modulé a la forme générale suivante:

$$m_k(t) = a_k \cdot g(t - kT_s) \cdot \cos(\omega_0 t + \varphi_0) - b_k \cdot g(t - kT_s) \cdot \sin(\omega_0 t + \varphi_0)$$

et plus simplement : $= A \cos(\omega_0 t + \varphi_0) \cos(\varphi_k) \cdot g(t - kT_s) - A \sin(\omega_0 t + \varphi_0) \sin(\varphi_k) \cdot g(t - kT_s)$

Cette dernière expression montre que la phase de la porteuse est modulée par l'argument φ_k de chaque symbole ce qui explique le nom donné à la MDP. Remarquons aussi que la porteuse en phase $\cos(\omega_0 t + \varphi_0)$ est modulée en amplitude par le signal $A \cos(\varphi_k)$ et que la porteuse en quadrature $\sin(\omega_0 t + \varphi_0)$ est modulée en amplitude par le signal $A \sin(\varphi_k)$.

La figure I.13 représente le principe de cette modulation :

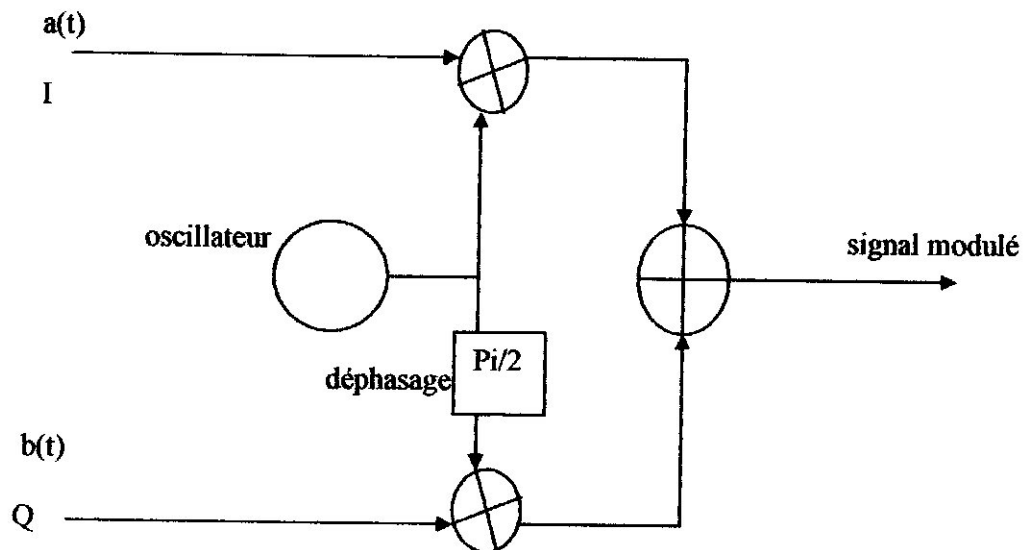


Figure I.13 : principe de la modulation de phase

On rappelle que $a(t) = \sum_k a_k g(t - kT)$ et $b(t) = \sum_k b_k g(t - kT)$

Les deux trains modulants $a(t)$ et $b(t)$ sont deux trains binaire codés en standard NRZ (non retour à zéro), on peut notée aussi I (en phase) et Q (en quadrature).

Remarque :

Le codage NRZ est un technique de codage qui code un bit 1 par 1, et un bit 0 par -1.

Les figures I.14, I.15, et I.16 illustrent les chronogrammes des canaux I et Q et le signal modulé en modulation MDP-4.

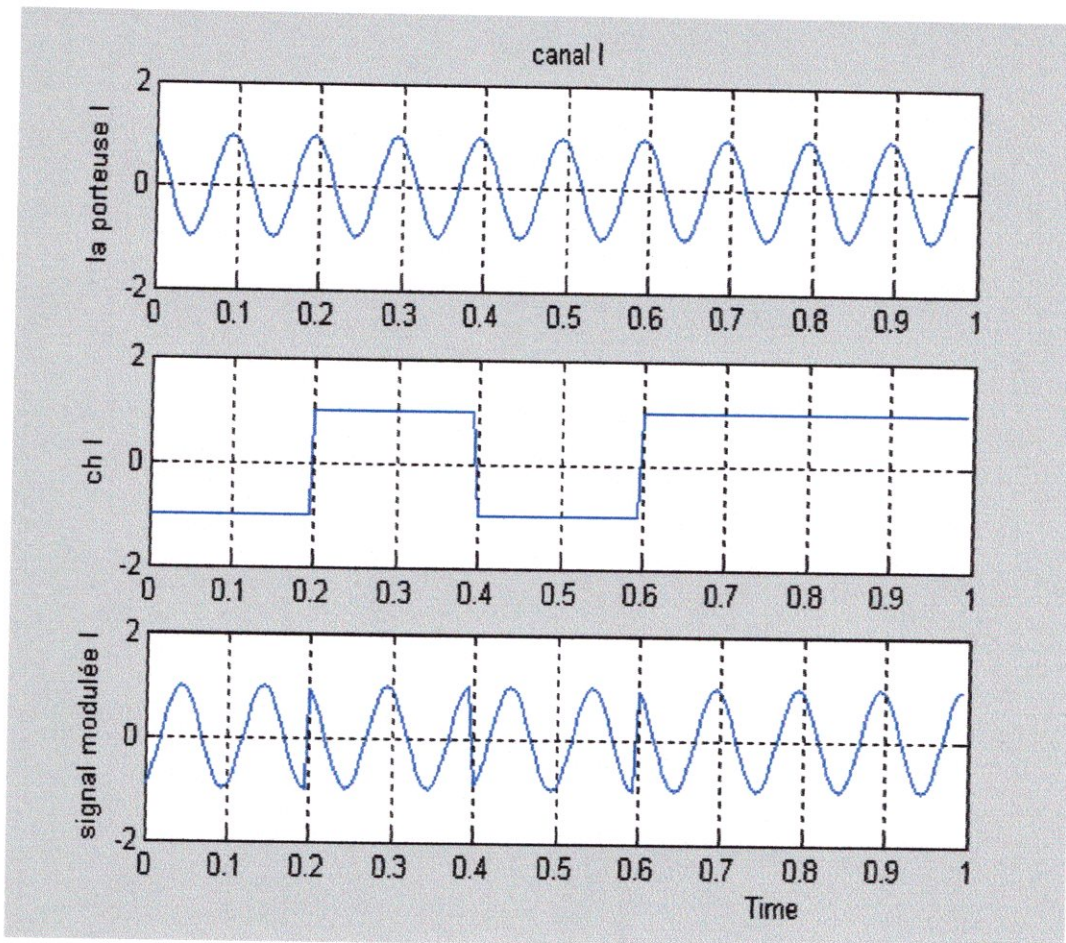


Figure I.14 : chronogramme de canal I en modulation MDP-4

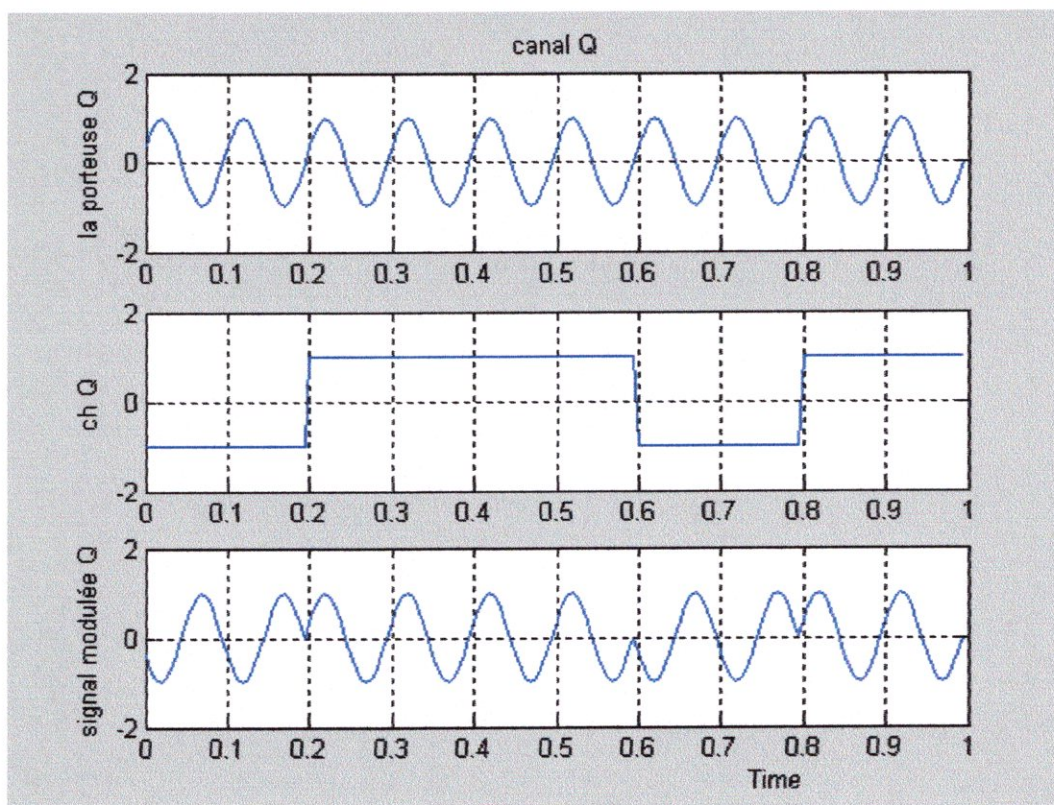


Figure I.15 : chronogramme de canal Q en modulation MDP-4

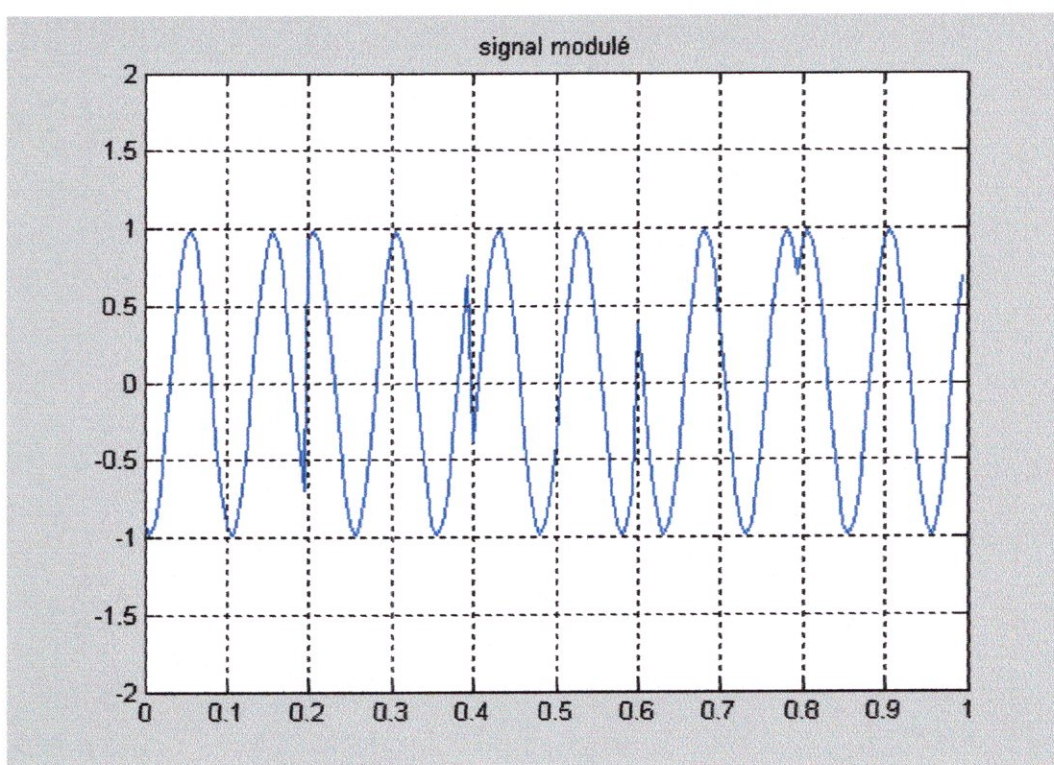


Figure I.16 : signal modulé en MDP-4

❖ La démodulation

Le signal reçu est démodulé dans deux voies parallèles par deux porteuses en quadrature. Certaines techniques permettent de synchroniser l'oscillateur local avec la porteuse à l'émission. Le signal en quadrature est généré à partir de l'oscillateur local et d'un déphaseur de $\pi / 2$.

Soit le signal reçu dans un période T_s :

$$r(t) = a_k \cdot g(t - kT_s) \cdot \cos(\omega_0 t + \varphi_0) - b_k \cdot g(t - kT_s) \cdot \sin(\omega_0 t + \varphi_0)$$

Pour la voie I et après multiplication avec la porteuse récupérée, on obtient :

$$S a_1(t) = (a_k \cdot g(t - kT_s) \cdot \cos(\omega_0 t + \varphi_0) - b_k \cdot g(t - kT_s) \cdot \sin(\omega_0 t + \varphi_0)) \cdot \cos(\omega_0 t + \varphi_0)$$

Donc, après filtrage pour éliminer la composante à la fréquence $2f_0$: $S a_2(t) = \frac{a_k}{2} \cdot g(t - kT_s)$

De la même manière on obtient pour la voie Q: $S b_2(t) = \frac{b_k}{2} \cdot g(t - kT_s)$

Le récepteur doit encore récupérer le rythme des symboles transmis, puis échantillonner les signaux $S a_2(t)$ et $S b_2(t)$ pour obtenir $a(t)$ et $b(t)$.

Dans les procédures précédentes on utilise les relations suivantes :

$$\cos x \cdot \cos y = 1/2[\cos(x+y) + \cos(x-y)]$$

$$\sin x \cdot \cos y = 1/2[\sin(x+y) + \sin(x-y)]$$

Le schéma synoptique d'un démodulateur MDP-4 est présenté à la figure I.17.

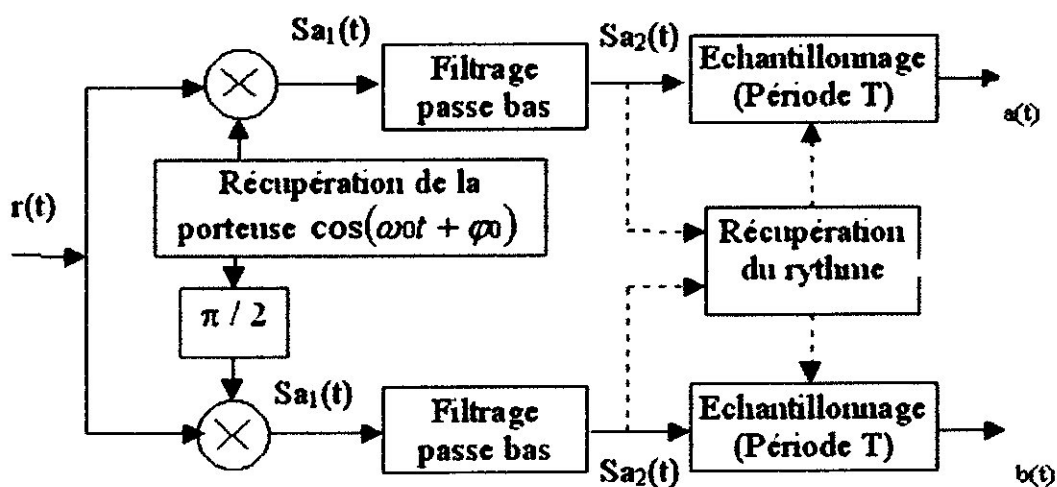


Figure I.17 : principe de démodulation MDP-4

I.7 Modulation d'amplitude sur deux porteuses en quadrature (MAQ)

Définition : la modulation MAQ (appelée aussi QAM) est une modulation dans laquelle la phase et l'amplitude de la porteuse varient au rythme des symboles transmis.

Dans ce type de modulation on utilise aussi la notion d'enveloppe complexe, et alors on a deux canaux I et Q, donc on a $c_k = a_k + j b_k$.

Les symboles a_k et b_k prennent leurs valeurs dans l'alphabet

$$A_i = (2i - (M + 1)) \cdot a_0 \text{ avec } i = 1, 2, \dots, M$$

où $M = 2^n$ est le nombre d'états et n le nombre de bits

La différence entre MDP et MAQ est la distribution des points sur le diagramme de constellation, où dans MDP tous les points sont situés sur un cercle, alors ils ont la même amplitude. Mais dans MAQ l'amplitude dépend de la distance entre le point et le centre du diagramme de constellation.

Par exemple, la MAQ-16 est construite à partir de deux modulations en MDA à quatre états. Les symboles a_k et b_k prennent leurs valeurs dans l'alphabet $\{\pm a_0, \pm 3a_0\}$ où a_0 est une constante donnée. On a deux bits donc quatre états dans chaque canal, et quatre bits transmis donc 16 états différents. Une représentation de la constellation de cette modulation est donnée Figure I.18.

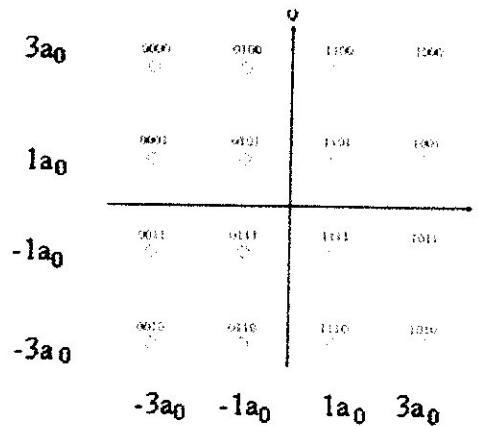
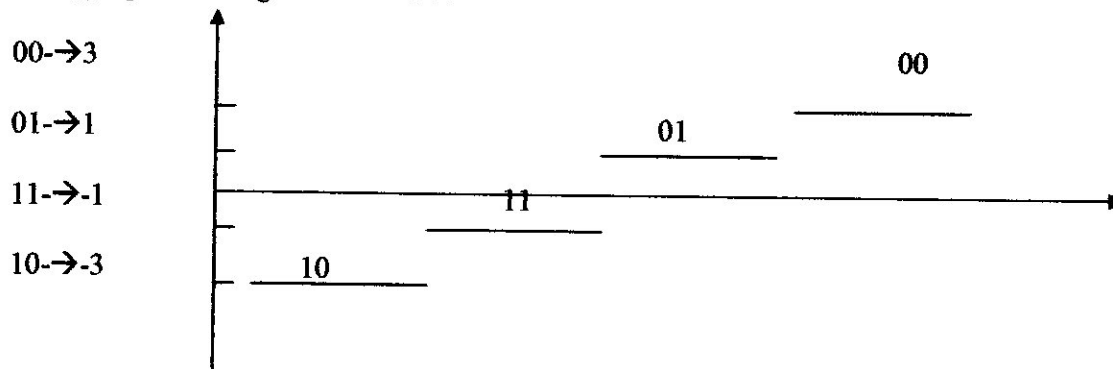


Figure I.18 : Constellation des symboles en modulation MAQ -16

Le tableau I.3 précise les différentes valeurs en fonction du symbole à transmettre.

On applique le codage comme suit :



Couple de bits impair	Couple de bits pair	Symbole	φ_k	a_k	b_k
00	00	00 00	$\pi/4$	3	3
00	01	00 01	$\pi/10$	3	1
01	00	01 00	$4\pi/10$	1	3
01	01	01 01	$\pi/4$	1	1
11	01	11 01	$3\pi/4$	-1	1
11	00	11 00	$6\pi/10$	-1	3
10	01	10 01	$9\pi/10$	-3	1
10	00	10 00	$3\pi/4$	-3	3
11	11	11 11	$5\pi/4$	-1	-1
11	10	11 10	$14\pi/10$	-1	-3
10	11	10 11	$11\pi/10$	-3	-1
10	10	10 10	$5\pi/4$	-3	-3
01	11	01 11	$7\pi/4$	1	-1
01	10	01 10	$16\pi/10$	1	-3
00	11	00 11	$19\pi/10$	3	-1
00	10	00 10	$7\pi/4$	3	-3

Tableau I.3 : les valeurs de MAQ-16 en fonction des symboles

❖ La modulation

Lorsque le signal $m(t)$ est obtenu par une combinaison de deux porteuses en quadrature modulées en amplitude par des symboles a_k et b_k indépendants, cela simplifie le modulateur et le démodulateur.

Dans un période T_s le signal modulé a la forme générale suivante:

$$m_k(t) = a_k \cdot g(t - kT_s) \cdot \cos(\omega_0 t + \varphi_0) - b_k \cdot g(t - kT_s) \cdot \sin(\omega_0 t + \varphi_0)$$

On rappelle que $a(t) = \sum_k a_k \cdot g(t - kT)$ et $b(t) = \sum_k b_k \cdot g(t - kT)$

Les données peuvent être

- ❖ Binaire : on divise les données en deux trains $\{a_k\}$ et $\{b_k\}$ puis on utilise des convertisseurs pour qui donnent pour chaque groupe de bits une valeur dans l'alphabet $\{\pm a_0, \pm 3 a_0\}$.
- ❖ entier de 0 jusqu'à $M-1$: dans ce cas on divise le train de données en deux trains $\{a_k\}$ et $\{b_k\}$, chaque symbole de ces trains a une valeur dans l'alphabet $\{\pm a_0, \pm 3 a_0\}$.

Le modulateur est similaire au modulateur MDP comme montre la figure I.19 :

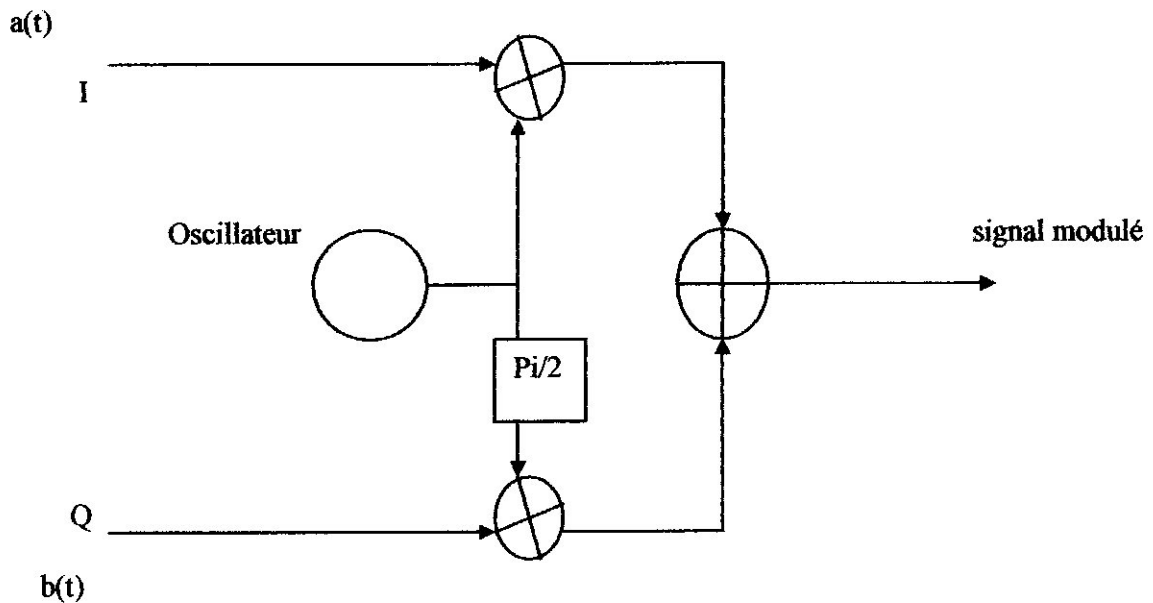


Figure I.19 : principe de la modulation MAQ

Les chronogrammes des figures I.20, I.21, et I.22 donnent une idée plus simple de ce modulateur :

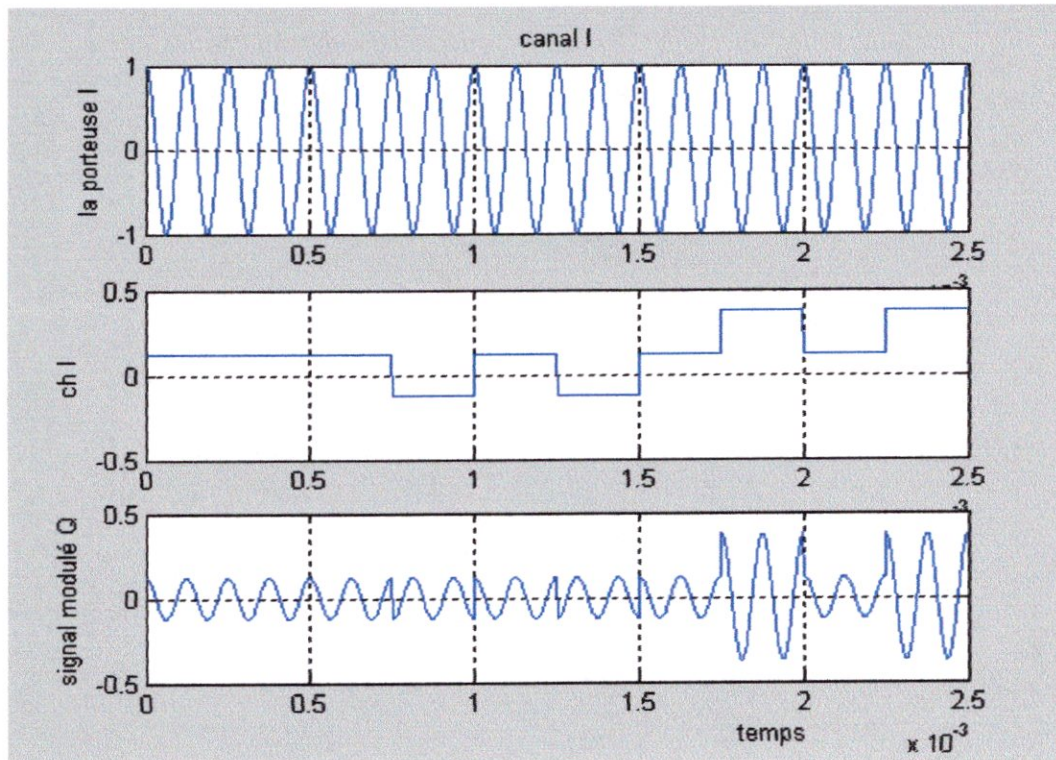


Figure I.20 : chronogramme de canal I en modulation MAQ-16

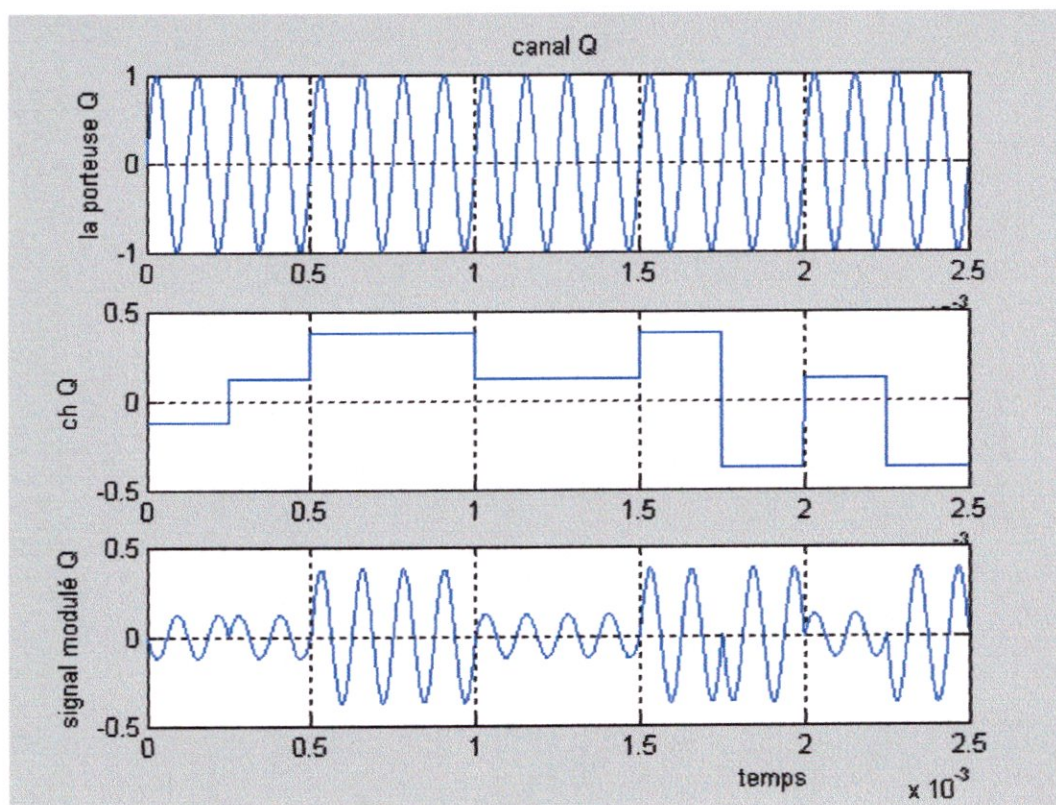


Figure I.21 : chronogramme de canal Q en modulation MAQ-16

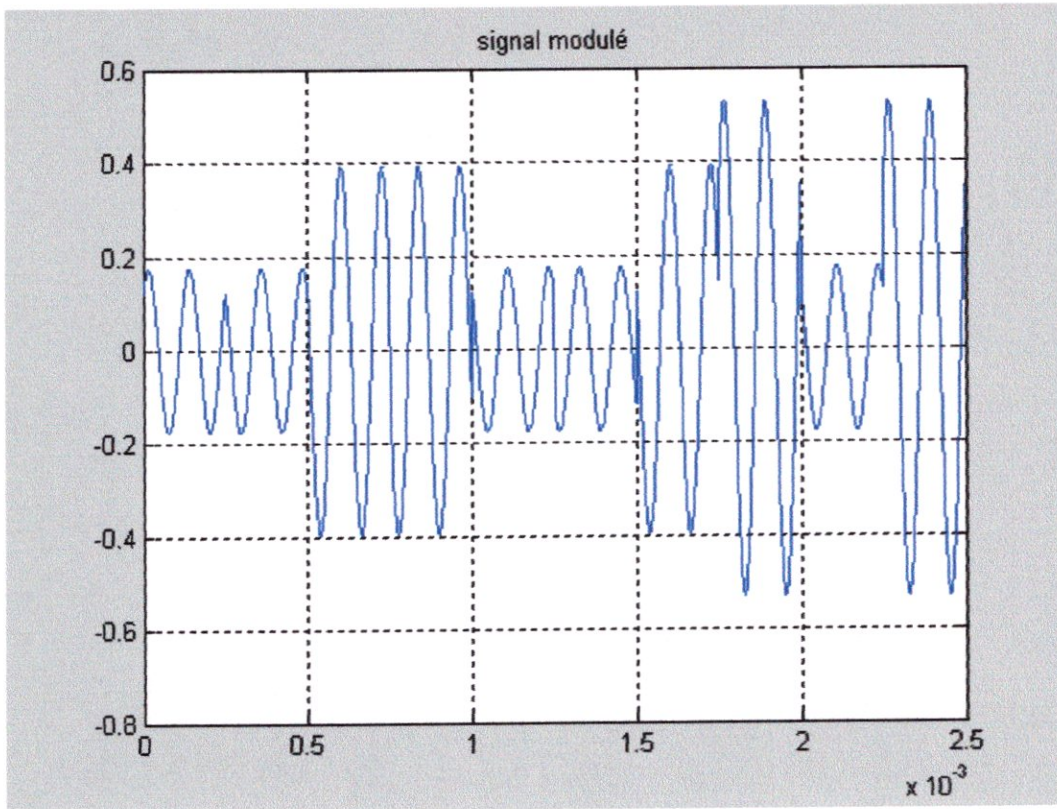


Figure I.22 : signal modulé en modulation MAQ-16

On remarque que cette modulation est un MDA à M états dans chaque canal, et la somme des deux canaux donne la modulation MAQ.

❖ La démodulation :

La réception d'un signal MAQ nécessite l'extraction d'une porteuse synchronisée en phase et en fréquence avec la porteuse à l'émission. Le signal reçu est démodulé dans deux branches parallèles, sur l'une avec la porteuse en phase et sur l'autre avec la porteuse en quadrature. Les équations de démodulation sont les mêmes que celles de la modulation MDP.

$$S a_1(t) = (a_k \cdot g(t - kTs) \cdot \cos(\omega_0 t + \varphi_0) - b_k \cdot g(t - kTs) \cdot \sin(\omega_0 t + \varphi_0)) \cdot \cos(\omega_0 t + \varphi_0)$$

Ensuite, un filtrage est utilisé pour éliminer la composante à la fréquence $2f_0$: $S a_2(t) = \frac{a_k}{2} \cdot g(t - kTs)$

De la même manière on obtient pour la voie Q: $S b_2(t) = \frac{b_k}{2} \cdot g(t - kTs)$

Ce principe est montré à la figure I.23.

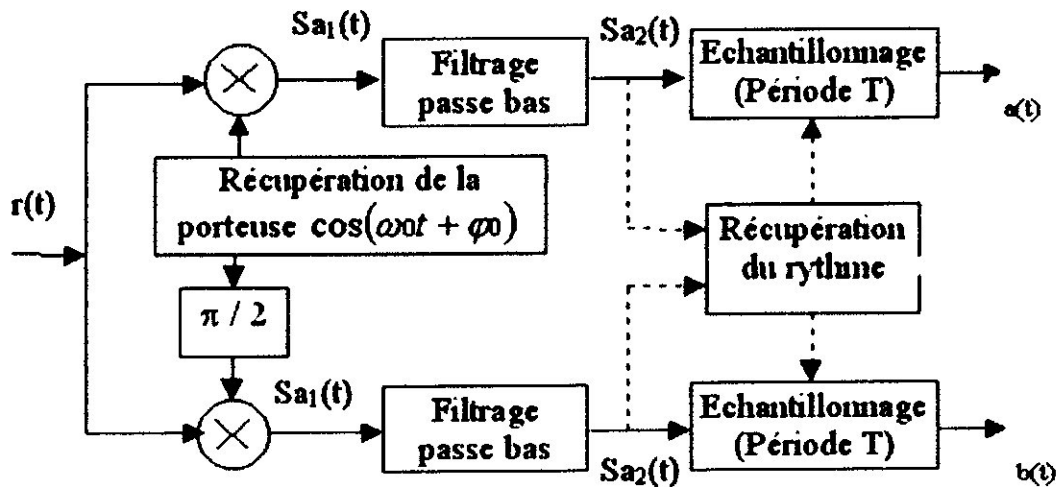


Figure I.23 : principe de démodulation MAQ-16

I.8 Modulation par Déplacement de Fréquence (MDF)

Définition : c'est la modulation (appelée aussi FSK) dans laquelle on associe à chaque symbole transmis a_k un signal de fréquence f_i .

Le signal modulé s'écrit alors : $m_k(t) = A \cos(2\pi (f_0 + \frac{\Delta f}{2} a_k) t)$

Où Δf est la différence de la fréquence instantanée correspondant à l'émission de deux symboles adjacents.

Si les symboles sont de type binaire c.à.d chaque symbole est un « 1 » ou « 0 », le signal modulé sera une sinusoïde de fréquence f_1 pour les niveau « 0 » et f_2 pour les signaux de niveau « 1 ».

La modulation MDF le plus simple est montrée à la figure I.24.

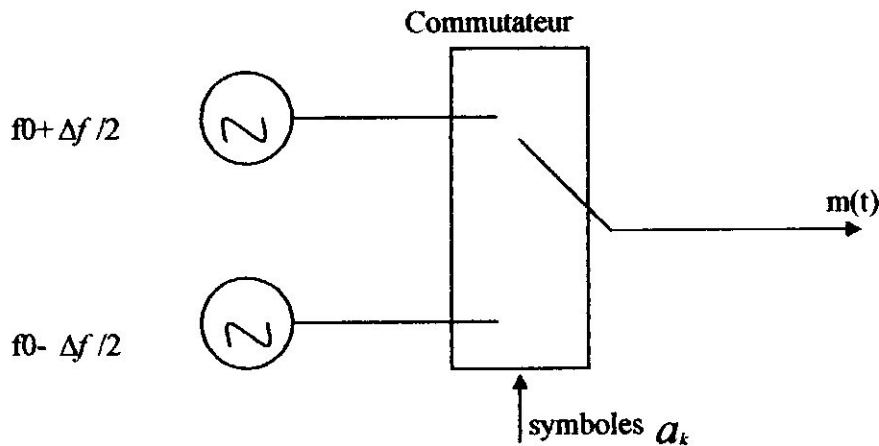


Figure I.24 : principe de la modulation MDF

La figure I.25 montre deux fréquences différentes sur la même forme d'onde correspondant à deux symboles différents « 0 » ou « 1 ».

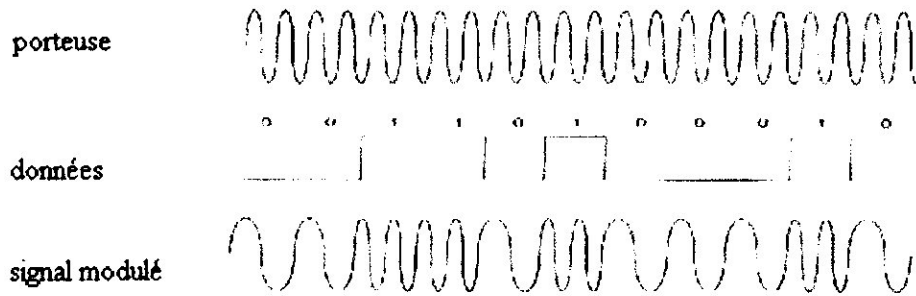


Figure I.25 : la forme d'onde de la modulation MDF

❖ La démodulation

Pour démoduler, on utilisera une P.L.L (phased locked loop) qui s'accrochera sur f_1 lors de la réception d'un « 0 » et sur f_2 lors de la réception d'un « 1 ». Il suffira alors de récupérer le signal en entrée du V.C.O (voltage controlled oscillator) de la boucle qui prendra la forme du signal numérique.

Il ne reste plus qu'à remettre le signal en forme pour obtenir des niveaux de tensions voulus qui correspondent à « 0 » ou « 1 ». Pour cela, on va supprimer la valeur moyenne du signal récupéré et envoyer ce dernier sur un comparateur.

Remarque :

On devra calculer le filtre de la boucle pour que cette dernière puisse s'accrocher autour de f_1 et f_2 . Le principe de la modulation MDF est montré à la figure I.26.

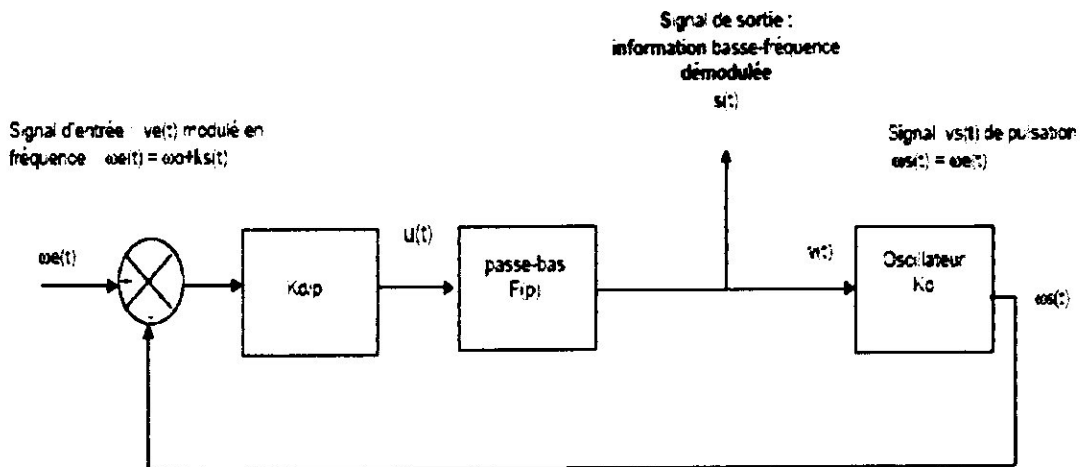


Figure I.26 : principe de la démodulation MDF

1.9 Applications

Les domaines d'applications des diverses techniques de transmission numérique que nous venons d'exposer sont très variés. Quelques-uns sont décrits dans le tableau I.4 :

Type de modulation	applications
MDP-4	Satellite, DVB (digital video broadcasting)-satellite, modem du câble.
MDF	Faisceaux hertziens.
MDP-8	Satellite.
MAQ-16	Radio numérique, modem, DVB- câble, DVB-terrestre, liaison numérique hyperfréquence.
MAQ-32	Micro-onde terrestre, DVB-T.
MAQ-64	DVB-c, modem.
MAQ-256	Modem, DVB-c, vidéo numérique (USA)

Tableau I.4 : quelques applications de la modulation numérique

I.10 Conclusion

L'extraordinaire variété des applications que nous venons d'exposer met en évidence l'importance capitale des différentes techniques de transmission numérique sur porteuse. Les systèmes modernes de communication numérique sont complexes et requièrent des circuits de modulation et de démodulation de plus en plus sophistiqués. Nous avons examiné un certain nombre de modulations qui sont aujourd'hui utilisées. Il s'avère que le choix d'un type de modulation est toujours déterminé par les contraintes de l'application.

Chapitre II

Présentation du logiciel MATLAB/SIMULINK

Références [6], [7]

II.1 Introduction.	25
II.2 La structure du SIMULINK	25
II.3 Description des blocs de Simulink.	29
II. 4 Création des schémas blocs.	49
II. 5 Ajustement des paramètres de la simulation.	50
II.6 conclusion.	51

II Présentation du logiciel MATLAB/SIMULINK

II.1 Introduction

Ce chapitre a pour objectif premier de présenter un support à l'utilisation des outils Matlab/simulink.

L'interpréteur MATLAB (MATrix LABoraty) de la compagnie Mathworks est un programme interactif pour le calcul scientifique qui facilite le calcul matriciel dans différents domaines scientifiques comme le traitement de signal, l'électronique, les commandes de processus et qui permet une visualisation rapide de données.

L'ajout de "boîtes à outils" (Toolboxes) permet d'introduire de nouvelles fonctions mathématiques spécifiques à un domaine d'applications particulier (Signal Processing Toolbox, Control System Toolbox, Neural Network Toolbox, Signal Processing Toolbox, Image Processing Toolbox, etc.) afin de tester rapidement les performances de nouvelles méthodes. Dans cet optique, MATLAB contient de nombreuses fonctionnalités pour :

- l'analyse de données et la visualisation,
- le calcul numérique et symbolique,
- la modélisation, la simulation,
- le développement d'applications et d'interface graphiques.

SIMULINK est un outil interactif graphique de MATLAB permettant de modéliser, simuler et analyser une large variété de systèmes physiques et mathématiques, y compris ceux avec des éléments non - linéaires et ceux qui se servent du temps continu et discret ou hybride. Il est communément utilisé dans la conception de systèmes automatiques, la conception de systèmes à base de processeurs de traitement du signal et les systèmes de communication. Pour cela, il dispose d'une bibliothèque de blocs de calcul élémentaire (figure II.2) que l'utilisateur peut assembler pour créer ses systèmes sans écrire des lignes de codes.

Dans SIMULINK, la simulation se fait en deux étapes. La première consiste à créer le modèle graphique du système à simuler en utilisant l'éditeur de modèle de SIMULINK. Ensuite, en spécifiant l'étendu temporel, il est possible de simuler le comportement du système.

II.2 La structure du SIMULINK

Simulink étant un outil de Matlab, pour le démarrer on doit d'abord démarrer Matlab en allant dans le menu démarrer. La fenêtre de commande de Matlab est montrée à la figure II.1.

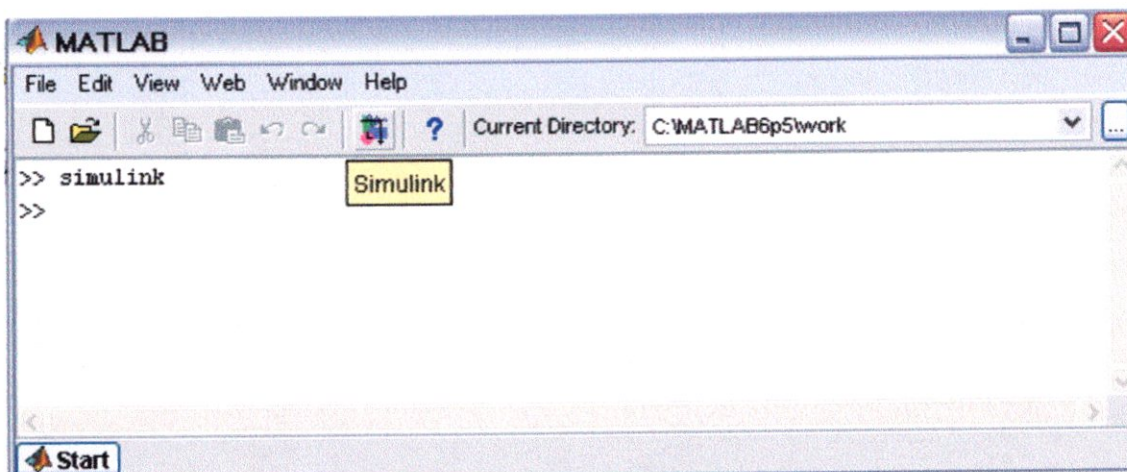



Figure II.1 : la fenêtre de commande du Matlab

Lorsque la fenêtre principale de Matlab est ouverte, cliquer sur l'icône  Simulink dans la barre d'outils de Matlab ou taper simulink dans la fenêtre de commande puis valider.

Dès le lancement, Simulink propose dans une fenêtre la bibliothèque des structures (figure II.2) prédéfinies usuellement rencontrées, ainsi que les objets graphiques définis dans les Toolbox.

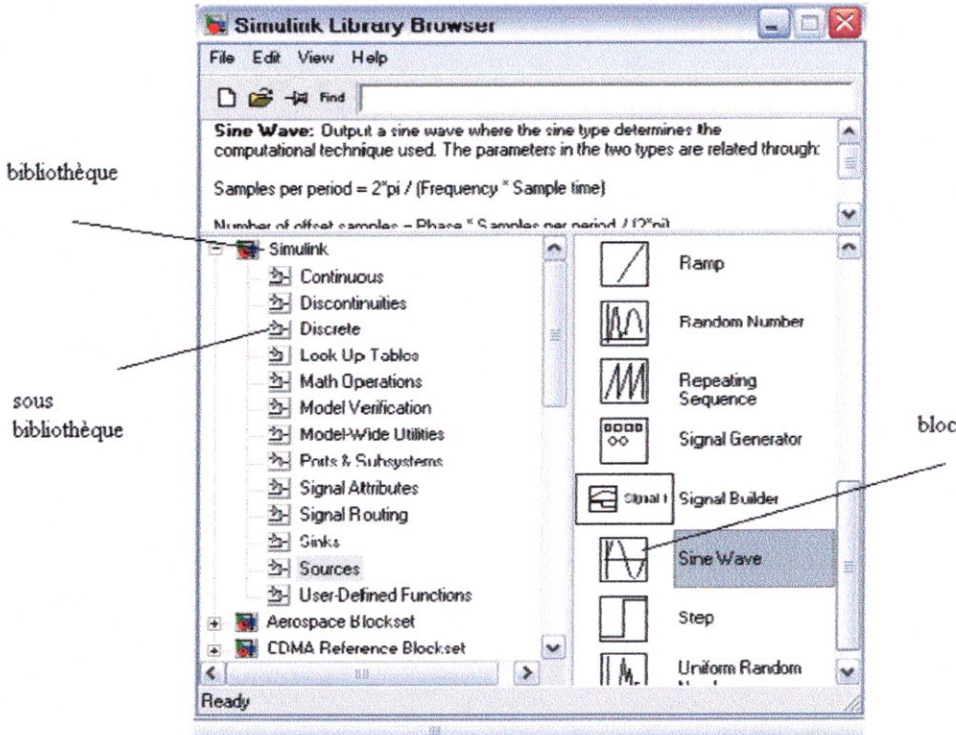


Figure II.2 : Liste des librairies de Simulink

On aboutit à l'écran représenté par la fenêtre d'exploration Simulink library contenant les différentes librairies que l'on peut ouvrir en double cliquant dessus :

Continuous

Cette bibliothèque (figure II.3) propose des dérivateurs, des intégrateurs, des fonctions de transfert, des blocs de retard pour construire des systèmes à temps continu.

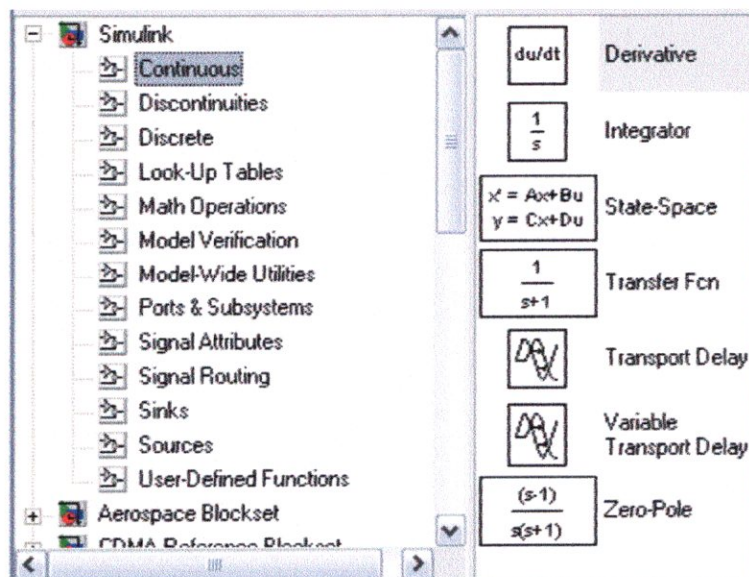


Figure II.3 : la bibliothèque continuous

Math Operations

Cette bibliothèque (figure II.4) fournit des opérations mathématiques sur les signaux telles que des gains, des multiplicateurs, des sommateurs.

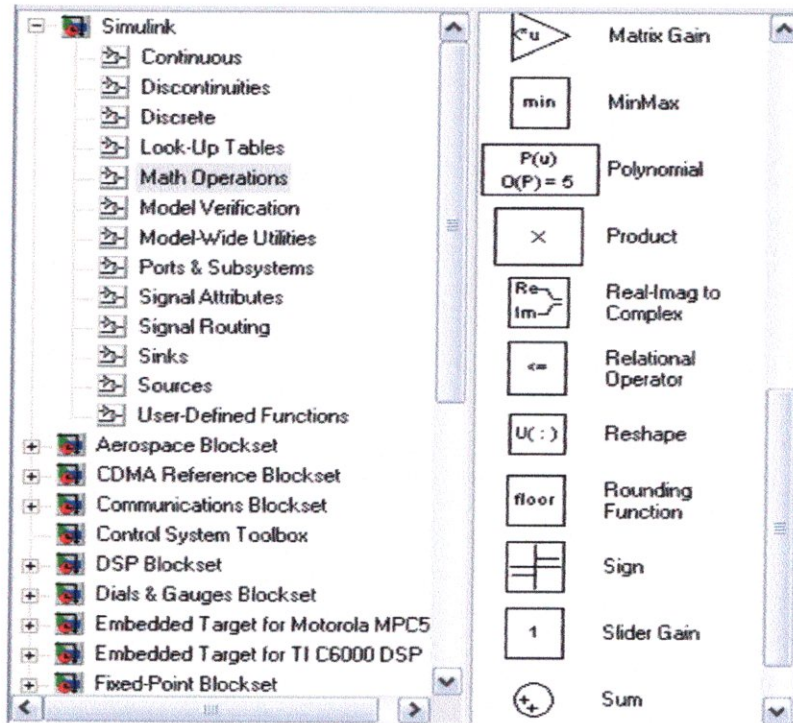


Figure II.4 : la bibliothèque math operations

Sources :

Cette bibliothèque (figure II.5) contient des éléments générateurs de signaux tels que saut unité, sinusoïde, carré, rampe, fichiers de points (*.mat), variable MATLAB, bruit, séquences, le temps courant de la simulation (horloge), etc, sans oublier le générateur de signal lui-même. Sans parler du fait que l'on peut soi-même créer son propre générateur de signal, on note ici qu'il y a vraiment intérêt à paramétrer ces blocs à l'aide de variables définies proprement dans MATLAB.

En sélectionnant par un double clic, le bloc sources, on ouvre la librairie des sources. Cette fenêtre contient des blocs qui vont permettre de simuler différents types de sources.

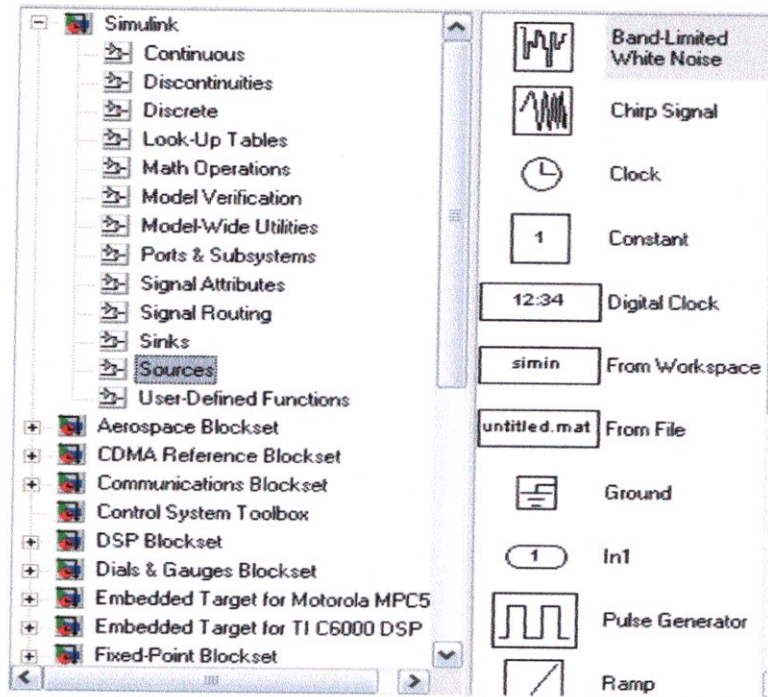


Figure II.5 : la bibliothèque sources

Sinks

Cette bibliothèque (figure II.6) contient des blocs permettant d'envoyer des données qui peuvent être traitées par un fichier *.m ainsi que des éléments de visualisation et d'affichage (scope, display). L'oscilloscope est à considérer plutôt comme un gadget permettant de vérifier le bon fonctionnement de la simulation. La combinaison de l'oscilloscope avec un multiplexeur analogique (bibliothèque Connections) permet de grouper plusieurs signaux sur une seule ligne de créer ainsi un oscilloscope multi traces. La présence d'un grand nombre d'oscilloscopes dans un même schéma ralentit considérablement la simulation, comme d'ailleurs la sauvegarde systématique des signaux dans des variables MATLAB.

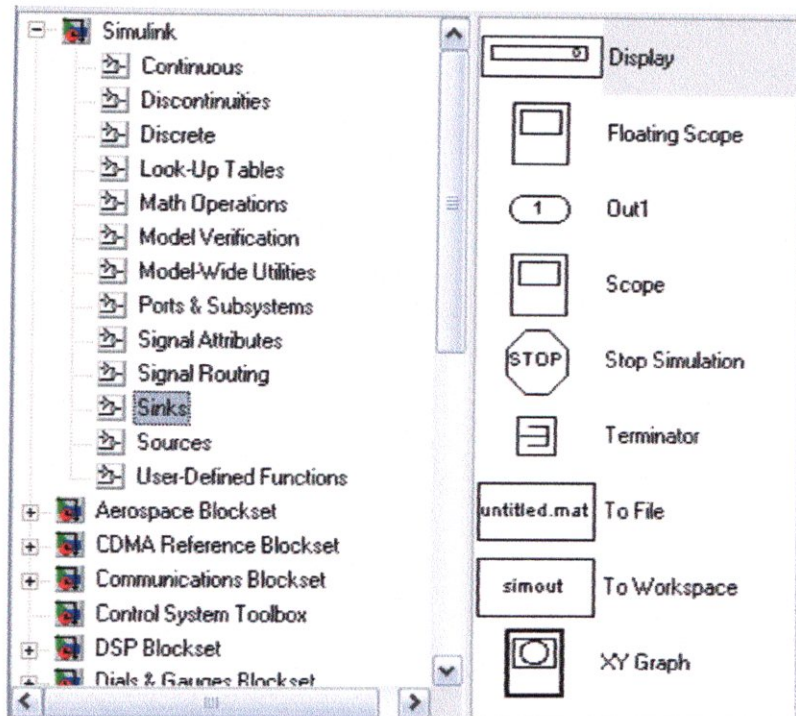


Figure II.6 : la bibliothèque sinks

II.3 Description des blocs de Simulink

Nous attardons uniquement sur les blocs les plus utiles, en particulier sur ceux dont nous avons eu besoin pour réaliser notre projet. Des informations approfondies sont disponibles dans le manuel d'utilisateur de Simulink disponible au site Internet de Mathworks.Inc :

<http://www.mathworks.com/access/helpdesk/help/toolbox/simulink/simulink.shtml>

Notre projet utilise des blocs, nous allons les citer selon les bibliothèques qui comprennent ces blocs.

II.3.1 La bibliothèque « Simulink »

Cette bibliothèque comprend des blocs d'utilisation générale.

II.3.1.1 Bloc Sine wave (générateur du sinus)

L'icône Sine Wave de la librairie « Sources » dont on fixe les paramètres (fréquence, amplitude, etc.) permet de générer un signal sinusoïdale. Le bloc est montré à la figure II.7.

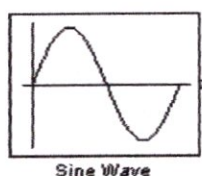


Figure II.7 : bloc de générateur du sinus

La spécification de la fréquence est en radians par seconde, alors la fréquence doit être multipliée en Hz par $2 * \pi$. La boîte de dialogue de ce bloc est montrée à la figure II.8.

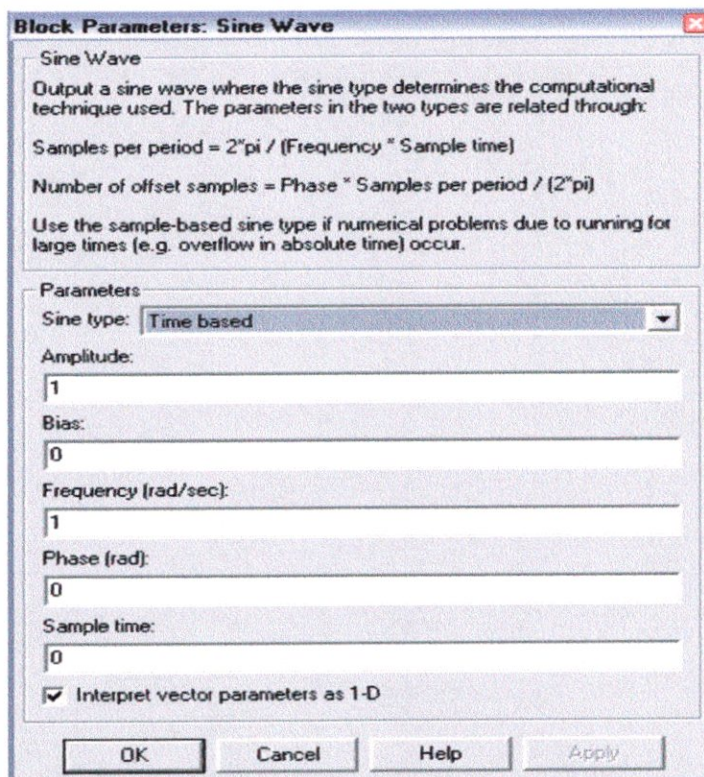


Figure II.8 : boîte de dialogue du générateur du sinus

II.3.1.2 Bloc Constant (Constant)

Le bloc Constant de la librairie « Sources » produit un signal de valeur constante, est montré à la figure II.9.

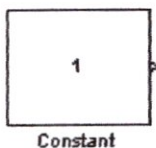


Figure II.9 : bloc de constate

Selon la dimension du paramètre de la valeur Constante, le bloc peut indiquer un scalaire, un vecteur, ou une matrice. Il est possible de fixer, via la fenêtre ci-dessous (figure II.10), la valeur de la constante.

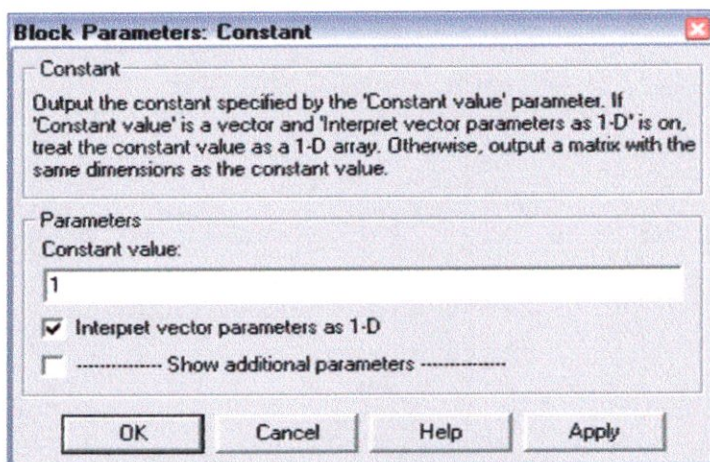


Figure II.10 : configuration du bloc constant

II.3.1.3 Bloc product (Produit)

Le Bloc Produit de la librairie « math operations » sert à multiplier ou à diviser des entrées Il est montré à la figure II.11.

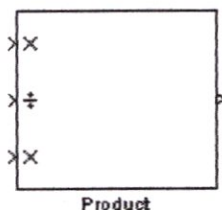


Figure II.11 : bloc de multiplication

Nous spécifions les opérations avec le paramètre du nombre d'entrées. Les symboles de multiplication (*) et de division (/) indiquent les opérations à exécuter sur les entrées. S'il y a deux d'entrées ou plus, alors le nombre de symboles doit être égal au nombre d'entrées. Par exemple, les symboles "*" / "*" exige trois entrées. Le paramétrage du bloc « Product » est montré à la figure II.12.

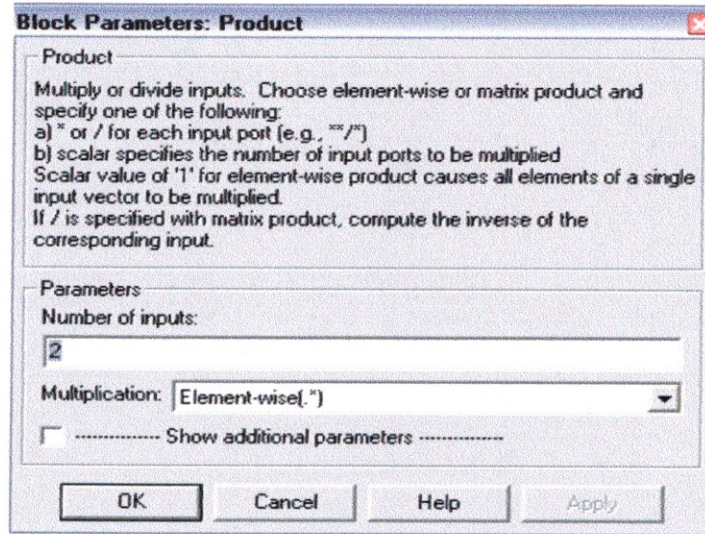


Figure II.12 : paramétrage du bloc « Product »

II.3.1.4 Bloc Sum (somme)

Pour réaliser la somme entre le signal de consigne et le signal de sortie, on choisit le bloc Sum (figure II.13), de la librairie « Math operations ».

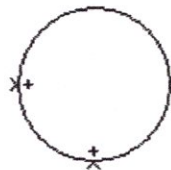


Figure II.13 : bloc « Sum »

En cliquant deux fois sur cet icône : une boîte à dialogue (figure II.14) apparaîtra et nous permet de définir le nombre d'entrées et de transformer ce bloc en soustracteur par des signes « + - ».

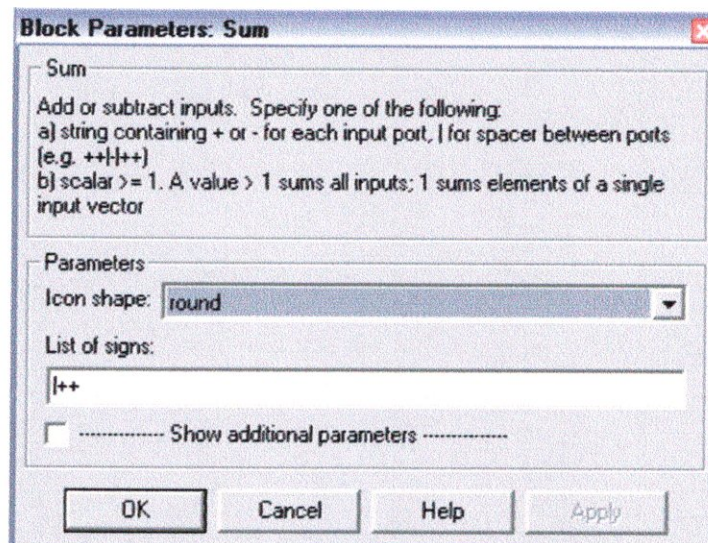


Figure II.14 : boîte de dialogue de bloc « Sum »

II.3.1.5 Quantizer (Quantificateur)

Ce bloc de la librairie « discontinuités » permet de discrétiser une entrée sur un intervalle spécifié, le bloc est montré à la figure II.15.

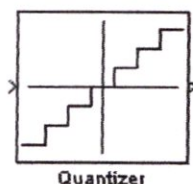


Figure II.15 : bloc de quantificateur

La quantification consiste à donner à un signal échantillonné dans un intervalle une valeur constante. Si L est le niveau de quantification, tous les points entre $L - q/2$ et $L + q/2$ prennent la valeur de L . (voir figure II.16).

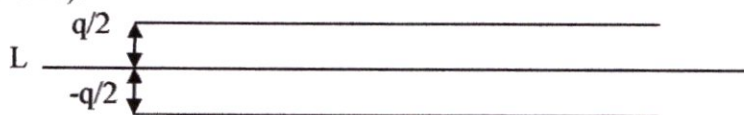


Figure II.16 : le niveau de quantification

La relation décrivant la quantification est donnée par l'expression $y = q * \text{round}(u/q)$

Où y représente la sortie, u l'entrée, et q le pas de quantification.

Le bloc est configuré à l'aide de sa boîte de dialogue montrée à la figure II.17

Le bloc est paramétré, non par une valeur numérique, mais par une variable littérale q , dont l'affectation sera réalisée dans la fenêtre de commande MATLAB. Cette méthode permet de changer la valeur de paramètres sans ouvrir à chaque fois les blocs.

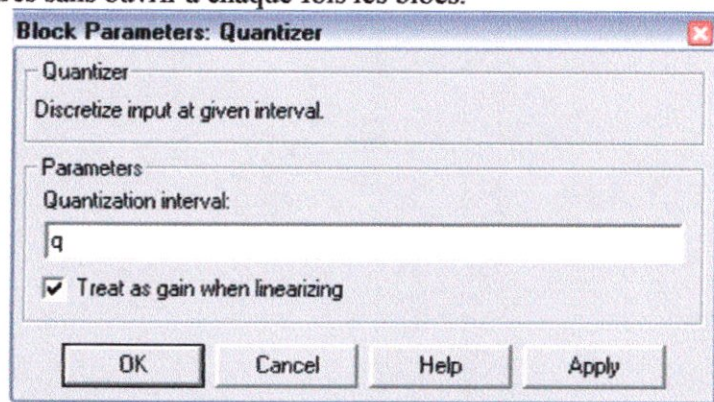


Figure II.17 : boîte de dialogue de bloc de quantificateur

Par défaut, l'intervalle de quantification est de 0.5

II.3.1.6 Bloc Mux (multiplexeur)

Il est possible de regrouper plusieurs signaux en utilisant le bloc Mux. Ceci rend ainsi un diagramme plus lisible. Le multiplexeur Mux, de la librairie « signal routing », permet d'envoyer simultanément plusieurs signaux vers un fichier ou un instrument de visualisation.

Le bloc est montré à la figure II.18.



Figure II.18 : bloc « Mux »

On effectue un double clic sur le composant Mux, la fenêtre de paramétrage (figure II.19) s'ouvre. On tape les valeurs désirées : ici la valeur 2 pour indiquer 2 entrées et on indique le modèle d'affichage de l'icône du bloc. On ferme cette fenêtre par Close, les nouvelles valeurs sont prises en compte.

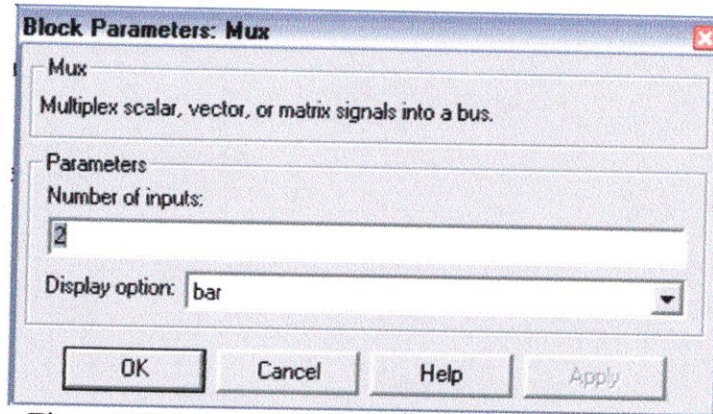


Figure II.19 : paramétrage du bloc « Mux »

II.3.1.7 Demux (Demultiplexeur)

Le démultiplexeur se trouve dans la librairie « signal routing ». Le bloc est montré à la figure II.20.



Figure II.20 : bloc de démultiplexeur

Un double clic nous permet de configurer le nombre de voies (figure II.21)

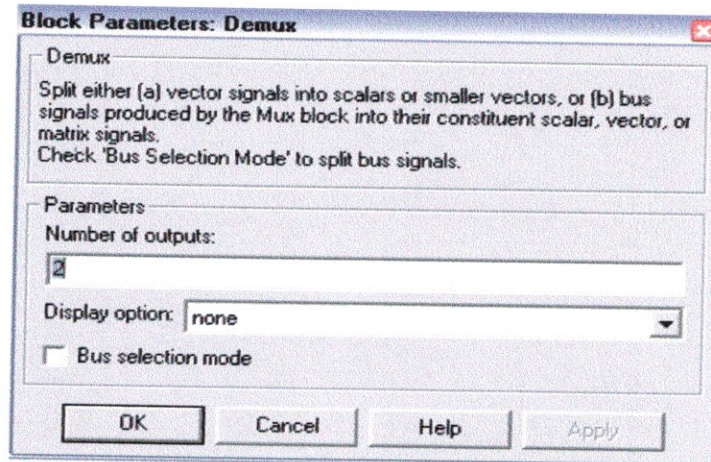


Figure II.21 : boîte de dialogue de bloc démultiplexeur

II.3.1.8 Bloc Scope (Oscilloscope)

Le bloc scope, outil de visualisation du signal, se trouve dans la librairie « Sinks ». Il est montré à la figure II.22.

Il permet la mesure des signaux, la loupe x agrandit l'axe X, la loupe y agrandit l'axe Y, la paire de jumelle permet l'adaptation d'échelle automatique, et l'icône autoscale permet d'ajuster les échelles pour voir le signal complet. On y a accès que lors de la simulation. On peut imprimer les résultats.

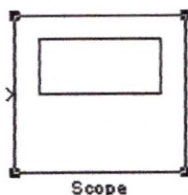


Figure II.22 : bloc « Scope »

Celui – ci permet de visualiser les courbes résultant de la simulation et aussi d'enregistrer ces données pour qu'elles soient récupérées dans Matlab.

- Double – cliquer sur le bloc « Scope ». Une fenêtre s'ouvre (voir figure II.23) : c'est là que seront affichées les courbes suite à la simulation.

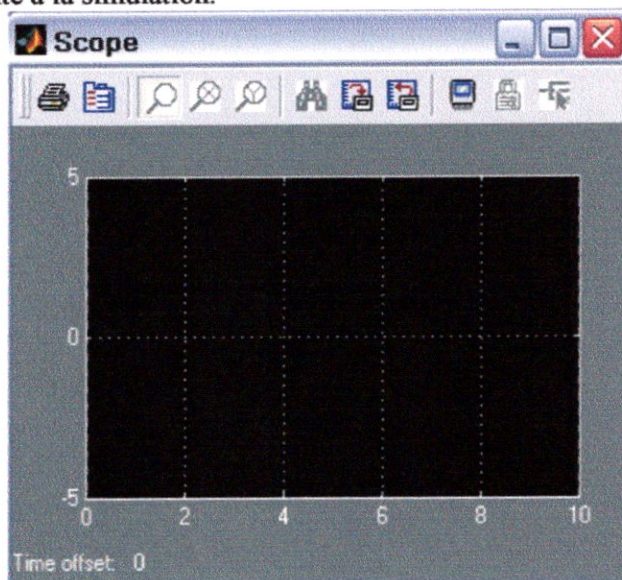


Figure II.23 : la fenêtre du « Scope »

- cliquer droite sur la fenêtre de l'oscilloscope pour modifier les propriétés de l'axe Y (figure II.24)

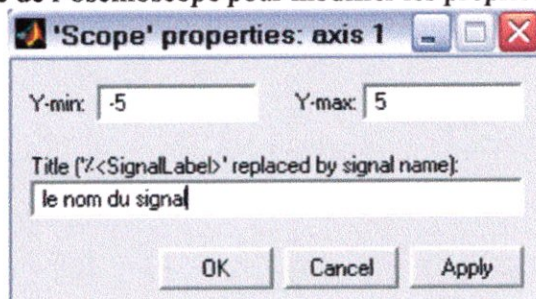


Figure II.24 : boîte de dialogue de propriétés des axes.

Il est possible d'avoir plusieurs entrées dans le bloc « Scope ». Pour ce faire, il s'agit :

- double- cliquer sur le bouton **Paramètres** de la fenêtre scope



- aller dans les deux onglets General Data History (voir figure II.25 et II.26)
- Sélectionner Save data to workspace,
- au besoin, modifier le nom de variable « Scope Data » pour le nom voulu,
- choisir le format Array.
- Cliquer sur l'onglet **General** et configurer les paramètres du bloc scope suivant la figure II.25
- Cliquer sur le bouton **OK** pour confirmer

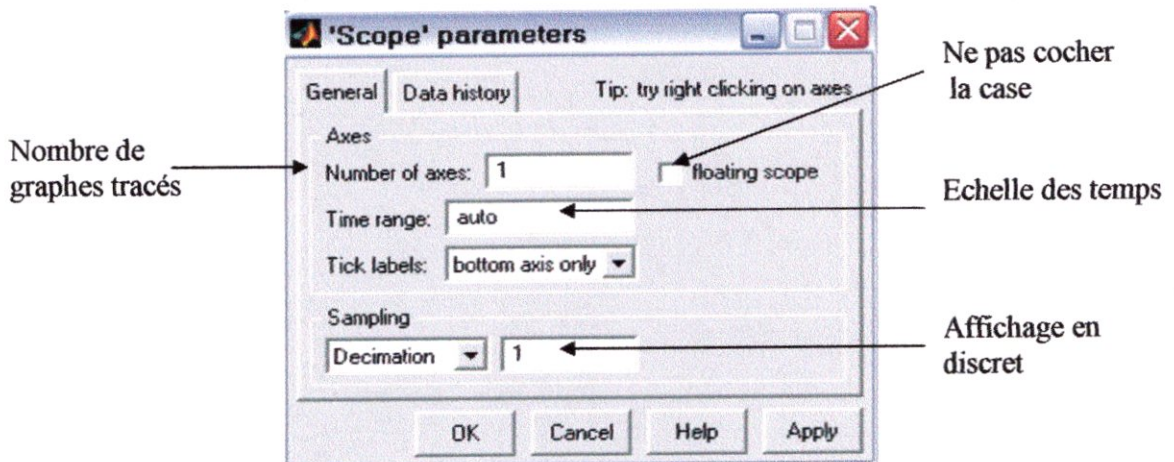


Figure II.25 : onglet permettant d'ajuster les paramètres du « Scope »

L'oscilloscope « Scope » ne conserve que les dernières données de la simulation. Il est possible de fixer le nombre de points conservés en l'indiquant à la suite de la mention **Limit data points to last** de la fenêtre **Data History** (figure II.26). Par défaut, le nombre de points est fixé à 5000, ce qui s'avère généralement adéquat.

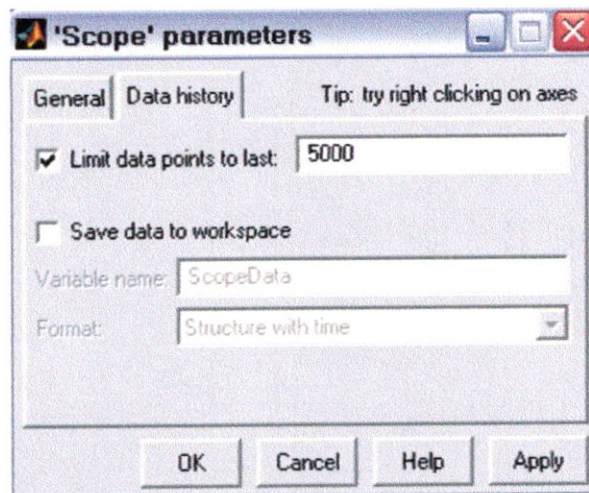


Figure II.26 : onglet permettant d'ajuster les paramètres du « Scope »

II.3.1.9 Bloc To Workspace

Pour récupérer les données générées par la simulation, il faut insérer dans le schéma un bloc provenant de la librairie « Sinks ». Le bloc « to workspace » enregistre les données dans Matlab mais ne les affiche pas.

Le bloc To Workspace est montré à la figure II.27.

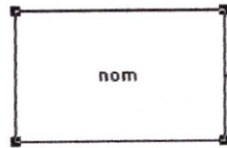


Figure II.27: bloc « To Workspace »

Pour ce bloc, il faut suivre les étapes suivantes avant l'exécution de la simulation :

- double – cliquer sur le bloc,
- donner le nom de variable voulu,
- fixer le nombre de points à conserver (**Limit data points last**),
- enregistrer le format de sauvegarde (Save format) du bloc sous la forme **Array**.

Ce bloc a pour paramètres (figure II.28) le nom de la variable MATLAB dans laquelle les résultats seront sauvés, ainsi qu'une indication relative au nombre maximum de pas de calcul à sauver. Si la durée du pas est faible et que celle de la simulation est grande, on conçoit que le nombre de données à sauver soit prohibitif et provoque une saturation de l'espace mémoire. Pour l'éviter en partie, on peut en option spécifier au bloc To Workspace de ne sauver qu'à intervalles réguliers, de durée plus grande que le pas d'intégration.

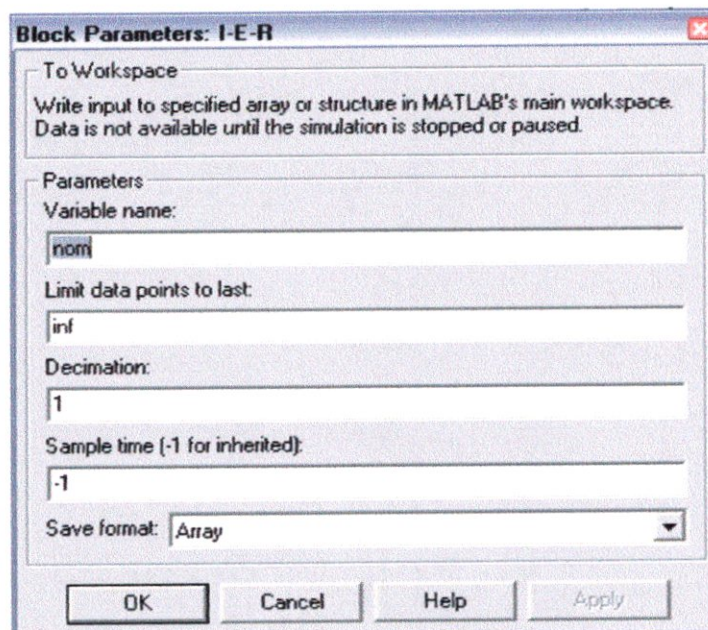


Figure II.28 : boîte de dialogue de bloc à workspace

Une fois la simulation exécutée, on sauve les données dans des variables en écrivant dans la fenêtre de commande Matlab ou encore dans le M-File les commandes :

```
t = nom( : , 1)
t = nom( : , 2)
```


Le nom « nom » est le nom de variable que l'on a choisi lorsque l'on a ajusté les paramètres du bloc « To Workspace ». Les données de la première colonne de cette variable correspondent au temps où ont été prélevés les échantillons, tandis que les données de la deuxième colonne sont les résultats.

II.3.1.10 Display (display)

Le bloc Display de la librairie « sinks » sert à afficher la valeur de l'entrée, est montré à la figure II.29.

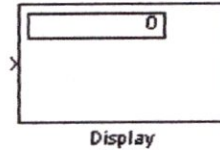


Figure II.29: bloc « Display »

Le type de format de l'affichage est sélectionné comme suit :

- ❖ Format short : affiche une valeur de 5 chiffres avec une virgule fixe.
- ❖ Format Long : affiche une valeur de 15 chiffres avec une virgule fixe.
- ❖ Format short_e : pour afficher les résultats numériques en notation scientifique.
- ❖ Format long_e : affiche une valeur de 16 chiffres avec une virgule flottante.
- ❖ Format Banque : affiche une valeur en dollars fixes et les cent formats

Le paramétrage de ce bloc est donné à la figure II.30.

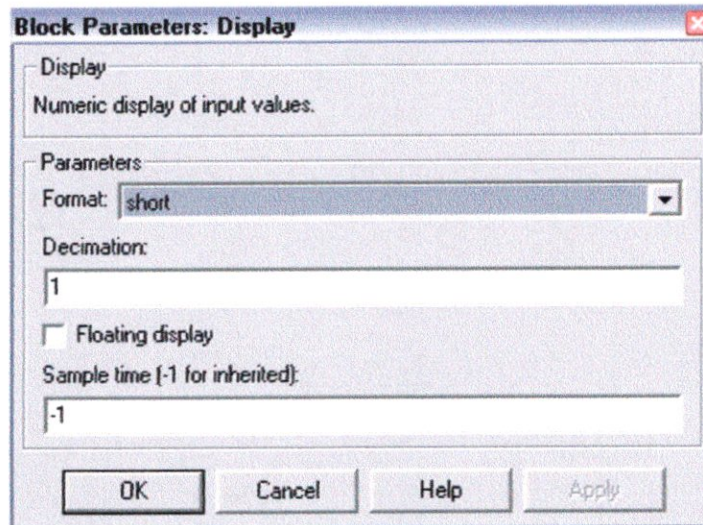


Figure II.30 : boîte de dialogue du bloc « Display »

II.3.1.11 Zero-order hold (Bloqueur d'ordre zero)

Le bloc du bloqueur d'ordre zéro de la librairie « Discrete » est présenté à la figure II.31.

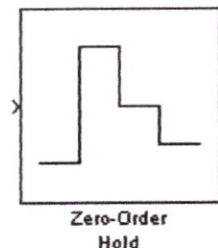


Figure II.31: bloc « Zero – order hold »

Le bloc du bloqueur des échantillons bloque le signal d'entrée pour une période d'échantillonnage donnée. Le paramétrage de ce bloc est donné à la figure II.32.

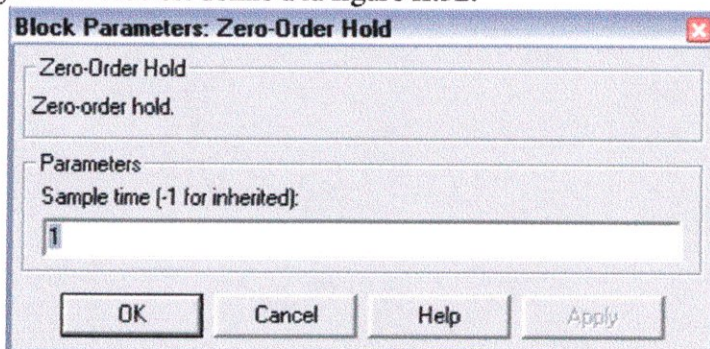


Figure II.32 : paramétrage du bloc « Zero – order hold »

II.3.1.12 Bloc Terminator (Termineur)

Il se trouve dans la librairie « sinks ». Il faut placer ce termineur dès qu'une borne de sortie n'est pas utilisée

Le bloc est montré à la figure II.33.

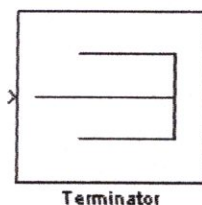


Figure II.33 : bloc de termineur

La boîte de dialogue est montrée à la figure II.34.

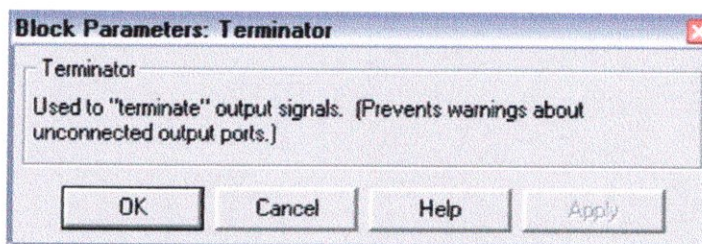


Figure II.34 : boîte de dialogue de bloc termineur

II.3.1.13 Bloc transport delay

Le bloc « Transport delay » de la librairie « continuous » produit un retard pur de l'entrée, est montré à la figure II.35.

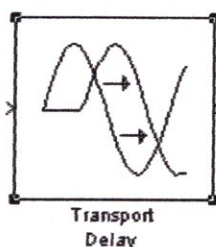


Figure II.35 : bloc de retard

Il est utilisé pour simuler un délai du temps. Au début de la simulation, le bloc produit une entrée initiale jusqu'à ce que le temps de la simulation dépasse le temps de délai, quand le bloc commence à produire l'entrée décalée. Le temps de délai ne doit pas être négatif. La boîte de dialogue est montrée à la figure II.36.

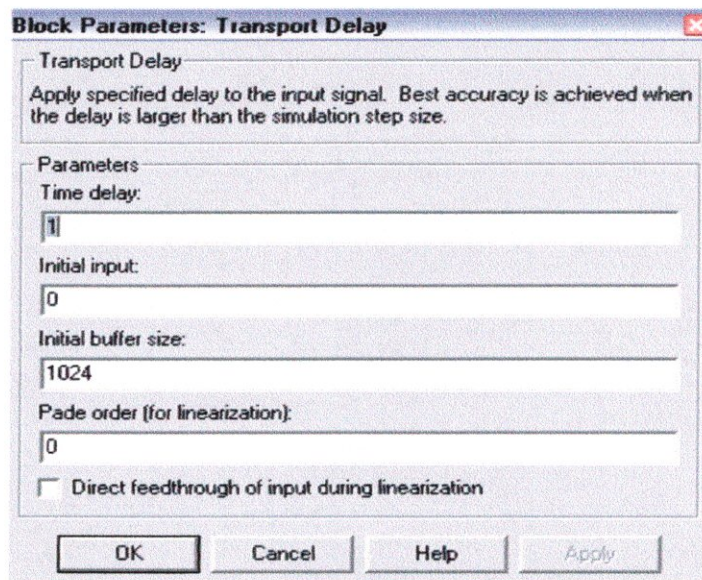


Figure II.36 : boîte de dialogue de bloc de retard

II.3.1.14 Sign (Signe)

Le bloc « Sign » du groupe « math operations » indique le signe de l'entrée, est montré à la figure II.37.

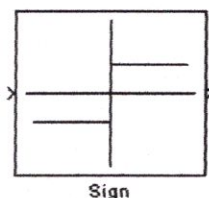


Figure II.37 : bloc de signe

La sortie est :

- ❖ 1 quand l'entrée est plus grande que zéro.
- ❖ 0 quand l'entrée est égale à zéro.
- ❖ -1 quand l'entrée est inférieure à zéro.

Le bloc de Signe accepte les signaux de tous les types.

La boîte de dialogue est montrée à la figure II.38.

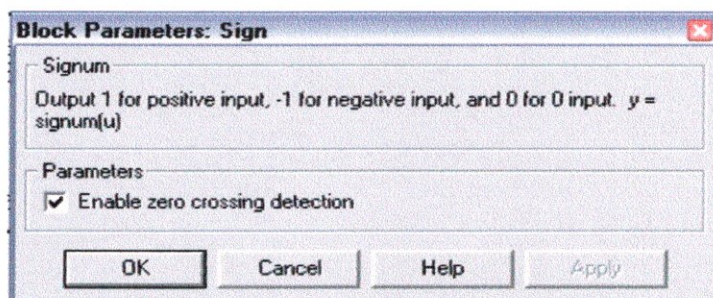


Figure II.38 : boîte de dialogue de bloc de signe

II.3.1.15 Bloc de Saturation (Saturation)

Ce bloc de la librairie « discontinuities » permet de limiter la tension de sortie d'un signal entre deux valeurs +V et -V. Il est montré à la figure II.39.

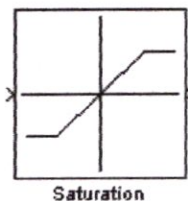


Figure II.39 : bloc de saturation

Le bloc de la Saturation impose des limites supérieures et inférieures à un signal. Le paramétrage de ce bloc est donné à la figure II.40.

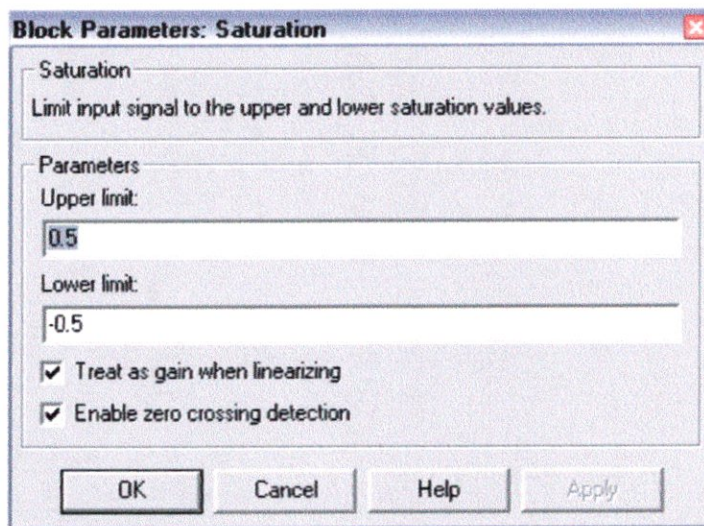


Figure II.40 : boîte de dialogue de bloc de saturation

II.3.1.16 Bloc Inport (Bloc d'entrée)

Ce bloc de la librairie « Ports & subsystem » permet de créer un bloc d'entrée pour un sous-système ou une entrée externe. Celui – ci est montré à la figure II.41.



Figure II.41 : Bloc Inport

Le bloc d'entrée sert de lien entre l'extérieur d'un système et l'intérieur d'un système. Le paramétrage du bloc Inport est montré à la figure II.42.

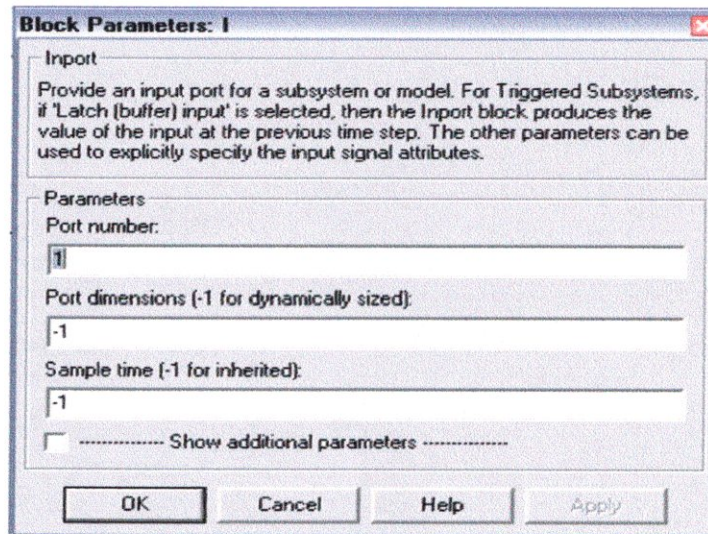


Figure II.42 : boîte de dialogue de bloc inport

On spécifie le nombre, la dimension du port du bloc inport et la période d'échantillonnage du signal d'entrée.

II.3.1.17 Bloc Output (Bloc de sortie)

Ce bloc de la librairie « Ports & subsystem » permet de créer un bloc de sortie pour un sous-système ou une production externe. Le bloc est montré à la figure II.43.

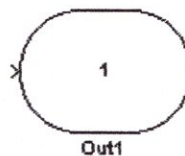


Figure II.43 : porte de sortie

Le bloc de sortie est le lien d'un système à une destination à l'extérieur du système. La boîte de dialogue est montrée à la figure II.44.

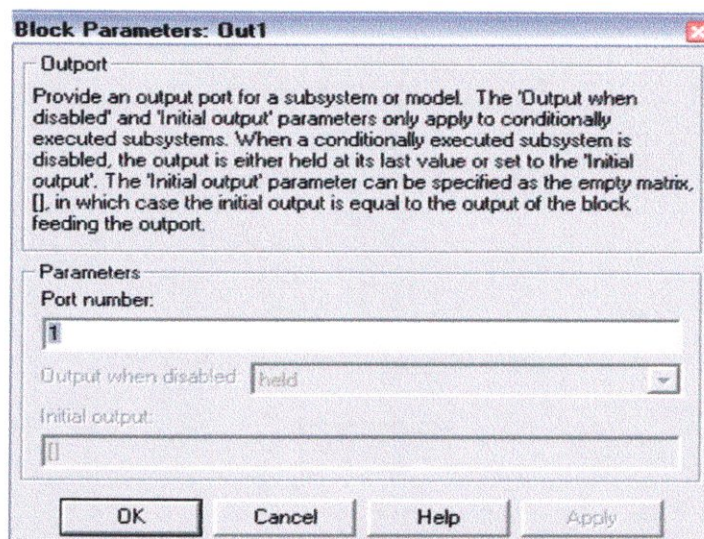


Figure II.44 : boîte de dialogue de bloc de porte de sortie

On spécifie le nombre de port du bloc output. La sortie désactivée, et la sortie initiale sont utilisées avec des systèmes spéciaux.

II.3.2 La bibliothèque « communications »

Cette bibliothèque comprend des blocs qui sont utilisés en communication.

II.3.2.1 Random integer generator (Générateur des nombres entiers aléatoires)

Ce bloc de la bibliothèque « comm Sources » sert à produire aléatoirement des nombres entiers distribués dans la gamme $[0, M - 1]$.

Le bloc est montré à la figure II.45.

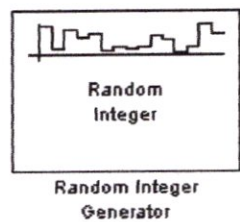


Figure II.45 : bloc de générateur des nombres entiers aléatoires
La boîte de dialogue est montrée à la figure II.46.

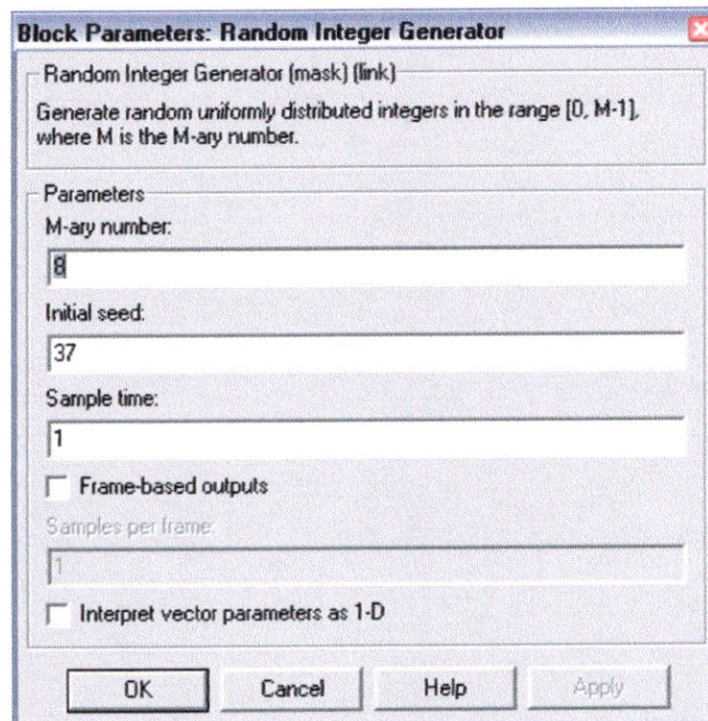


Figure II.46 : boîte de dialogue de bloc de générateur des nombres entiers aléatoires

II.3.2.2 Bit to integer converter (Convertisseur bits _entier)

Ce bloc de la bibliothèque « utility functions » convertit un groupe de bits présent à l'entrée en un nombre entier en sortie.

Le bloc est montré à la figure II.47.

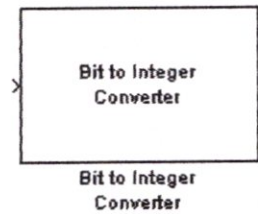


Figure II.47 : bloc de convertisseur bits-entier

Si le M est le nombre de bits par entier, alors le bloc convertit chaque groupe de M bits en un nombre entier compris 0 et $2^M - 1$.

La boîte de dialogue est montrée à la figure II.48.

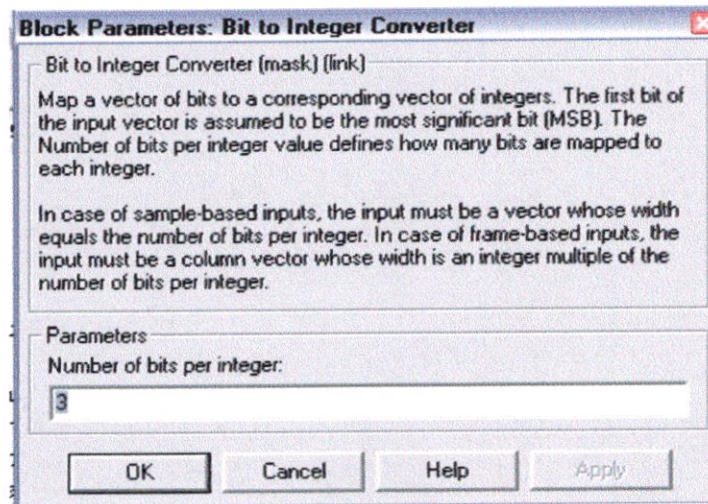


Figure II.48 : boîte de dialogue de bloc convertisseur bits entier

Exemple

Si l'entrée est [0; 1; 1; 1; 1; 1; 0; 1] et le paramètre du nombre de bits par entier est quatre, alors la sortie est [7; 13].

II.3.2.3 Integer to bit converter (Convertisseur entier-bits)

Ce bloc de la librairie « utility functions » convertit un entier en un groupe de bits, est montré à la figure II.49.

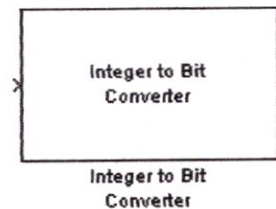


Figure II.49 : bloc de convertisseur entier bits

Si le M est le nombre de bits par entier, alors l'entrée est un nombre entier compris entre 0 et $2^M - 1$. La boîte de dialogue est montrée à la figure II.50.

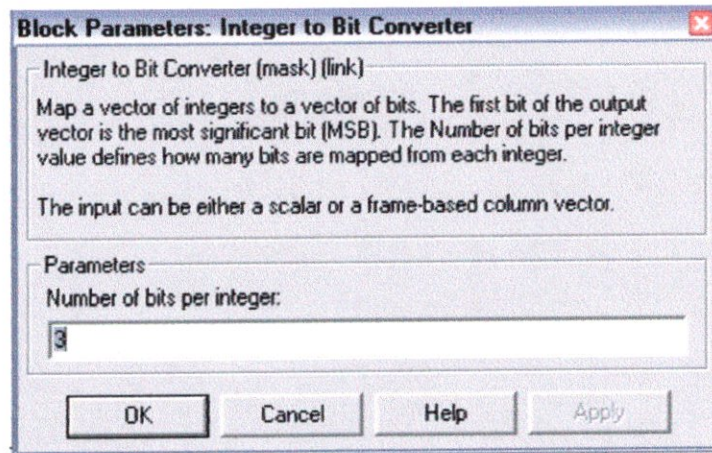


Figure II.50 : boîte de dialogue de bloc convertisseur entier bits

Le nombre de bits par entier doit être un nombre entier entre 1 et 31.

II.3.2.4 AWGN (Additive White Gaussian Noise) channel (canal AWGN)

Ce bloc de la librairie « channels » permet d'ajouter le bruit Gaussien blanc au signal d'entrée, est montré à la figure II.51.

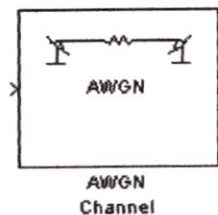


Figure II.51 : bloc de canal

Ce bloc hérite son temps de l'échantillon du signal de entrée.

Le paramétrage du bloc AWGN Channel est donné à la figure II.52.

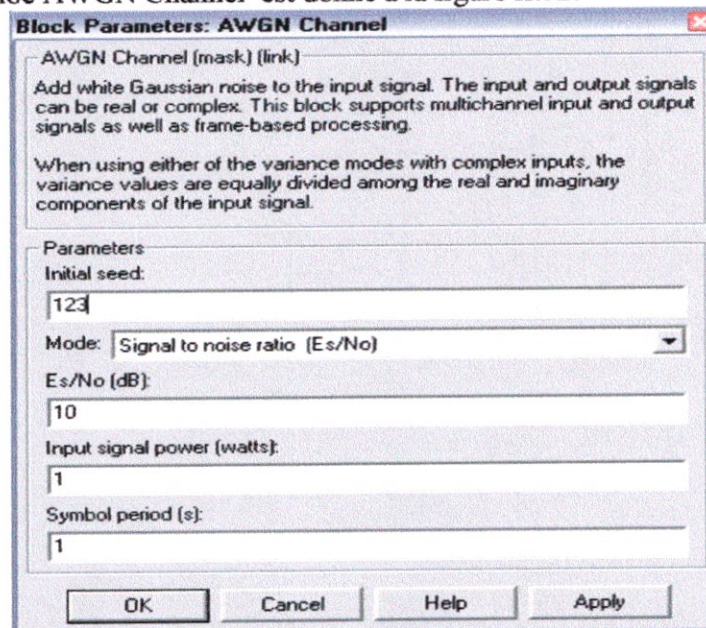


Figure II.52 : boîte de dialogue de bloc canal AWGN

II.3.2.5 PLL (boucle à verrouillage de phase)

Le bloc PLL de la bibliothèque « synchronisation » sert à implémenter une boucle à verrouillage de phase pour retrouver la phase du signal d'entrée, est montré à la figure II.53.

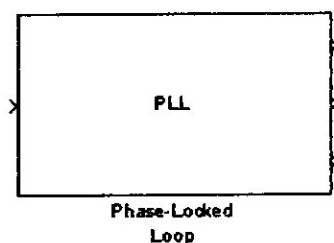


Figure II.53 : bloc de PLL

La Boucle à verrouillage de phase PLL (phased locked loop) est un système de contrôle de la réaction qui automatiquement ajuste la phase d'un signal généré localement pour égaler la phase d'un signal d'entrée. Ce PLL se compose de trois composants :

- ❖ Un multiplicateur a utilisé comme un comparateur de phase.
- ❖ Un filtre passe bas, dont la fonction du transfert est défini par un numérateur et un dénominateur. Ces derniers sont des vecteurs dont les coefficients du polynôme respectif sont en ordre de puissance décroissant de s . La fréquence de coupure du filtre est choisie très inférieure à la fréquence du signal d'entrée.
- ❖ Un oscillateur commandé en tension VCO (voltage controlled oscillator). Les caractéristiques du VCO sont spécifiés par les trois paramètres suivants : la fréquence tranquille de VCO, la phase initiale de VCO, et l'amplitude de sortie du VCO.

Le schéma synoptique d'une PLL est montré à la figure II.54.

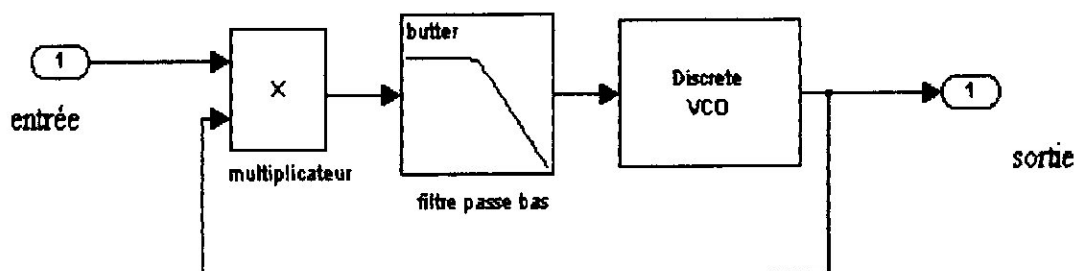


Figure II.54 : le schéma synoptique d'une PLL

On clique deux fois sur le bloc afin d'accéder aux paramètres : une boîte de dialogue apparaîtra comme indiquée à la figure II.55. La paramétrisation du filtre passe bas est représentée par deux polynômes numérateur et dénominateur avec les coefficients de la variable "s" ou p en ordre décroissant. N'oubliez pas que les polynômes doivent être complets : remplir avec des "0" dans l'occurrence.

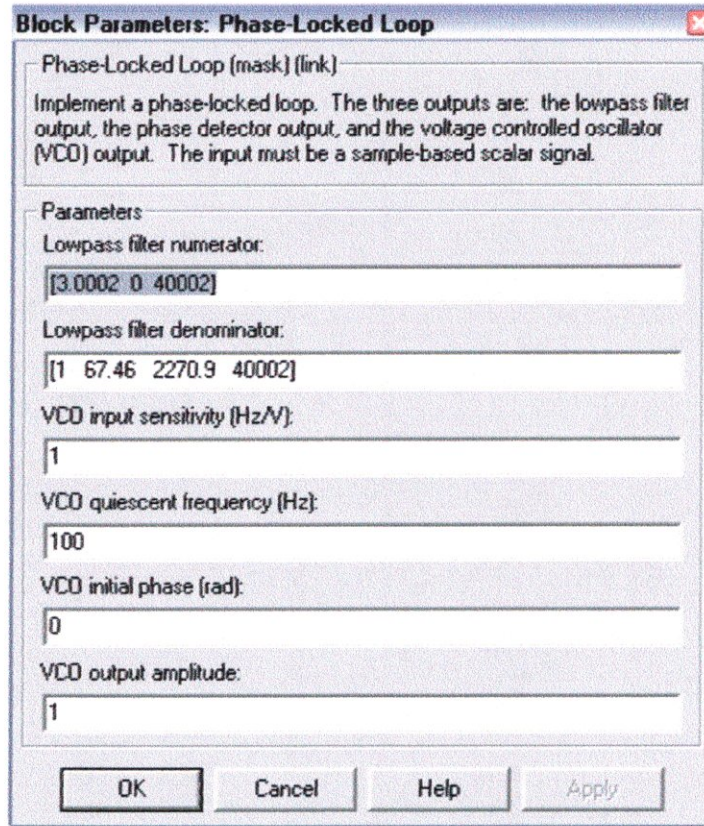


Figure II.55 : boîte de dialogue de PLL

II.3.2.6 Error rate calculation (Calcul du Taux de l'erreur).

Ce bloc de la librairie « comm sinks » calcule le taux d'erreur du bit ou taux d'erreur du symbole des données d'entrée, est montré à la figure II.56.

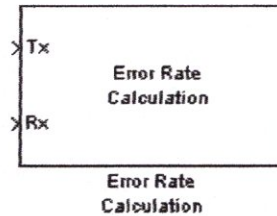


Figure II.56 : bloc de calcul de taux d'erreur

Le bloc du Calcul du Taux de l'Erreur compare les données d'entrée d'un émetteur avec les données d'entrée d'un récepteur. La boîte de dialogue est montrée à la figure II.57.

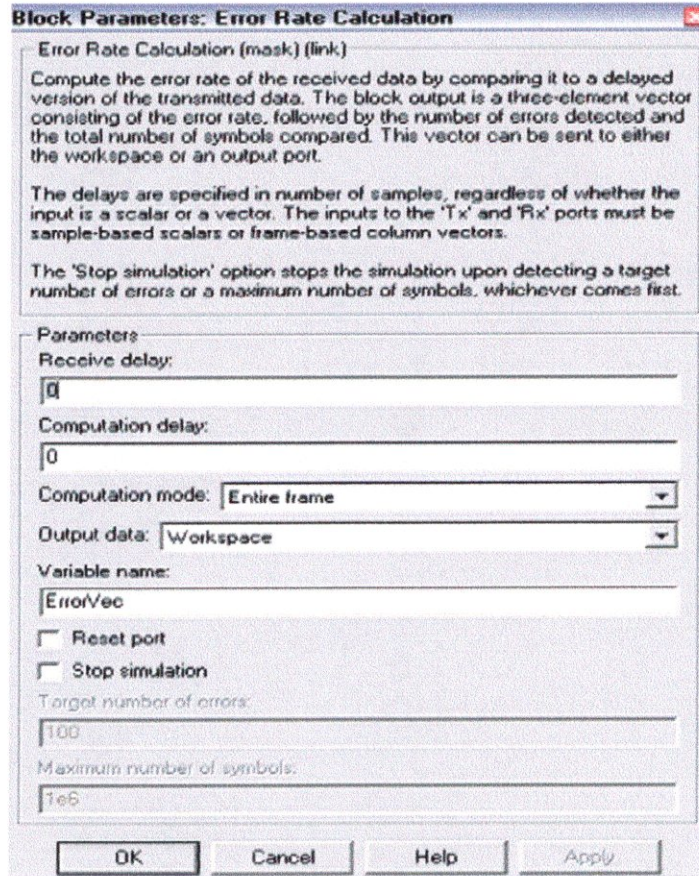


Figure II.57 : boîte de dialogue de bloc de calcul de taux d'erreur

II.3.3 La bibliothèque « DSP » Digital Signal Processing (traitement numérique du signal).

II.3.3.1 Analog filter design (Dessin du Filtre analogique)

Ce bloc de la bibliothèque « filtering » sert à construire un filtre analogique, est montré à la figure II.58.

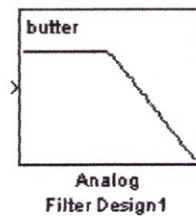


Figure II.58 : bloc de filtre analogique

Le bloc est un filtre de type : Butterworth, Chebyshev type I, Chebyshev type II, ou elliptique, passe-haut, passe-bas, passe-bande, ou stop bande. L'entrée doit être **un signal scalaire échantillon basé**. Le paramétrage du bloc (type de filtre, ordre du filtre, bande passante) est donnée à la figure II.59.

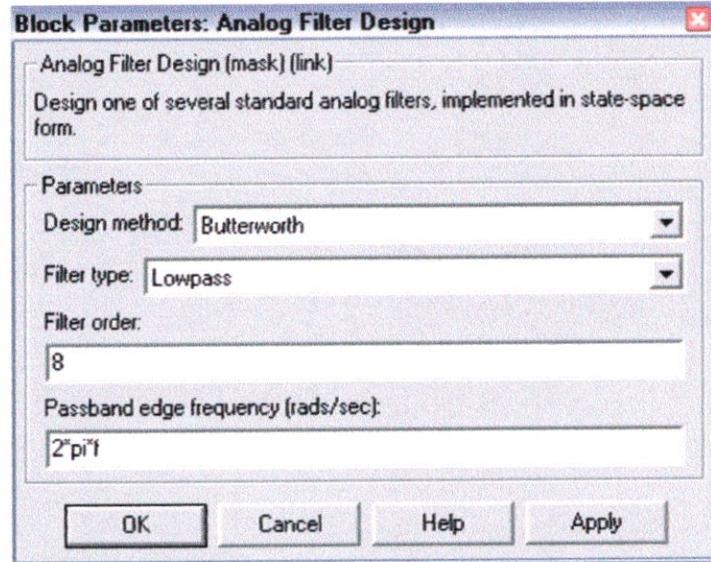


Figure II.59 : boîte de dialogue de filtre analogique

II.3.3.2 Simple and hold (échantillonneur- bloqueur)

Le bloc Simple and hold de la librairie « signal operations » permet d'échantillonner et de bloquer un signal d'entrée, est montré à la figure II.60.

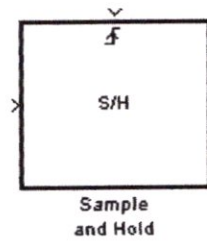


Figure II.60 : bloc de échantillonneur bloqueur

A chaque impulsion d'horloge, le bloc échantillonneur- bloqueur maintient la sortie à la valeur de l'entrée jusqu'à l'impulsion suivante de l'horloge. La boîte de dialogue est montrée à la figure II.61.

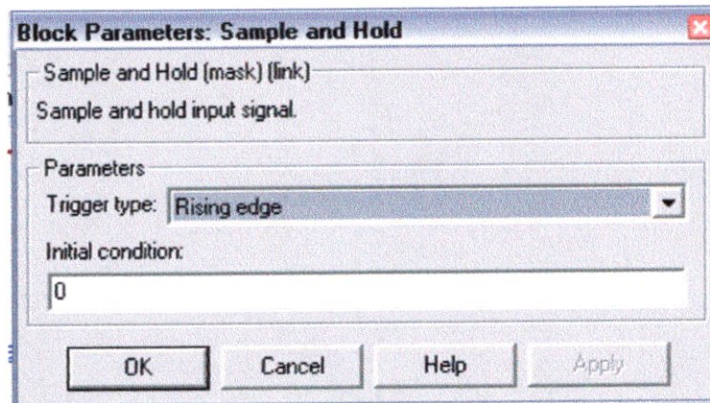




Figure II.61 : boîte de dialogue de bloc échantillonneur bloqueur

II. 4 Création des schémas blocs

Les schémas blocs doivent être créés dans un fichier Simulink distinct, de type *.mdl. Pour créer un nouveau fichier de travail Simulink : faire File, New, Model ou cliquer sur l'icône 

Pour ouvrir un ancien fichier : faire File, open ou cliquer sur l'icône 

Pour insérer un bloc dans un fichier, il faut cliquer sur le bloc désiré puis le faire glisser jusqu'à l'endroit approprié dans le fichier de travail (fenêtre du modèle), en maintenant le bouton gauche de la souris enfoncé. Relâchez le bouton de la souris lorsque le bloc est placé à l'endroit voulu (figure II.62).

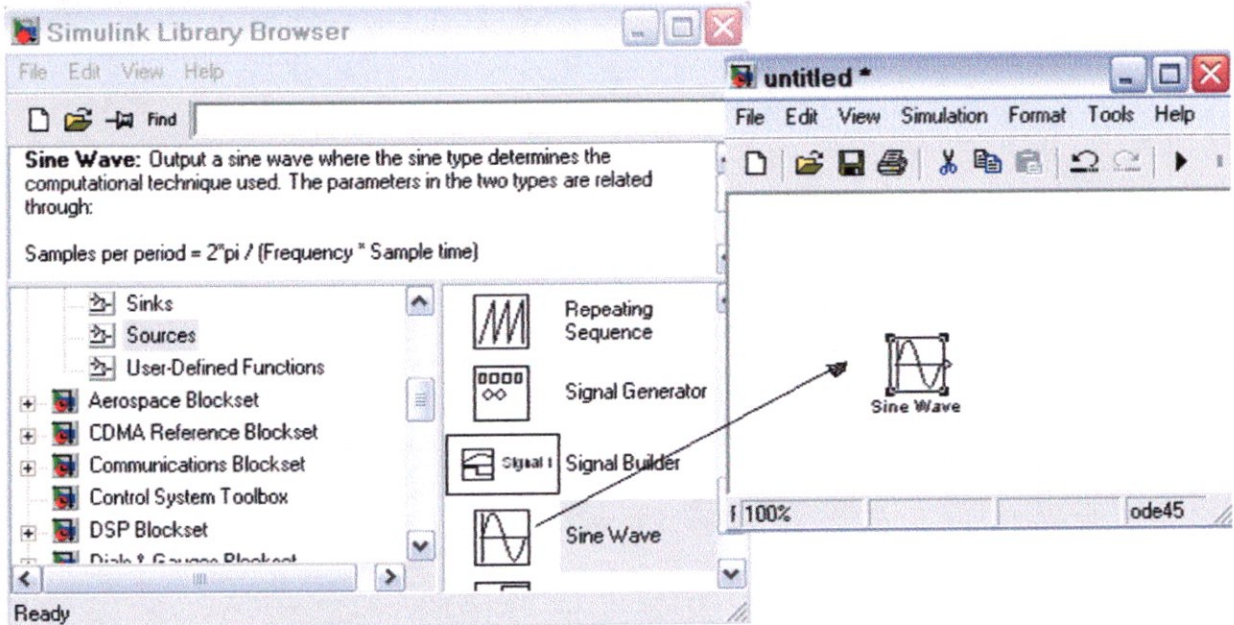


Figure II.62 : insérer un bloc dans un modèle

On peut rechercher un bloc de la liste en tapant une partie de son nom dans la case blanche apparaissant en haut de la librairie, puis en appuyant sur Retour.

Une fois tous les éléments du modèle placés, il faut les relier avec des connexions. Glissez le pointeur jusqu'à la sortie d'un élément : nous pouvons l'identifier par le sens de la flèche. Le pointeur change de forme et devient une croix. Cliquons puis glissons : un trait apparaît. Déplacez le trait jusqu'à l'entrée de l'élément correspondant (le pointeur change encore vers une croix à double trait) et relâchez la souris. La connexion est complétée (figure II.63).

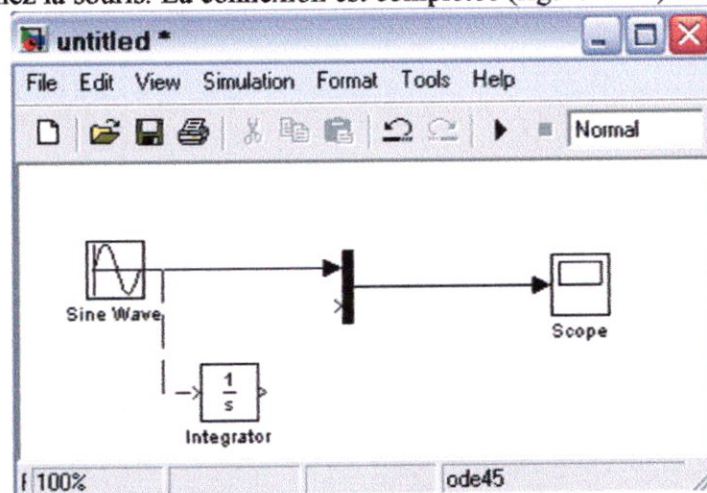


Figure II.63 : les connexions entre les blocs

Nous pouvons éditer et effacer un trait. Cliquons une fois sur le trait afin de l'activer (ses composants droits apparaîtront). Pressons "delete" pour l'effacer en entier ou bien cliquons sur les points d'union pour les déplacer. Il est aussi possible de déplacer un segment du trait et même d'ajouter un nouveau trait à partir de celui en cliquant avec le bouton droit.

Une fois les blocs insérés dans le fichier, leurs paramètres doivent être ajustés. Ceci se fait en double – cliquant sur un bloc. Une fenêtre de dialogue apparaît (on a déjà vu) dans laquelle on insère les valeurs appropriées.

Une fois les différents blocs assemblés, on sélectionne l'ensemble à la souris et on les groupe à l'aide de la commande *create subsystem* pour obtenir un seul bloc.

Un double clic sur l'icône sous-système permet l'ouverture de la fenêtre dans laquelle on retrouve les icônes assemblées auxquelles Simulink a ajouté automatiquement les blocs d'entrée Inport et de sortie Outport (figure II.64).

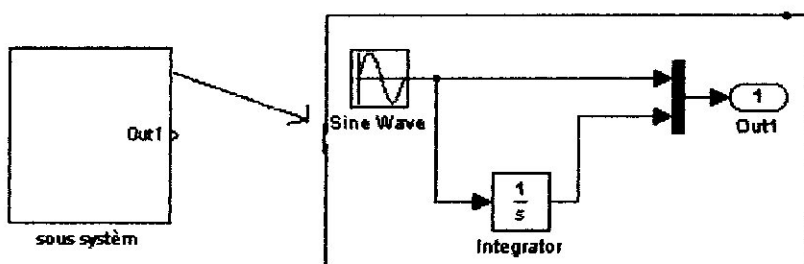


Figure II.64 : le contenu du sous système

II. 5 Ajustement des paramètres de la simulation

Après avoir créé le schéma à exécuter, cliquons dans le menu « Simulation » de la barre de menu dans le haut de la fenêtre où se trouve le schéma, puis sélectionnons « Normal ». Ensuite, toujours dans Simulation, il faut :

- cliquer sur Simulation Parameters,
- aller dans l'onglet Solver,
- définir les temps de début et de fin de simulation (par défaut, ces temps sont 0s et 10s respectivement)
- choisir le mode de simulation
- la boîte de dialogue de paramètres de la simulation est montrée à la figure II.65.

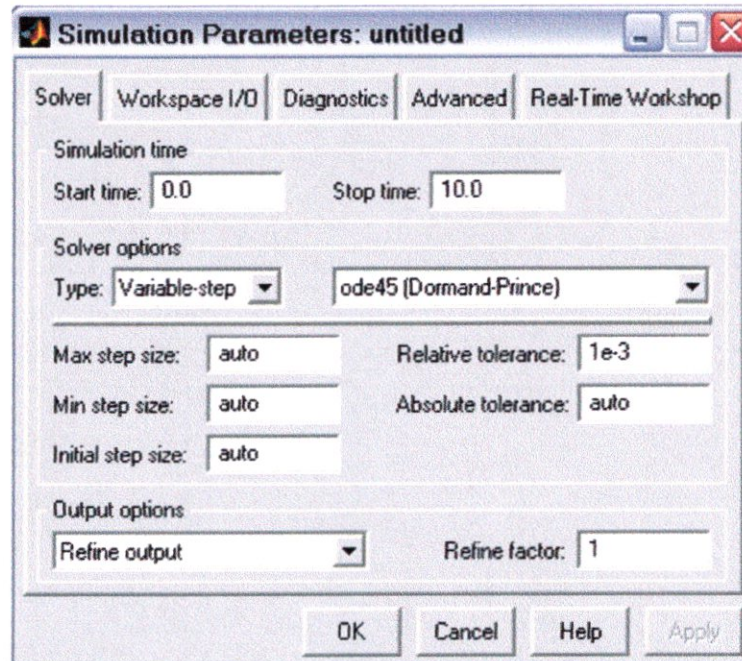



Figure II.65 : la boîte de dialogue de paramètres de la simulation

Enfin, lorsque les paramètres sont entrés, la simulation peut être amorcée de la façon suivante :

- cliquer sur l'icône  de la barre d'outils (flèche noire)
- ou aller dans l'item Simulation du menu et faire Start.

Le temps de simulation apparaît dans la case en bas du fichier.

II.6 Conclusion

Ce deuxième chapitre se veut seulement un « tour d'horizon » qui nous a permis de décrire les outils de la bibliothèque de Simulink et la fonctionnalité de ces blocs qui sont utilisés dans notre projet. Nous avons trouvé que Simulink est un outil de modélisation et de simulation facile avec une grande efficacité, où un bloc peut remplacer l'écriture d'un programme qui peut être compliqué, avec la possibilité de visualisation à n'importe quelle point du modèle.

Chapitre III

Simulation sous MATLAB/SIMULINK d'une chaîne de transmission numérique radio fréquence

Références [4], [6], [7], [8]

III.1 Introduction	52
III.2 Présentation de la chaîne de transmission	52
III.3 Gamme de fréquences.	54
III.4 Description du travail sous MATLAB.	54
III.5 Conclusion	69

III Simulation sous MATLAB/SIMULINK

d'une chaîne de transmission numérique radio fréquence

III.1 Introduction

La simulation est un outil de prédiction permettant d'étudier et d'analyser le comportement des systèmes complexes afin de prendre des décisions de façon plus objective et scientifique.

En simulation, le système est modélisé par un ensemble de structures de données interconnectées et un ensemble de procédures qui opèrent sur ces structures définissant ainsi la dynamique du système.

La simulation fonctionnelle de l'ensemble (émetteur -canal- récepteur) à modulation d'amplitude en quadratique à 16 états (QAM-16) est faite au moyen du logiciel simulink.

Ce choix a été motivé par les possibilités futures qui semblent offrir ce logiciel pour modéliser de manière fonctionnelle les différents blocs du système en numérique.

Dans ce contexte nous avons développé une méthode de modélisation et de simulation qui repose sur l'atelier Matlab/Simulink.

De plus, le caractère modulaire de Matlab/Simulink permet de découper la chaîne de transmission en blocs différents et ainsi de bien dissocier les problèmes : C'est-à-dire que la chaîne de transmission doit faire apparaître clairement les modules de base qui sont entre autres, l'émetteur pour la mise en forme physique des signaux, le canal de transmission, et le récepteur.

III.2 Présentation de la chaîne de transmission

La chaîne de transmission est composée de :

- ❖ Un émetteur
- ❖ Un canal de transmission, afin de modéliser les perturbations dues à la transmission dans l'air.
- ❖ Un récepteur.

Le schéma synoptique général de la chaîne de transmission est présenté à la figure III.1.



Figure III.1 : schéma synoptique de la chaîne de transmission

La chaîne de transmission permet de transmettre une information sous forme de données numériques codées en binaire.

Le train de donnée binaire est séparé en deux « sous trains » appelés I et Q. chaque « sous trains » est codé sur quatre niveaux ; chaque niveau représente deux bits.

La paire de valeur (ou doublet) constitue ce que l'on appelle un symbole.

Les différents possibilités sont : 00, 01, 11, 10, la somme des deux sous trains de symboles donne un mot de quatre bits à 16 possibilités.

À la réception il faut récupérer les symboles émis sans erreurs.

III.2.1 Emetteur

La figure III.2 représente le schéma bloc de la partie émettrice de la chaîne de transmission, il est composé des éléments suivants :

- ❖ Une source génère un train de nombres entiers de 0 à 15.
- ❖ Un codeur I/Q divise le train de nombres en deux sous trains composés de nombres entiers entre 0 et 3.
- ❖ Un oscillateur capable de générer un signal sinusoïde à une fréquence de 120 MHz (la porteuse).
- ❖ Un déphaseur $\pi/2$
- ❖ Deux multiplieurs des signaux à deux entrées.
- ❖ Un sommateur des signaux à deux entrées.

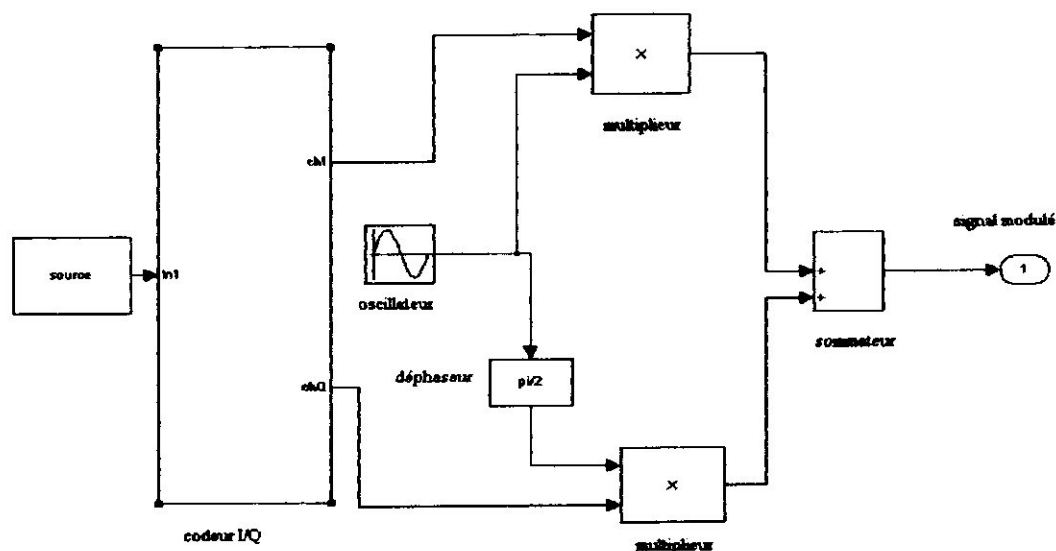


Figure III.2 : schéma bloc de la partie émettrice de la chaîne de transmission numérique

La transmission s'effectue par modulation d'amplitude sur deux porteuses de même fréquence mais déphasées de $\pi/2$.

La première porteuse est appelée porteuse en phase, la seconde est dite en quadrature.

Les deux signaux générés I (in-phase) et Q (quadrature) sont modulés par les deux sinusoïdes. Le signal à injecter dans le canal de transmission est la somme des signaux modulés I et Q.

III.2.2 Récepteur

La figure III.3 représente le schéma bloc du récepteur de la chaîne de transmission, il est composé des éléments suivants :

- ❖ Un oscillateur capable de générer un signal sinusoïde à une fréquence de 120 MHz.
- ❖ Un déphaseur $\pi/2$
- ❖ Deux multiplieurs des signaux à deux entrées.
- ❖ Deux filtres passe bas de fréquence de coupure $f_c=3 \cdot f_s$, où f_s est la fréquence symbole.
- ❖ Un décodeur I/Q, son rôle est de former un train de nombres à partir des deux trains I et Q.

Le signal reçu est multiplié par deux signaux sinusoïdaux déphasés de $\pi/2$. Afin d'ôter la fréquence porteuse, ce signal est filtré pour extraire les signaux I et Q.

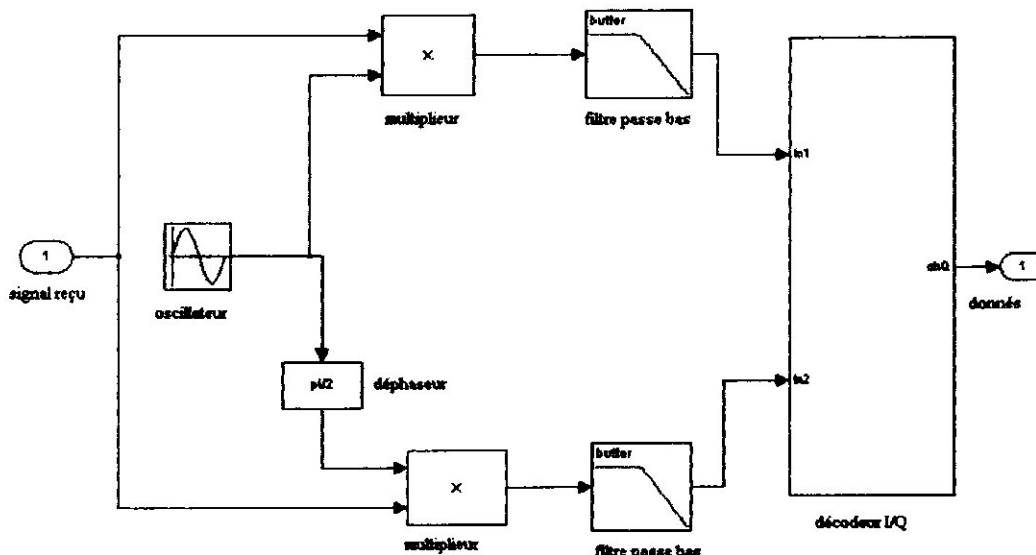


Figure III.3 : schéma bloc du récepteur de la chaîne de transmission numérique

III.3 Gamme de fréquences

La technique de modulation QAM16 est utilisée dans plusieurs applications de télécommunication. En aéronautique, la communication peut être vocale ou par liaison des données (data link). Différents systèmes de communication sont utilisés, on peut citer :

- ❖ Communication vocale VHF.
- ❖ VDL, La liaison numérique VHF (VHF digital link).
- ❖ SSR mode S, Radar secondaire de surveillance.
- ❖ Communication par satellite SATCOM.

Le tableau III.1 représente la bande de fréquence de chaque système.

Système	Bande de fréquence (porteuse)	information
Communication vocale	118-137 MHz	
VDL	118-137 MHz	Débit=42 Kbps
SSR mode S	1030 MHz émission 1090 MHz réponse	Message court 54 bits Message long 112 bits
SATCOM	27-40 GHz	

Tableau III.1 : gamme de fréquence aéronautique

III.4 Description du travail sous MATLAB

Dans cette partie on va assembler les différents blocs qui forment la chaîne de transmission de façon similaire à la réalisation pratique.

On va simuler chaque partie de la chaîne (émetteur, canal, récepteur) à part, et ensuite on fait l'assemblage des trois parties.

On va simuler la technique de modulation QAM 16 dans un réseau VDL avec une fréquence porteuse $f_p=120$ MHz, et une fréquence symbole $f_s=10.5$ KHz. (fréquence à laquelle sont envoyés les canaux I et Q).

Pour éviter le problème de temps de simulation on va remplacer la fréquence 10.5 KHz par une fréquence assez élevée par exemple $f_s=5$ MHz.

III.4.1 Emetteur :

L'émetteur dans la simulation est constituée de trois étages ou bien modules. Le premier module c'est la source d'informations, le second s'attache à la formation des signaux I et Q, et le troisième module procède à la génération de deux porteuses en quadrature.

III.4.1.1 Source d'information :

Nous présentons la source comme un générateur des nombres aléatoire de zéros jusqu'à M-1. Dans notre cas de 0 à 15. Une boîte de dialogue de la source est montré à la figure III.4.

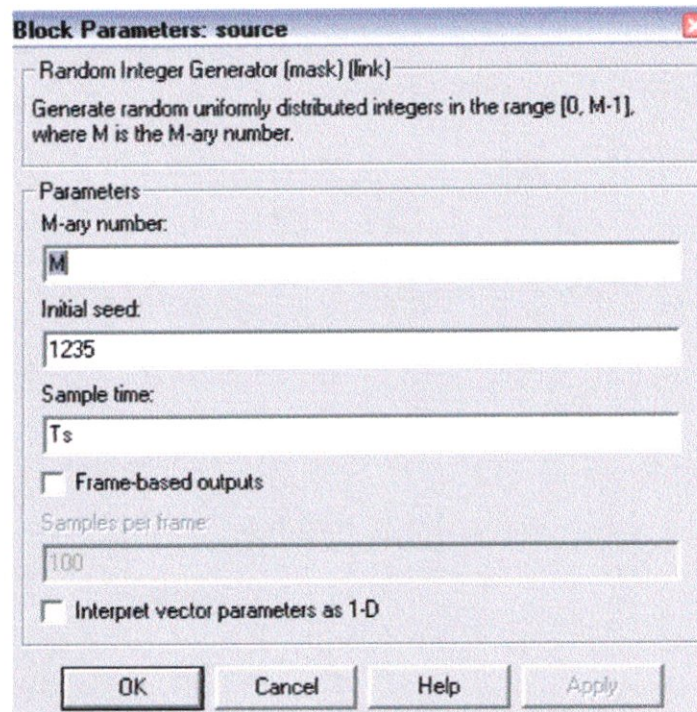


Figure III.4 : boîte de dialogue « source »

Le bloc de source est montré à la figure III.5.

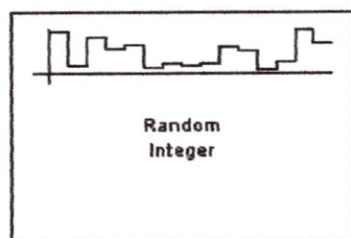


Figure III.5 : la source d'information

III.4.1.2 Codeur I/Q :

Pour créer les signaux I et Q, nous utilisons à la sortie de la source un convertisseur entier – binaire qui convertit chaque symbole issu de la source en quatre bits, puis nous faisons un démultiplexage de chaque symbole en deux symboles à deux bits.

Les paramètres de ce convertisseur sont montrés à la figure III.6 :

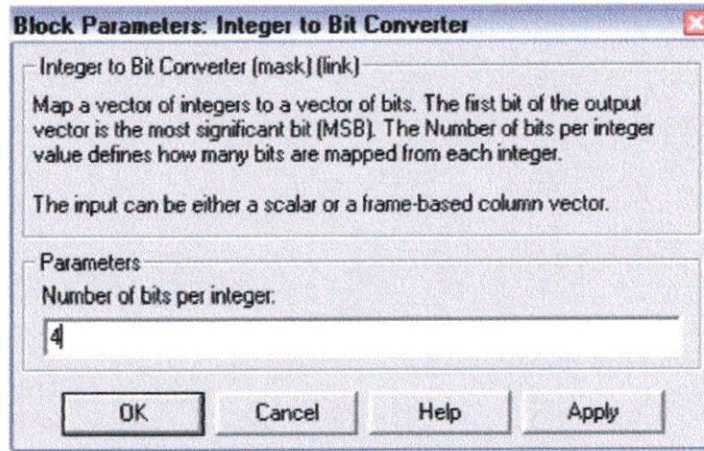


Figure III.6 : boîte de dialogue « convertisseur entier-bits »

Deux bits représentent un entier entre 0 et 3, pour cela nous avons utilisé un convertisseur binaire-entier à deux bits, soient les valeurs suivantes : 0, 1, 2, 3.

Les paramètres de ce convertisseur sont montrés à la figure III.7 :

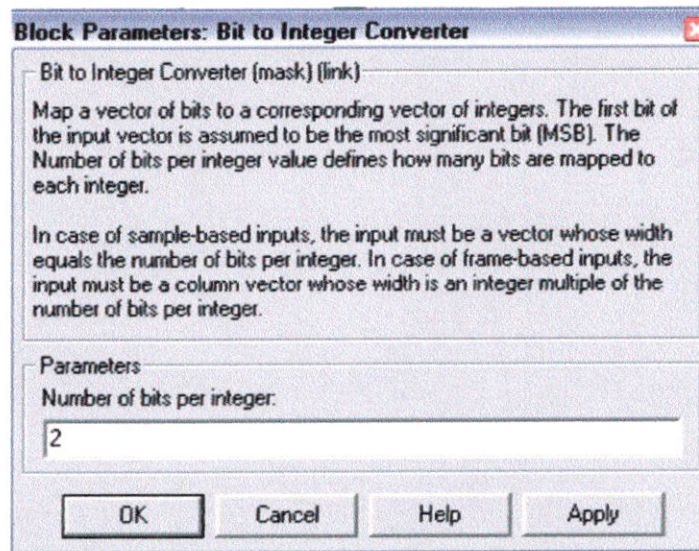
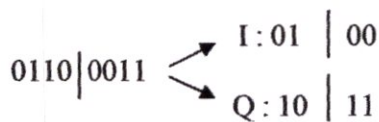


Figure III.7 : boîte de dialogue « convertisseur bits-entier »

Afin d'avoir des signaux de moyenne nulle, nous avons soustrait « 1.5 » de la valeur émise par le convertisseur binaire-entier. En sortie les signaux I et Q ainsi générés délivrent des valeurs respectives : -1.5, -0.5, 0.5, 1.5.

Remarque :

Le démultiplexage s'effectue selon l'exemple donné ci-dessous :



Le schéma de ce codeur est montré à la figure III.8.

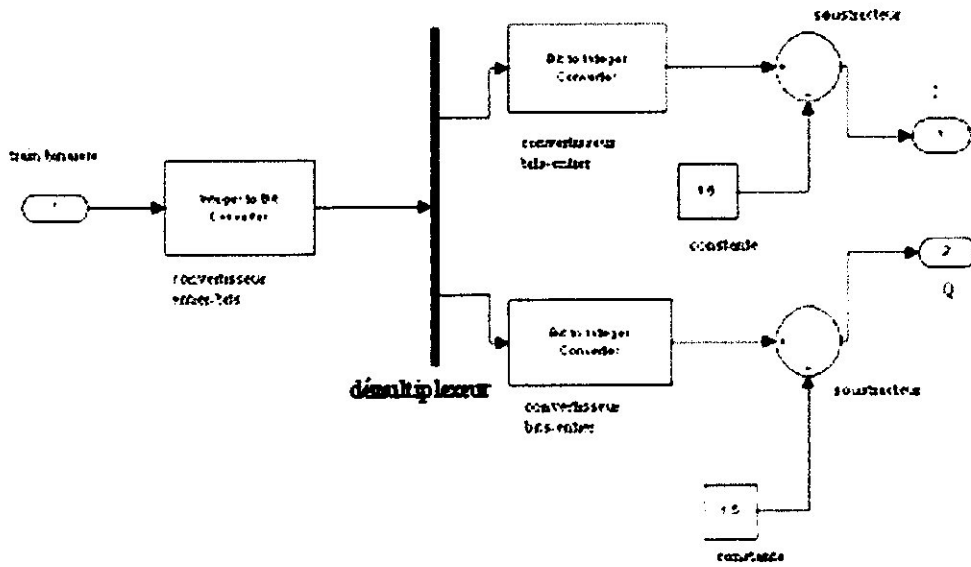


Figure III.8 : codeur I/Q

III.4.1.3 Modulateur des signaux I et Q :

Les signaux I et Q générés sont ensuite modulés par deux sinusoïdes en quadrature (déphasage de $\pi/2$) de fréquence $f_p = 120\text{MHz}$ (fréquence de la porteuse). Ces deux signaux modulés sont additionnés afin d'être transmis (figure III.9).

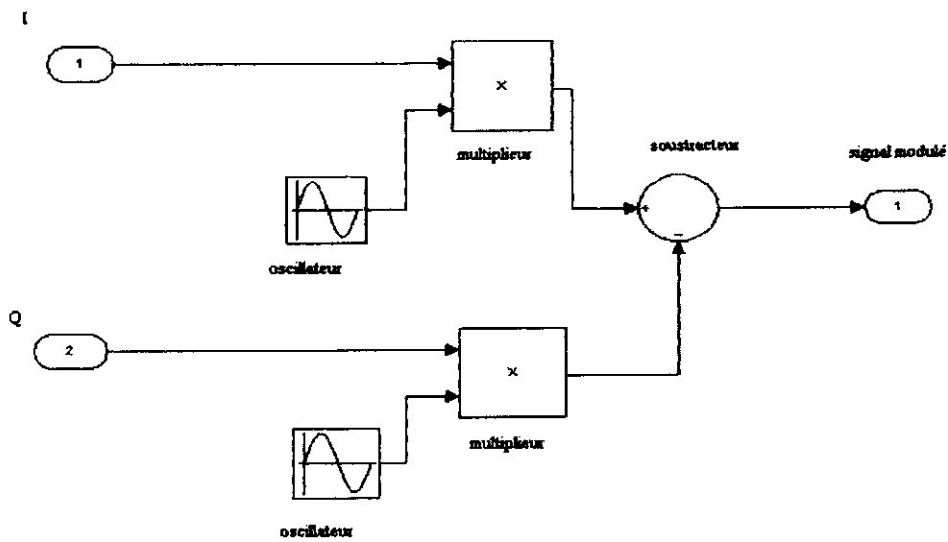


Figure III.9 : Modulateur des signaux I et Q

Nous avons choisi la fréquence porteuse et les autres paramètres de la boîte de dialogue de sine wave comme indiqués à la figure III.10 :

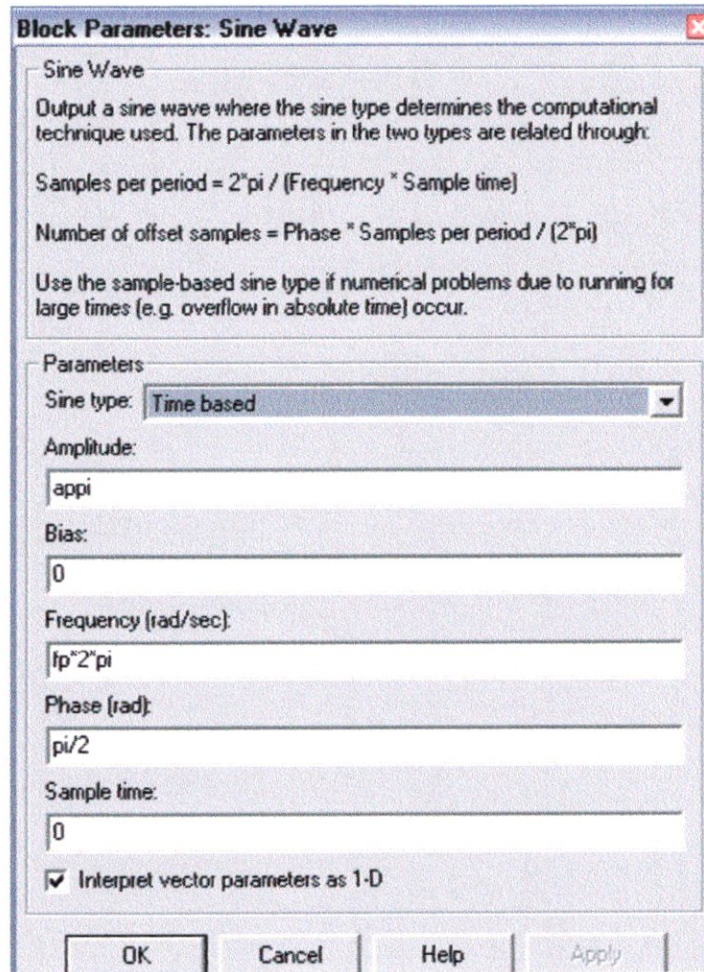


Figure III.10 : boîte de dialogue du bloc « sine wave »

III.4.2 Le canal de transmission :

Un canal peut être l'espace libre ou un câble. Chaque canal caractérisé par une largeur de bande B , dépend essentiellement de la fréquence du signal utile et du spectre de signal modulé, dans notre cas f_s , ou $B=2f_s$.

On a $f_s=5$ MHz $\rightarrow B=2*5=10$ MHz.

Les spectres des signaux utiles, porteuse, et modulé sont montrées aux les figures III.11, III.12, et III.13 :

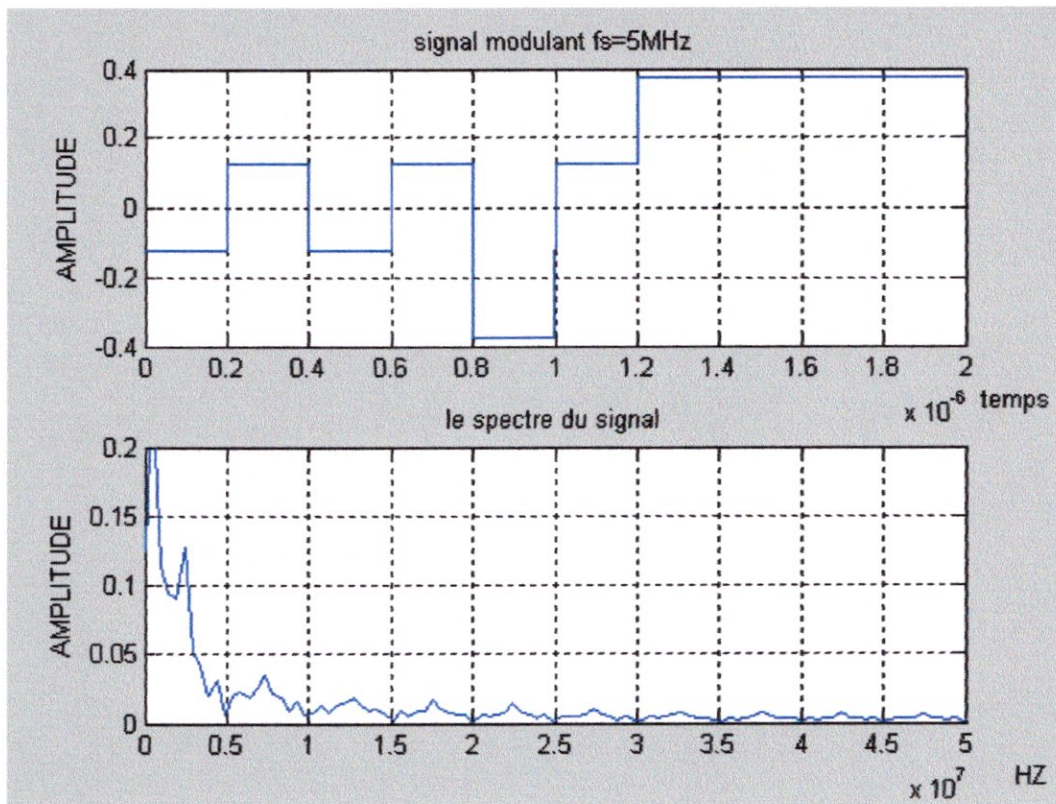


Figure III.11 : le spectre du signal utile

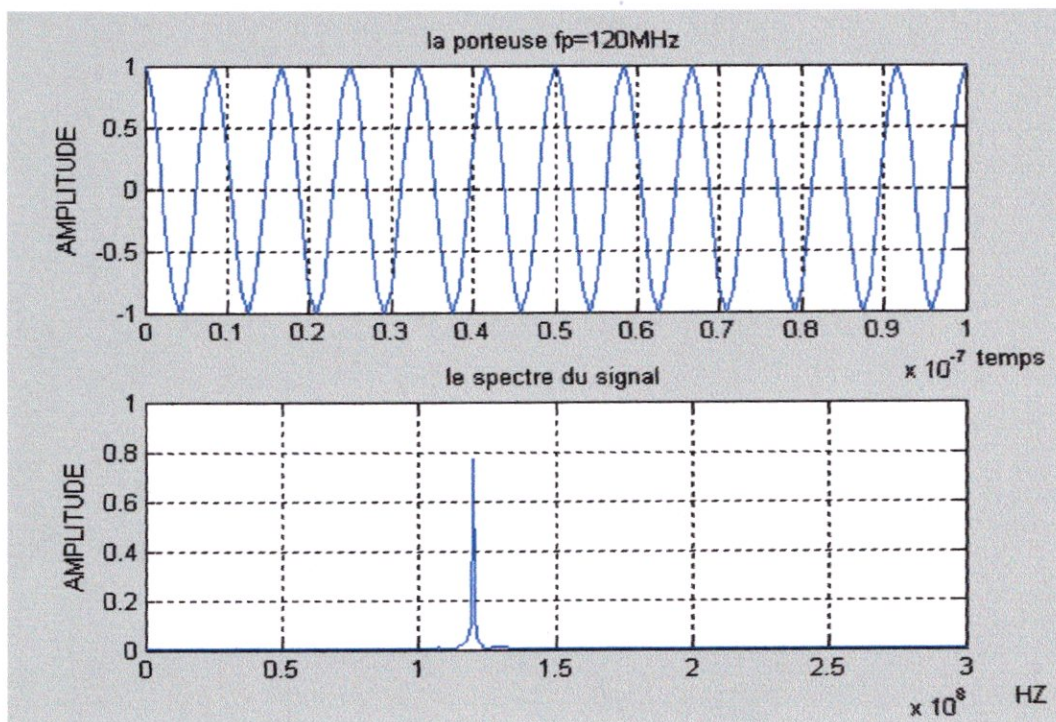


Figure III.12 : le spectre de la porteuse

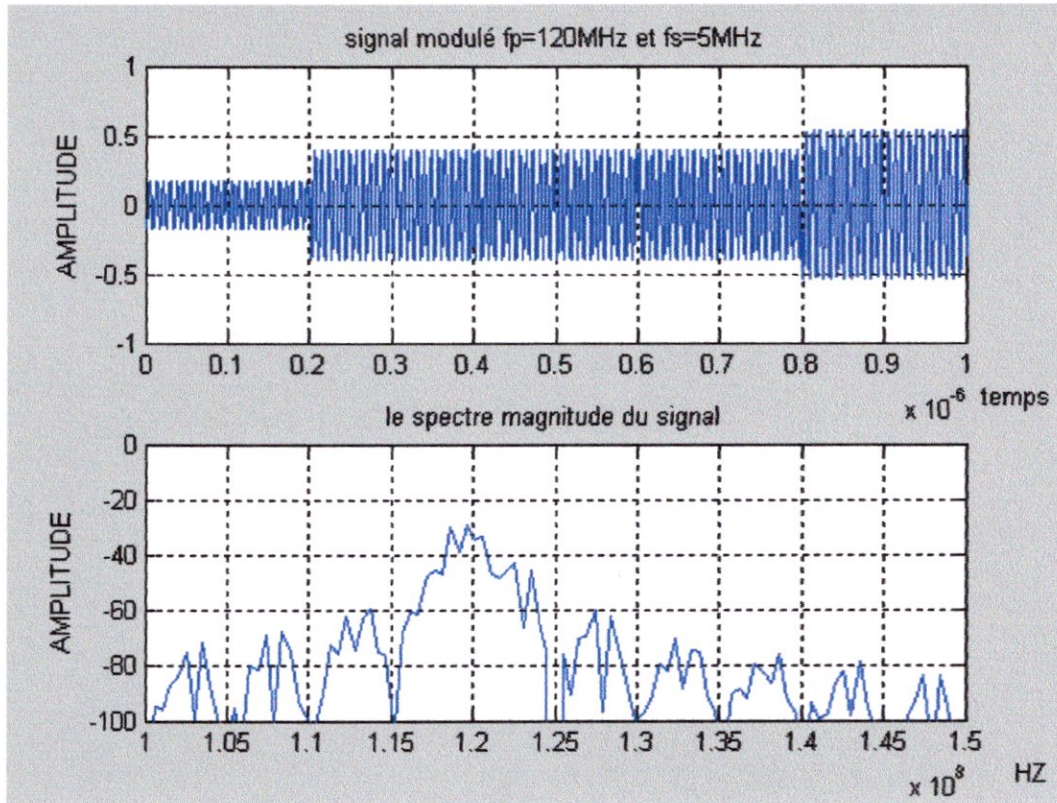


Figure III.13 : le spectre du signal modulé

Dans ce projet nous avons utilisé un canal classique (idéal) avec un bruit gaussien de type AWGN (additive-white-Gaussian-noise). Ce canal fonctionne avec deux blocs bloqueurs l'un à l'entrée et l'autre à la sortie du canal. Nous avons vu les caractéristiques de ce canal dans le chapitre II. Les paramètres du canal sont indiqués à la figure III.14.

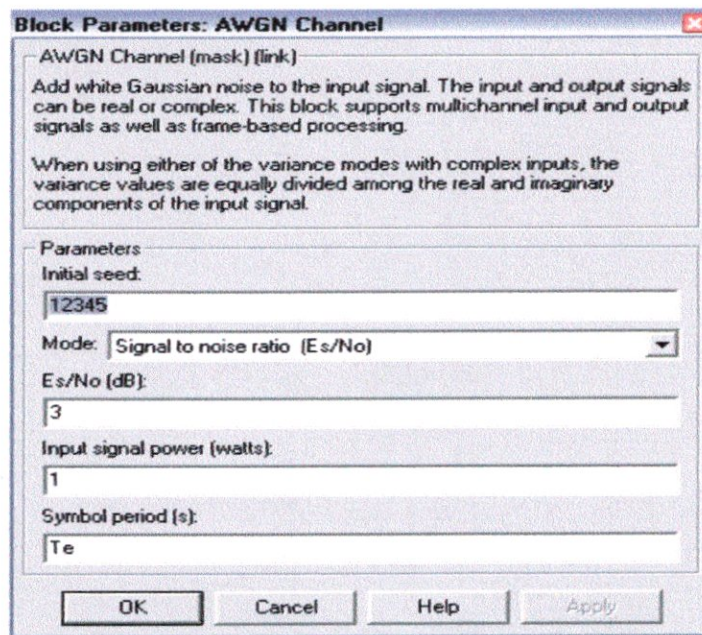


Figure III.14 : boîte de dialogue de « canal AWGN »

Nous avons choisi parmi les trois modes le mode Es/N0.

L'augmentation de la puissance du signal émis améliore la qualité du signal reçu.

III.4.3 Récepteur :

Afin de récupérer les données envoyées à partir du signal reçu, il faut faire l'opération inverse de la modulation, à savoir multiplier le signal reçu par deux signaux sinusoïdaux en quadrature.

III.4.3.1 Récupération de la fréquence porteuse

Le récepteur ne dispose pas de la fréquence porteuse de l'émetteur. Nous devons la recréer en phase avec la fréquence qui a servi à la modulation afin de pouvoir extraire les signaux I et Q. Cette opération est obtenue grâce à une boucle à verrouillage de phase, une PLL réglé à la fréquence porteuse. Nous avons réglé la PLL à partir d'une boîte de dialogue de bloc PLL comme indiquée à la figure III.15 :

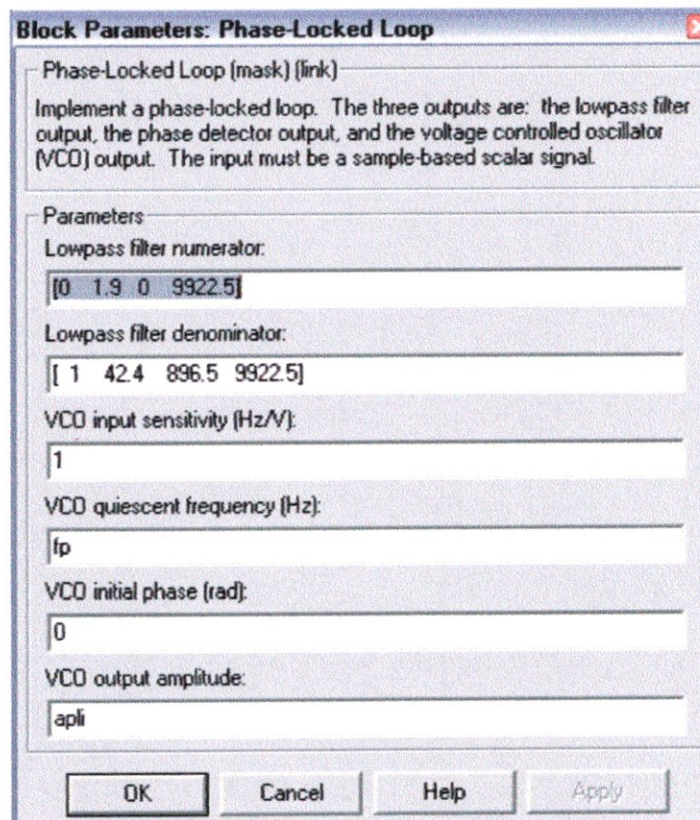


Figure III.15 : Boîte de dialogue du bloc PLL

Une PLL comprend les éléments suivants :

- ❖ Un comparateur de phase.
- ❖ Un filtre passe-bas de fréquence de coupure $f_c = 10\text{Hz}$.
- ❖ Un VCO (Voltage Controlled Oscillator) est, comme son nom le suggère, un oscillateur commandé par une tension de fréquence d'oscillation libre f_0 égale à la fréquence de la porteuse $f_0 = f_p$.

Les différents éléments sont organisés comme la montre la figure III.16 :

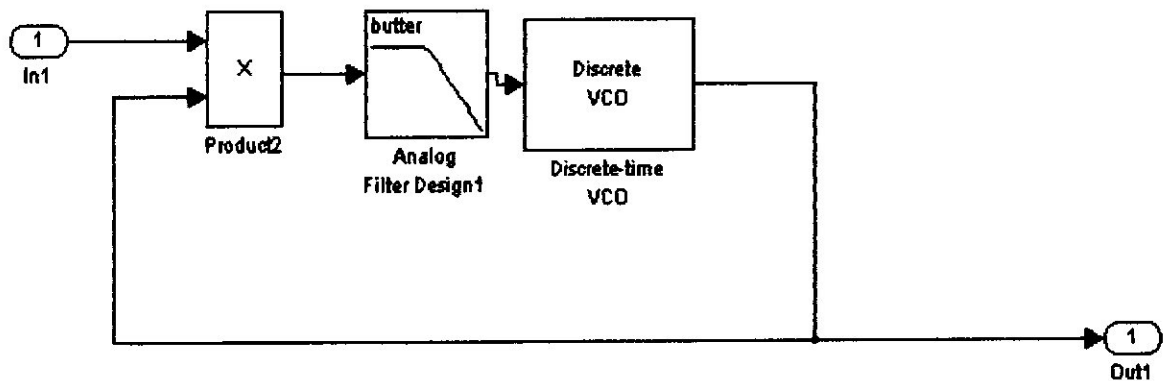


Figure III.16 : structure d'une PLL

Les deux PLL sont déphasés de $\pi/2$ afin d'obtenir à la sortie deux sinusoïdes $\cos(\omega_0 t)$ et $-\sin(\omega_0 t)$. Afin d'éviter le message des erreurs on va mettre des terminateurs aux sorties non utilisées, comme indiqué à la figure III.17.

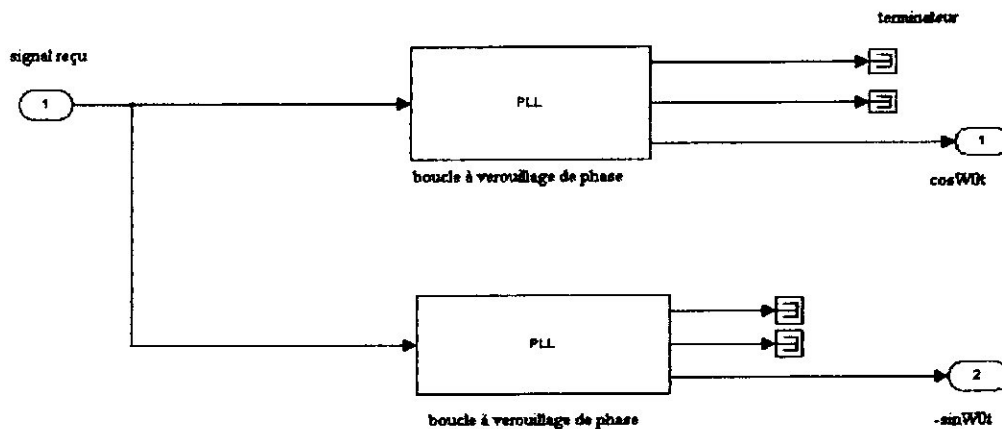


Figure III.17 : récupération la fréquence porteuse

III.4.3.2 Démodulation du signal reçu

La démodulation consiste à multiplier le signal reçu par les signaux en quadrature disponibles en sortie des PLL. Ce signal démodulé est ensuite filtré à nouveau pour récupérer les informations en bande de base. Nous avons choisi comme filtre passe bas un filtre de Butterworth d'ordre 8 ayant pour fréquence de coupure 15 MHz. La boîte de dialogue de la filtre est montré à la figure III.18.

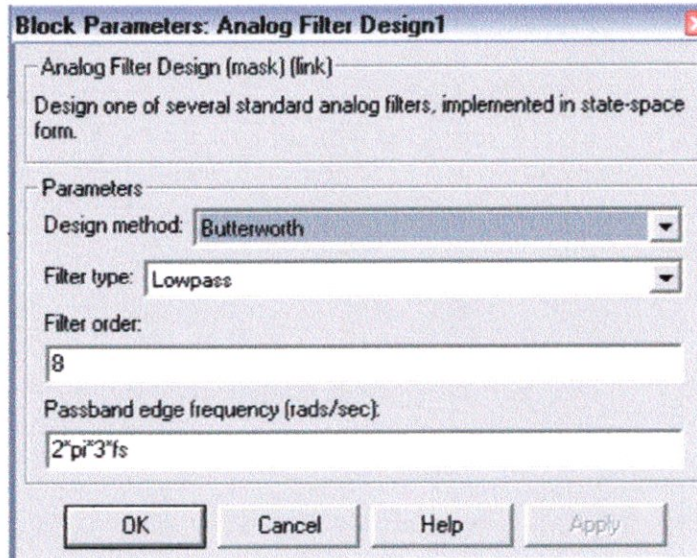


Figure III.18 : boîte de dialogue du bloc « analog filter design »

Il convient en effet de prendre une fréquence de coupure relativement haute (par rapport à l'information située à 5 MHz) pour conserver le maximum d'harmoniques afin de reconstituer un signal le plus carré possible. Après le filtrage on applique un bloc de saturation afin de limiter les extrémités du signal récupéré dans l'intervalle $[-1.5, 1.5]$ pour obtenir les mêmes signaux émis. Nous avons donc à ce moment les estimations de I et de Q figure III.19.

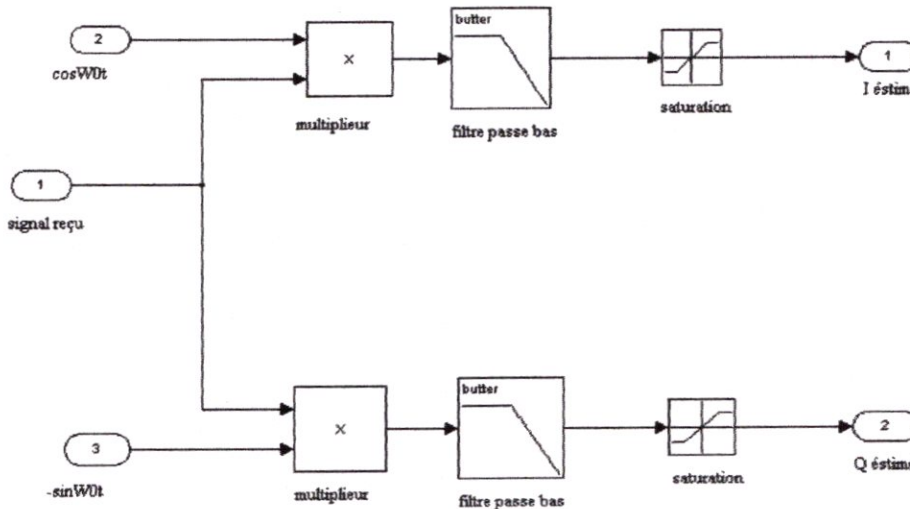


Figure III.19 : démodulation

III.4.3.3 Récupération de la fréquence symbole

Avant de reconstruire les signaux I et Q, nous avons besoin de retrouver la fréquence symbole (5 MHz) qui est la fréquence à laquelle sont envoyés les signaux I et Q. Pour cette opération, nous avons utilisé une autre boucle à verrouillage de phase PLL, réglée pour fonctionner à une fréquence

égale à la fréquence symbole f_s . A la sortie de PLL on obtient une sinusoïde. Pour transformer cette sinusoïde en signal carré, on met une fonction "signe" de Matlab pour pouvoir commander un élément échantillonneur-bloqueur. On a le schéma de cet étage à la figure III.20.

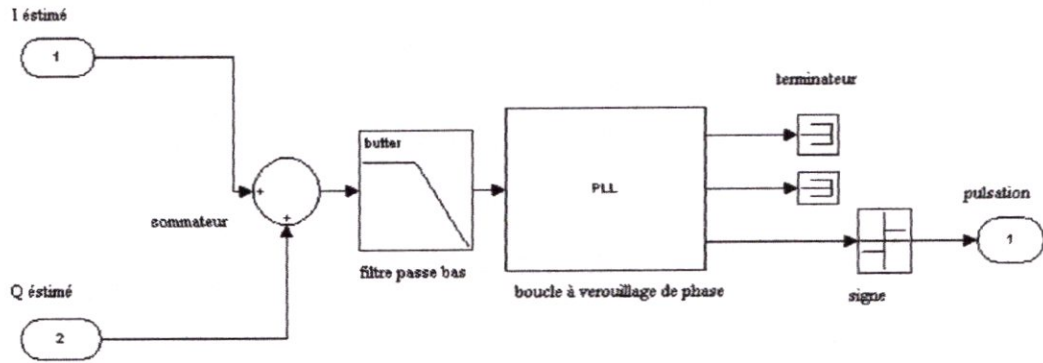


Figure III.20 : récupération de la fréquence symbole

III.4.3.4 Reconstruction des signaux I et Q

En effet, le front montant du signal carré issu de la récupération de la fréquence symbole commande l'échantillonneur-bloqueur, (la boîte de dialogue est montré à la figure III.21) qui garde la valeur des signaux I et Q le temps d'une période T_s .

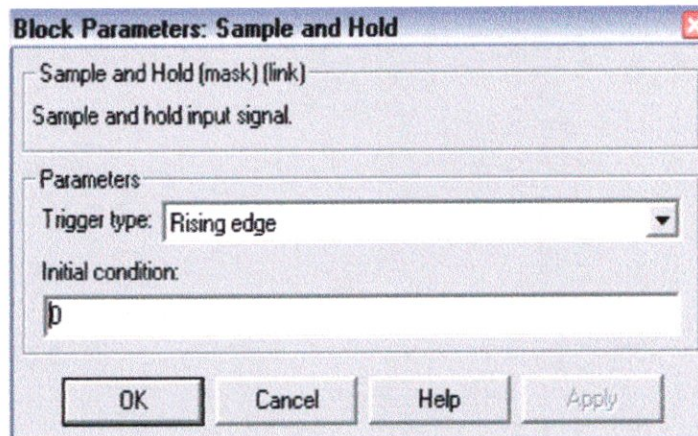


Figure III.21 : boîte de dialogue du bloc « sample and hold »

Après avoir rajouté la valeur constante « 1.5 », nous sommes capable, via le quantificateur, de "décider" si le symbole reçu est à « 0 » ; « 1 » ; « 2 » ; ou « 3 ». On prend un pas de quantification de « 1 ». La boîte de dialogue de quantificateur est montré à la figure III.22.

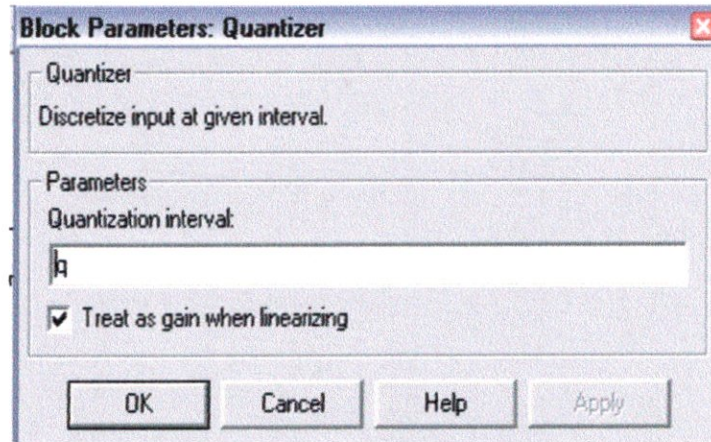


Figure III.22 : boîte de dialogue du bloc « quantizer »

Après quantification nous retrouvons exactement les signaux I et Q générés à la première étape. Ces étapes sont montrées à la figure III.23 :

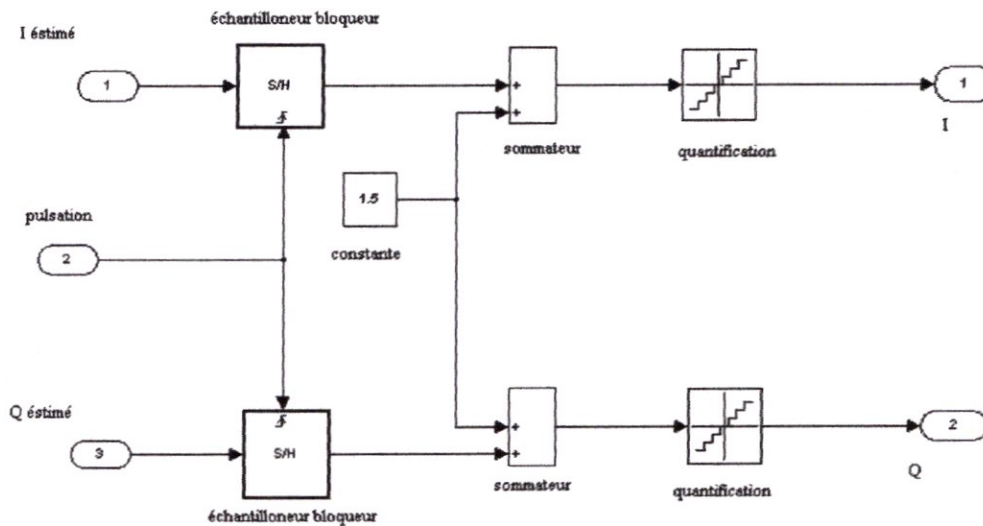


Figure III.23 : Reconstruction des signaux I et Q

III.4.3.5 Décodeur I/Q :

La fonction de cet étage est l'inverse de l'étage codeur I/Q.

Après avoir les signaux I et Q nous utilisons deux convertisseurs entier-binaire qui convertissent chaque symbole de 0 à 3 en code binaire à deux bits. Puis nous faisons un multiplexage afin d'obtenir un train binaire.

Comme nous avons vu au début, la source donne des symboles de 0 à 15. Pour ce faire nous avons utilisé un convertisseur bits-entier à quatre bits qui va convertir le train binaire en train de symboles de 0 à 15. Nous avons donc le signal de la source. La figure III.24 représente le décodeur I/Q.

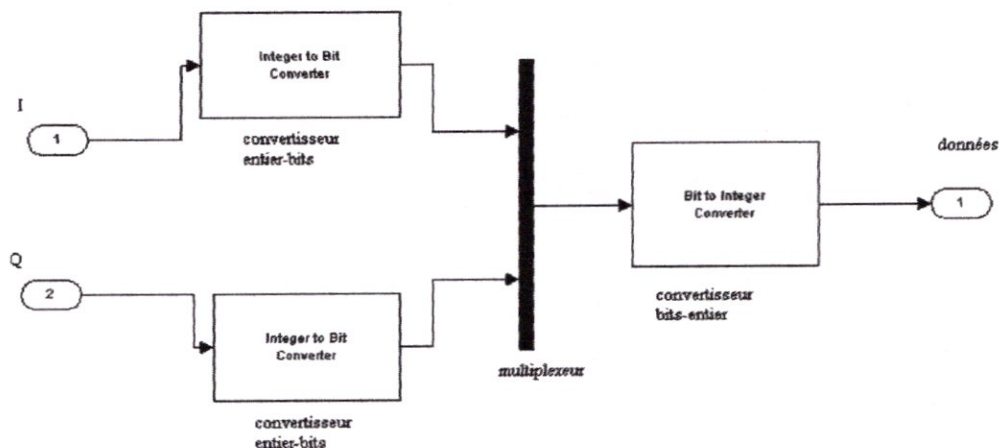


Figure III.24 : décodeur I/Q

III.4.4 Visualisation

En fait, il existe deux façons de visualiser le système.

Pour visualiser les signaux dans les différents étages nous utilisons un oscilloscope.

Pour comparer les résultats numériquement on utilise la fonction Matlab « to workspace ». Sa fonction est de transformer les signaux en forme d'un vecteur dont on peut traiter comme une variable à partir du workspace-Matlab. Les caractéristiques de cette fonction sont données à la figure III.25 :

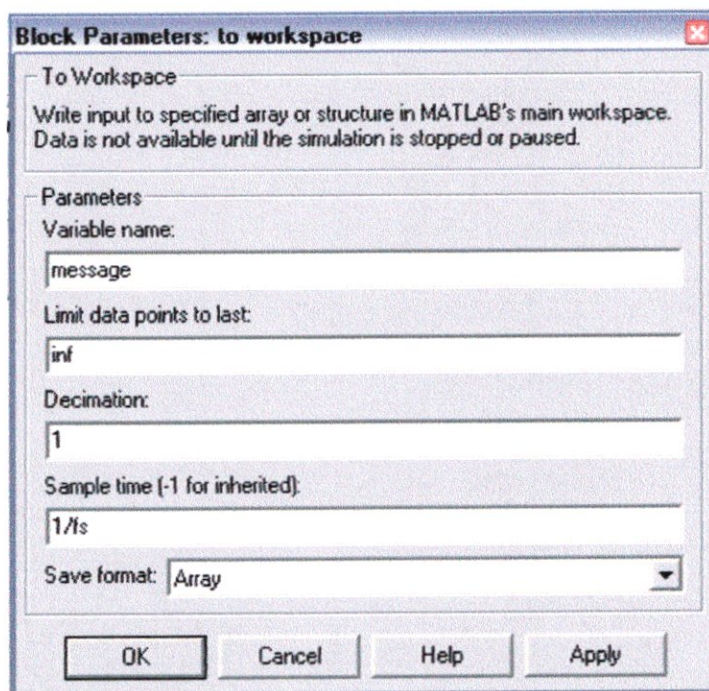


Figure III.25 : boîte de dialogue du bloc « to workspace »

III.4.5 Calculateur d'erreurs :

Dans la transmission numérique, des erreurs peuvent se produire à cause du bruit du canal de transmission, ou du mauvais filtrage à la réception ...etc. Afin de calculer le nombre d'erreurs, nous avons utilisé un calculateur qui compare les symboles à l'émission avec les symboles à la réception et donne : le nombre total des symboles émis, le nombre d'erreurs, et le taux d'erreur par symbole.

Le calculateur est montré à la figure III.26.

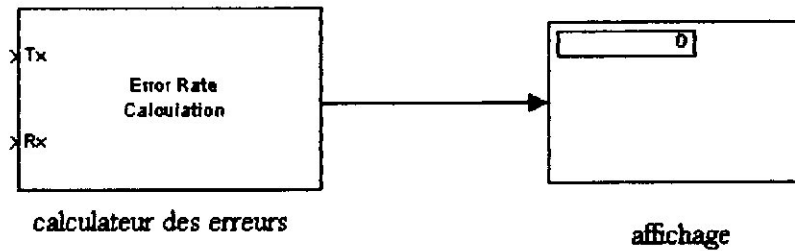


Figure III.26 : calculateur des erreurs

III.4.6 Modèle complet de la chaîne de transmission

Le signal reçu est décalé par rapport au signal émis. Pour synchroniser les deux signaux avant la comparaison nous utilisons un bloc retard au niveau de l'entrée du signal émis dans le calculateur des erreurs.

Compte tenu que le calculateur ne fonctionne pas que avec les signaux discrétisés, nous avons mis deux blocs (retard et échantillonneur) aux deux entrées du calculateur des erreurs.

Les deux blocs (retard et échantillonneur) sont montrés à la figure III.27 :

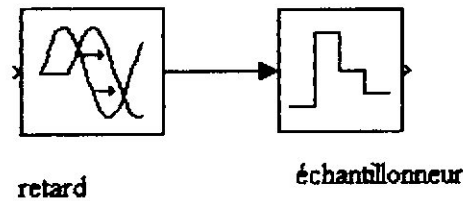


Figure III.27 : retard et échantillonneur

Nous avons complété notre modèle de la chaîne de transmission numérique et le schéma global est montré à la figure III.28.

III.5 Conclusion

Nous avons le modèle complet d'une chaîne de transmission numérique utilisant une modulation de type QAM16. Nous pouvons vérifier les résultats graphiquement à l'aide des oscilloscopes ou numériquement à l'aide de la fonction « to workspace » qui a pour nom « message ».

Si les résultats sont identiques, la chaîne fonctionne donc très bien en simulation.

Chapitre IV

Mode d'emploi de la simulation

Références [7]

IV.1 Introduction.	70
IV.2 L'ouverture du modèle	70
IV.3 Le démarrage de la simulation.	73
IV.4 Les visualisations.	74
IV.5 Conclusion	80

IV Mode d'emploi de la simulation

IV.1 Introduction

Une fois le modèle complet de la chaîne de transmission numérique est conçu, nous allons simuler et observer durant ce chapitre le comportement du système.

Avant d'entamer la simulation, il est nécessaire de saisir les valeurs des variables que nous allons utiliser dans le système. Pour cela, nous allons ouvrir un fichier Matlab (m-file) et écrire toutes les valeurs concernant la simulation comme un programme.

Quand nous exécutons le programme, les variables précédentes et présentes dans le workspace sont prêtes à l'utilisation quand nous démarrons la simulation.

IV.2 L'ouverture du modèle

La première étape pour ouvrir le modèle consiste à cliquer deux fois sur l'icône de Matlab se trouvant sur le bureau. Une fenêtre de commande de Matlab apparaît comme la montre la figure IV.1.

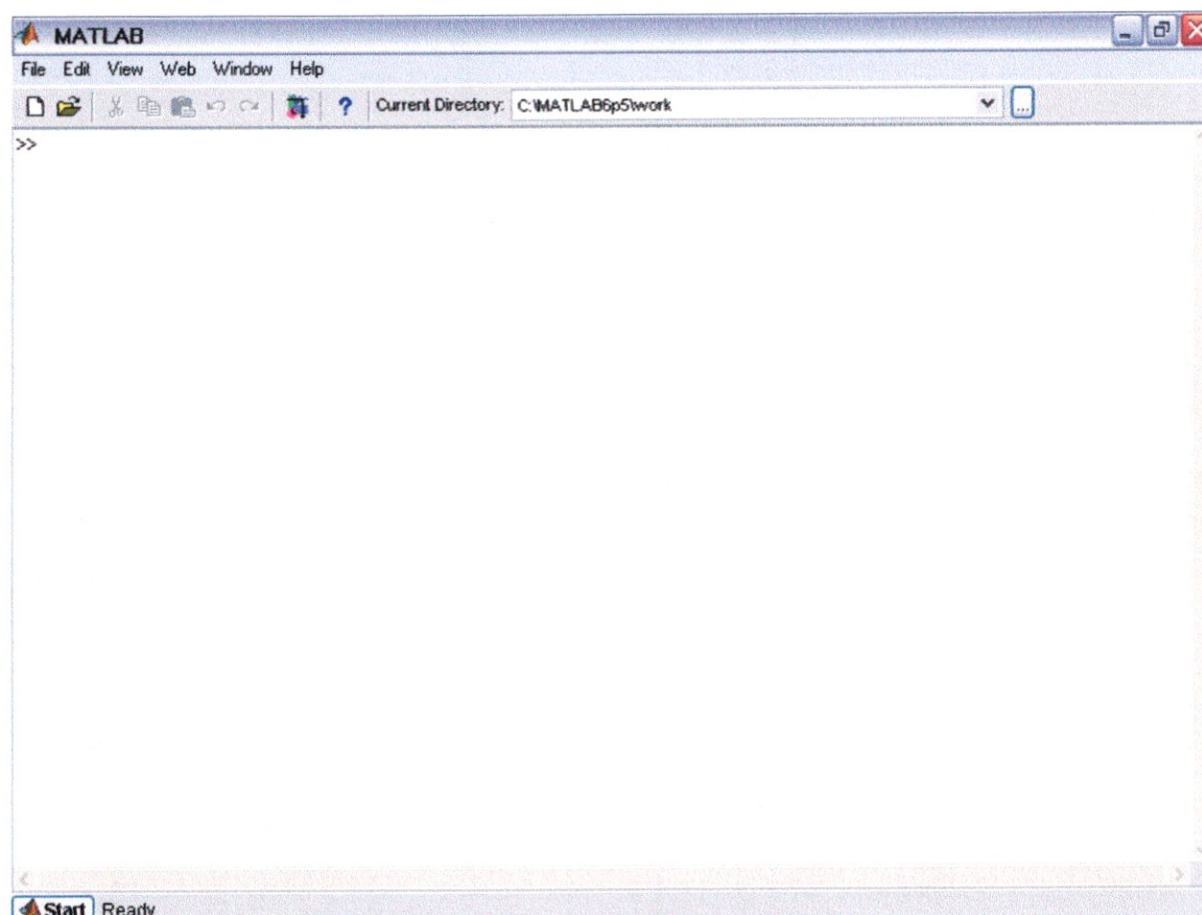


Figure IV.1 : la fenêtre principale du Matlab

La deuxième étape consiste à ouvrir l'outil Simulink à l'aide de l'icône de Simulink se trouvant sur la barre d'outils ou en utilisant la commande de Matlab « Simulink » comme le montre la figure IV.2.

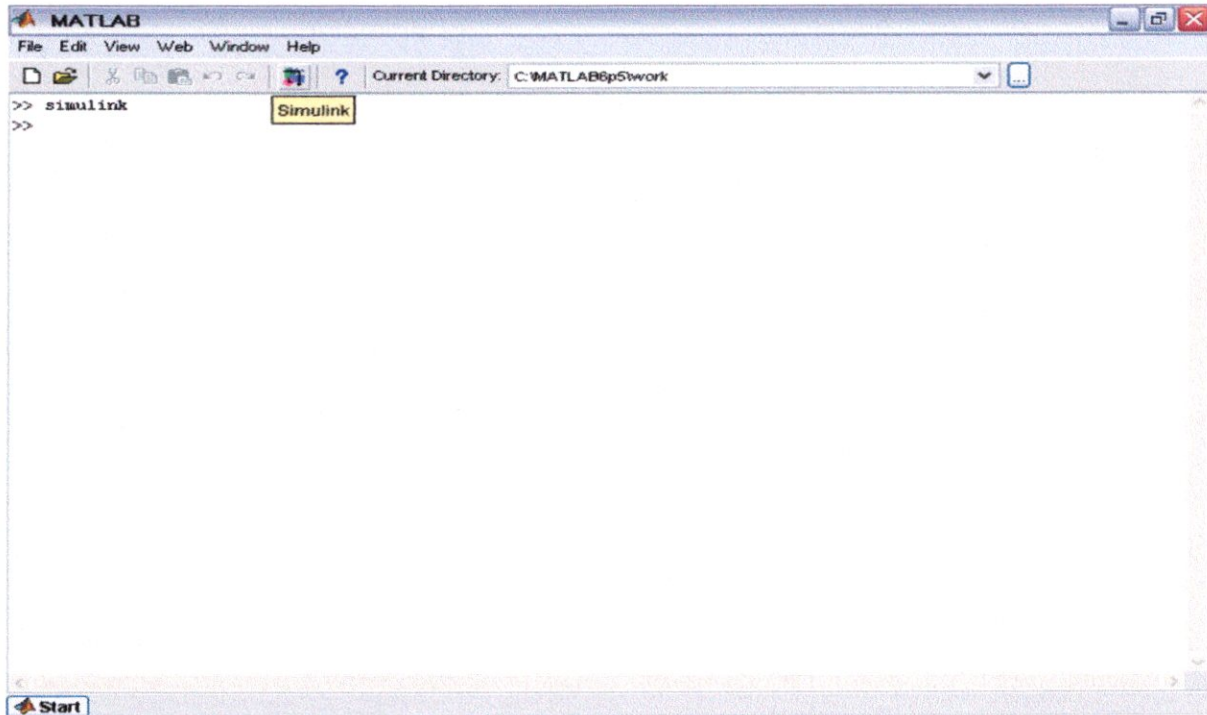


Figure IV.2 : l'icône du Simulink et la commande Simulink

Après cette étape, nous avons la fenêtre de Simulink montrée à la figure IV.3.

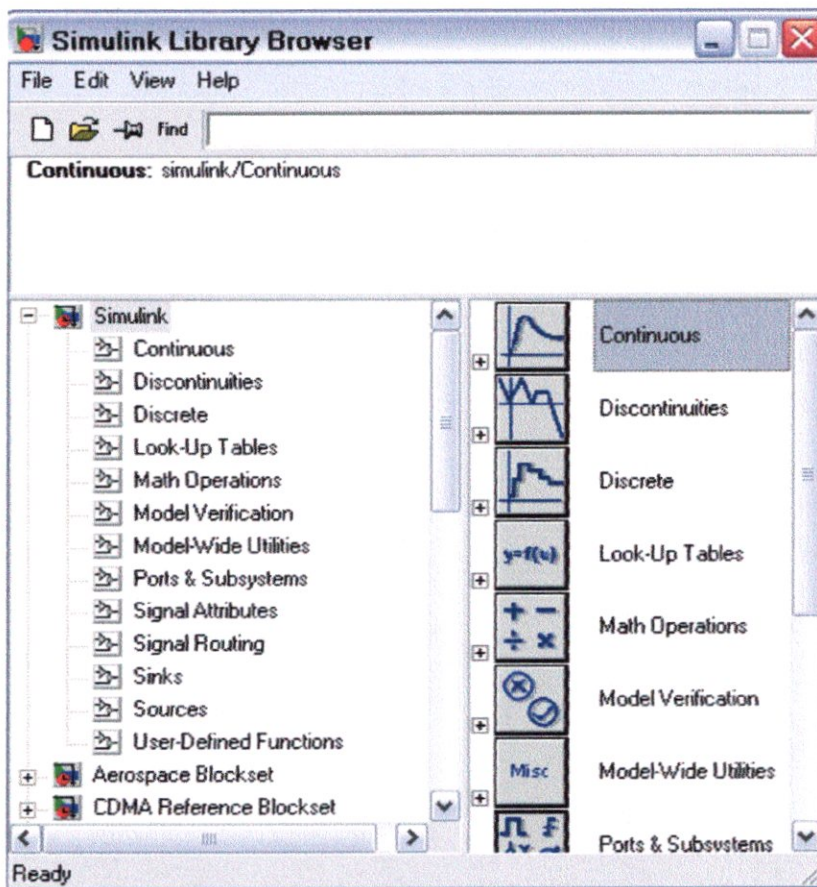


Figure IV.3 : la fenêtre du Simulink

Maintenant dans la fenêtre de Matlab nous cliquons sur le bouton « ouvrir » se trouvant sur la barre d'outils pour ouvrir le fichier de Matlab nommé « parametres » dans le répertoire « PFE ». La boîte de dialogue est montrée à la figure IV.4.

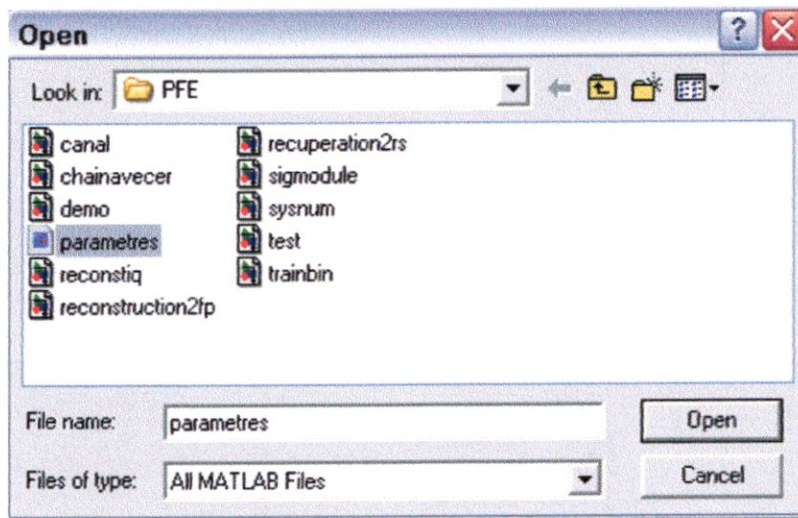


Figure IV.4 : la boîte de dialogue permettant d'ouvrir un fichier MATLAB

Le fichier comprend les instructions suivantes :

```
clear;
M=16;           % M aire
fp=120*1e6;    %la fréquence de la porteuse
fs=5*1e6;      %la fréquence de symbole
Ts=1/fs;
q=1;           %pas de quantification
cte=1.5;       %constante de codage
fe=1e10;       %fréquence d'échantillonnage
Te=1/fe;
fc=3*fs        %la fréquence de coupure du filtre dans le démodulateur
appi=2;        %amplitude de la porteuse en phase
appq=2;        %amplitude de la porteuse en quadrature
apli=1;        %amplitude vco en phase;
aplq=1;        %amplitude vco en quadrature
Tsim=10*Ts;    %temps de la simulation
```

Ce programme initialise les paramètres de la simulation et maintenant nous pouvons ouvrir le modèle nommé « gam » par le clic sur le bouton « ouvrir » se trouvant sur la fenêtre de Simulink. Le modèle va apparaître comme le montre la figure IV.5.

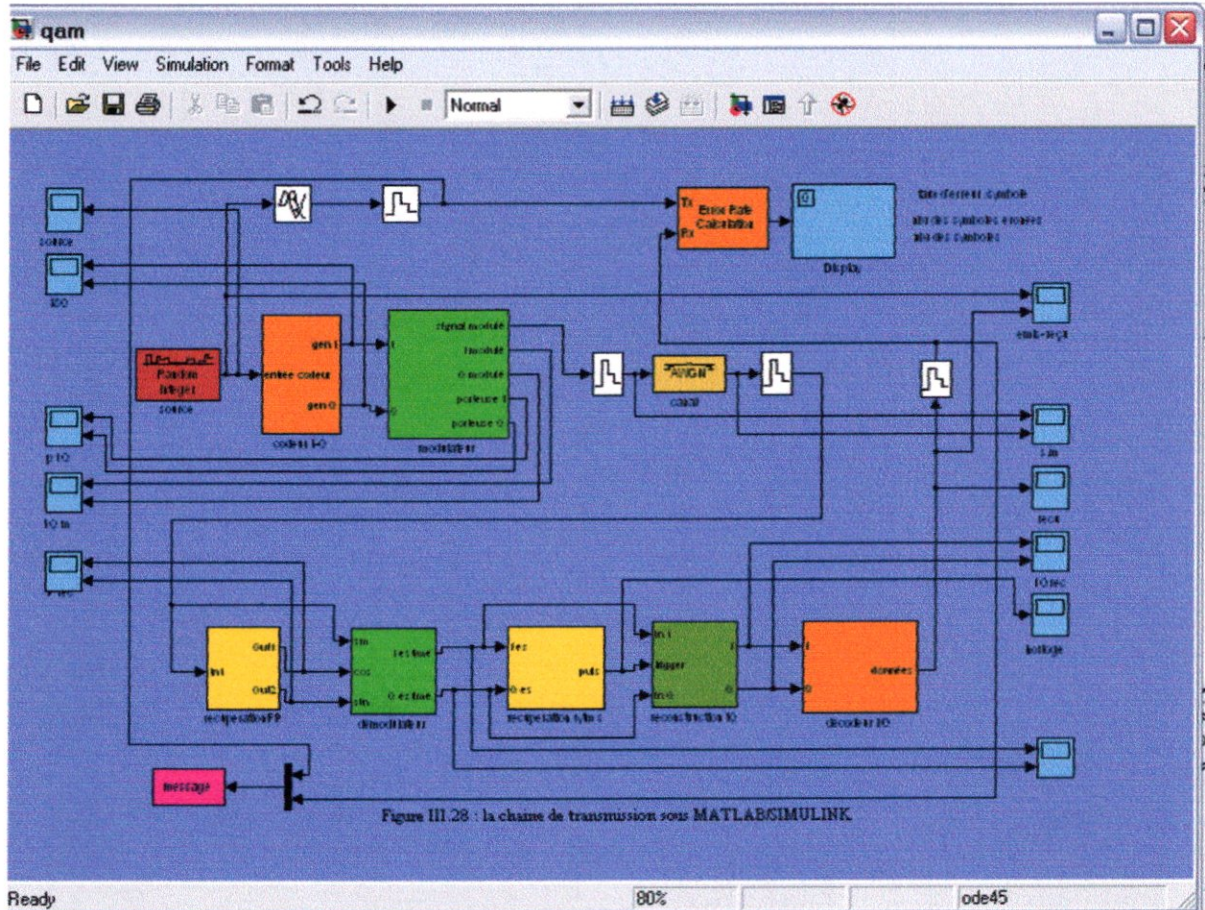


Figure III.28 : la chaîne de transmission sous MATLAB/SIMULINK.

Figure IV.5 : la fenêtre du modèle de la chaîne de transmission numérique

IV.3 Le démarrage de la simulation

Dans la fenêtre précédente, il faut cliquer sur le bouton « start simulation » comme indiqué à la figure IV.6

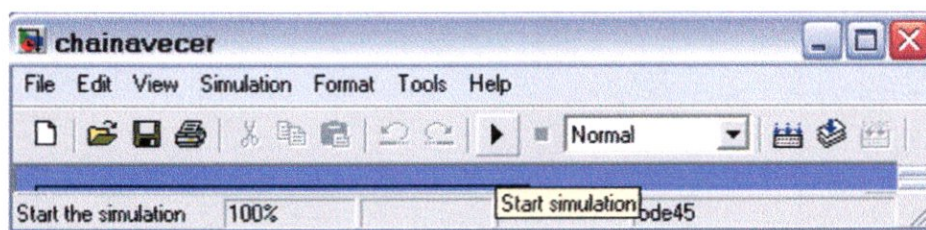



Figure IV.6 : démarrage la simulation

Pendant la simulation, nous avons dans le coin gauche en bas le mot anglais « running » (la simulation est en cours). Quand la simulation est finie, nous avons le mot « ready » (prête). Donc la simulation est terminée et nous pouvons observer les résultats.

IV.4 Les visualisations

Nous pouvons visualiser les signaux à n'importe quel point du modèle. Les visualisations des signaux sont effectuées à l'aide des oscilloscopes, en double cliquant sur l'icône de l'oscilloscope.

Nous pouvons avoir un agrandissement du signal en cliquant sur le bouton  (autoscale).

Nous allons visualiser les signaux de la source jusqu'à la destination.

Les figures de IV.7 à IV.17 représentent respectivement :

- ❖ le signal de la source (des entiers entre 0 et 15, chaque entier est de période T_s).
- ❖ les signaux I et Q sont obtenus sur les sorties du codeur I-Q
- ❖ les deux porteuses en quadrature de fréquence f_p .
- ❖ les signaux I et Q modulés sont obtenus en multipliant les signaux I et Q par les deux porteuses en quadrature.
- ❖ le signal modulé à l'entrée du canal et à la sortie du canal. Ce signal est la somme des signaux I et Q modulés.
- ❖ les deux porteuses sont récupérées au niveau du récepteur de fréquence f_p .
- ❖ les signaux I et Q à la sortie du démodulateur après l'élimination des porteuses.
- ❖ le signal de l'horloge a la fréquence symboles f_s .
- ❖ les signaux I et Q reconstitués.
- ❖ le signal reçu est la sortie du décodeur I-Q
- ❖ les deux signaux émis et reçus pour la comparaison.

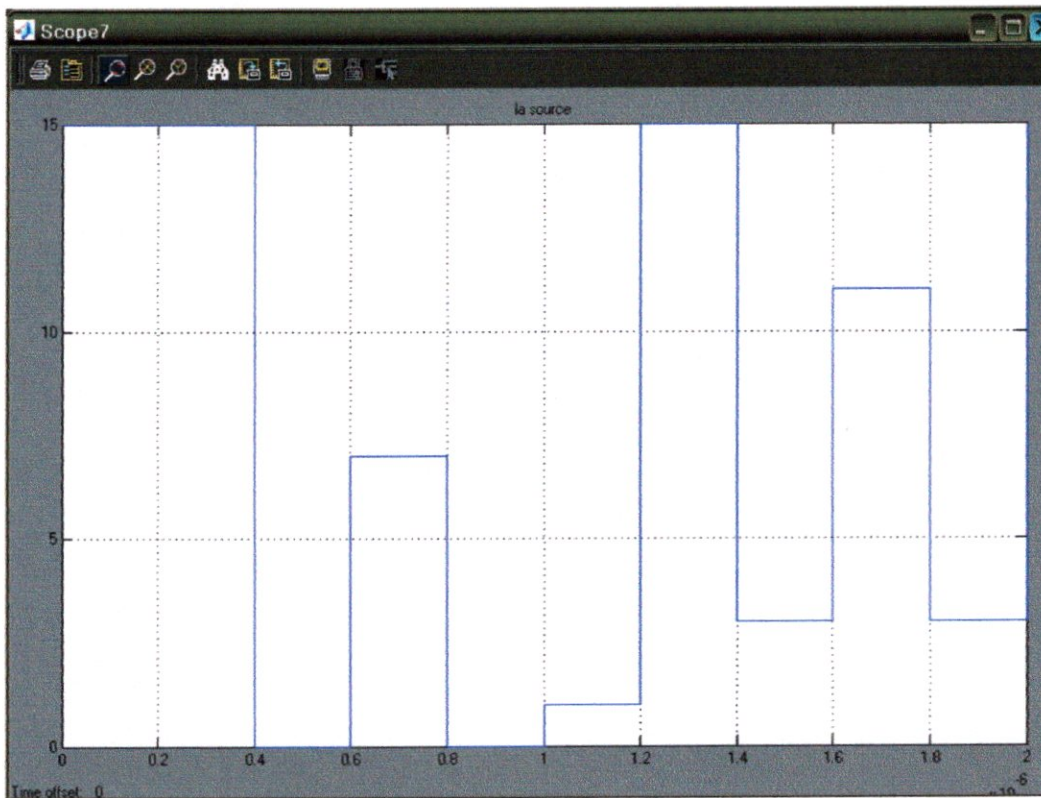


Figure IV.7 : le signal de la source

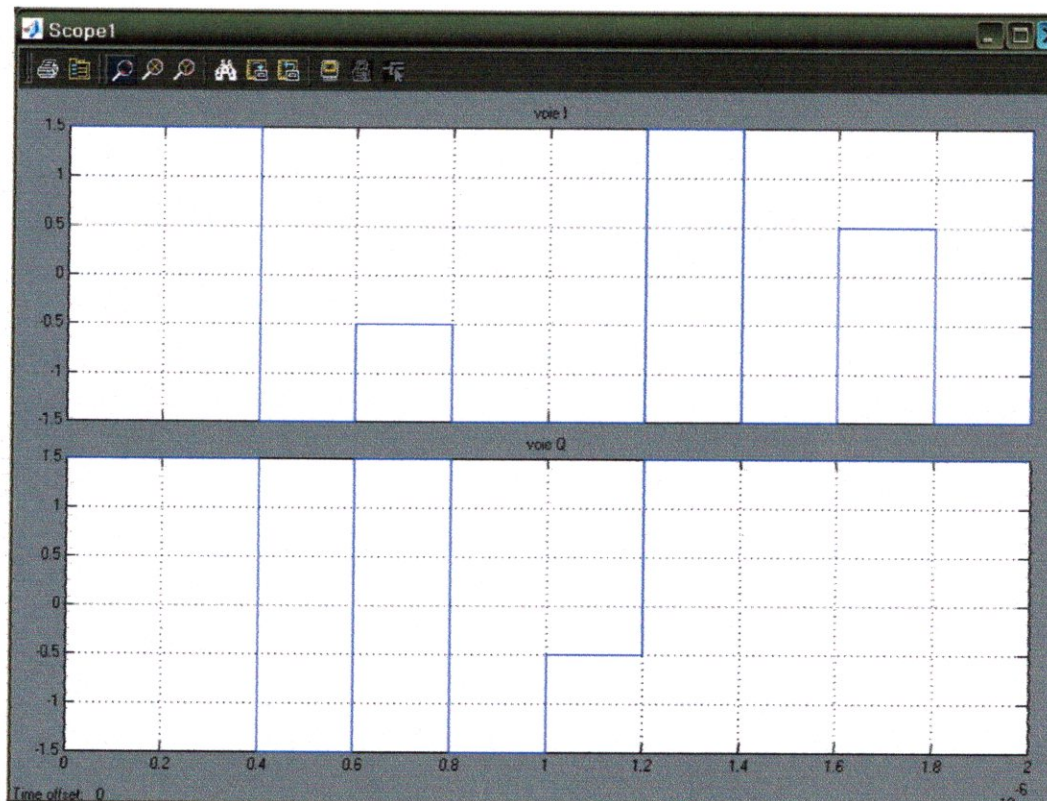


Figure IV.8 : les signaux I et Q générés

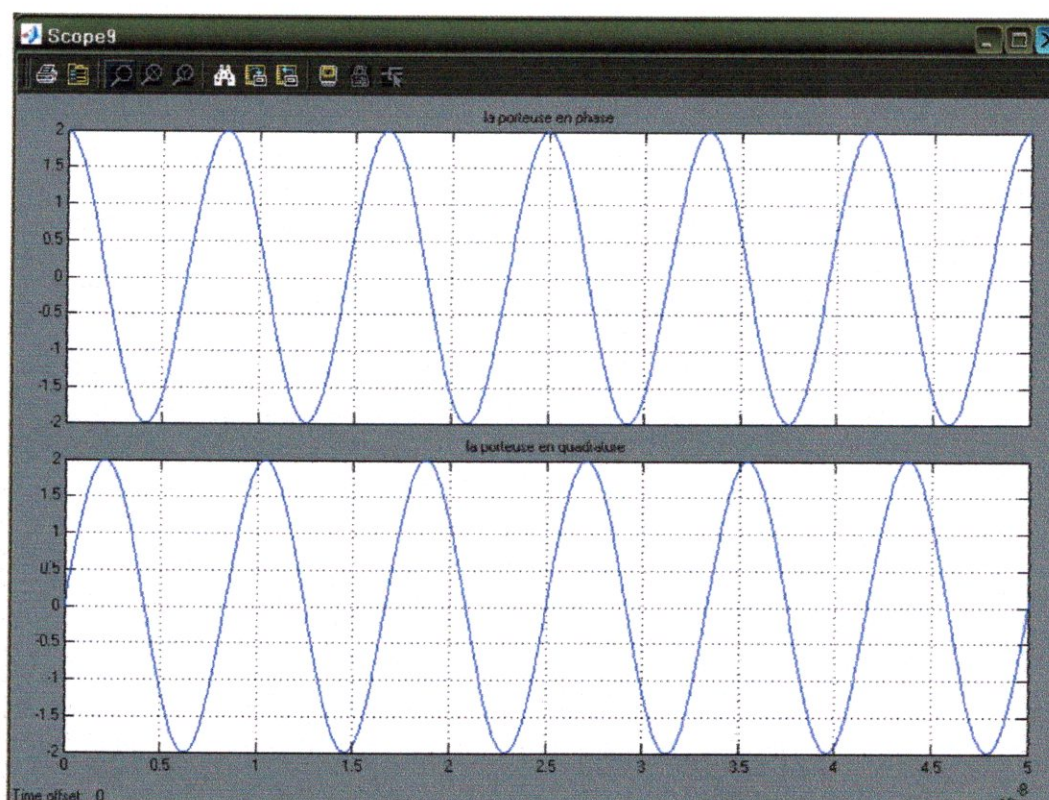


Figure IV.9 : les deux porteuses en quadrature

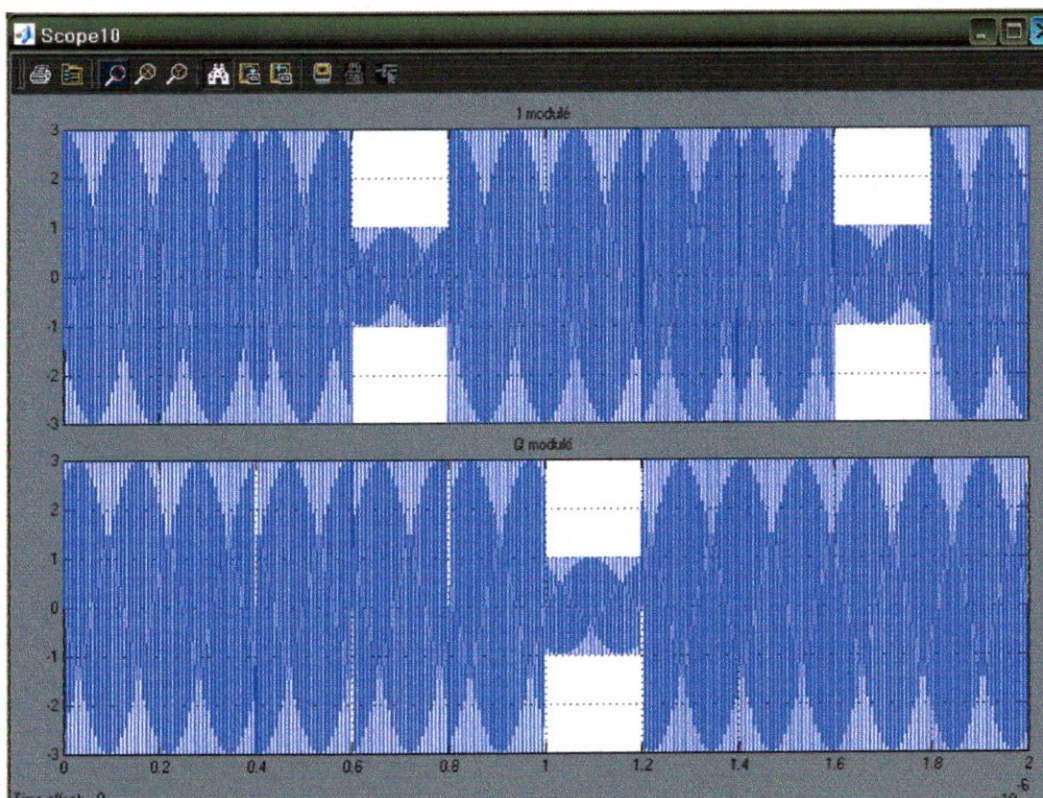


Figure IV.10 : les signaux I et Q modulés

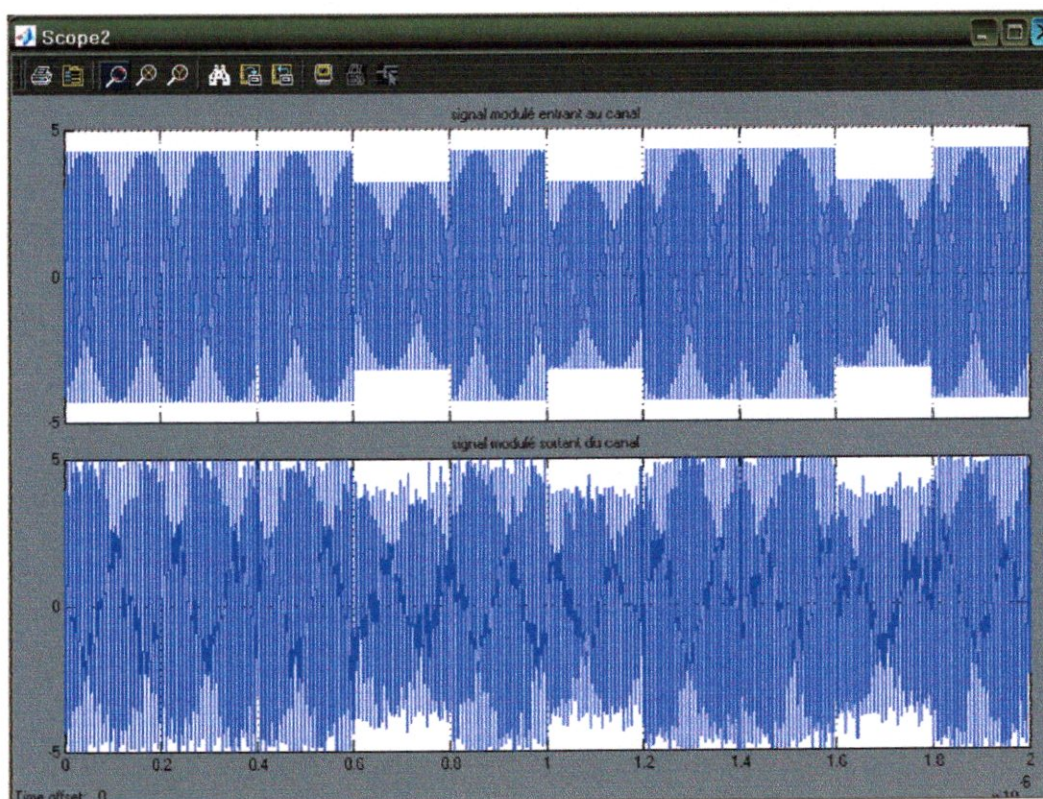


Figure IV.11 : le signal modulé à l'entrée et à la sortie du canal

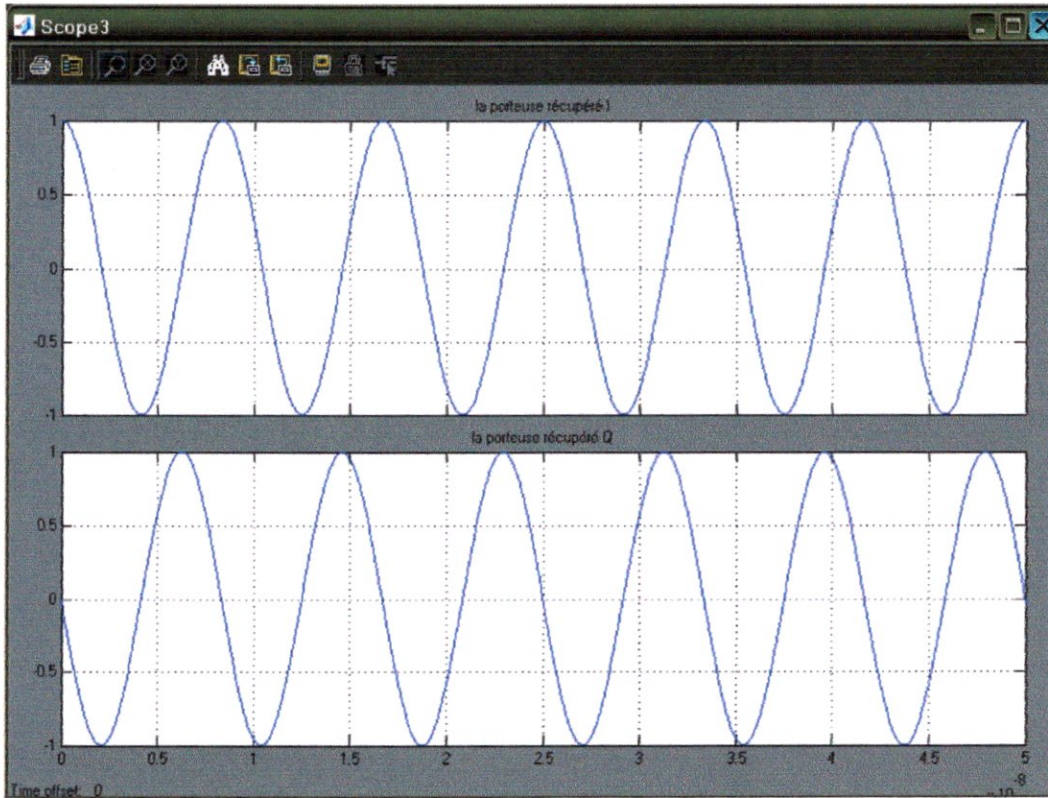


Figure IV.12 : les deux porteuses récupérées au récepteur

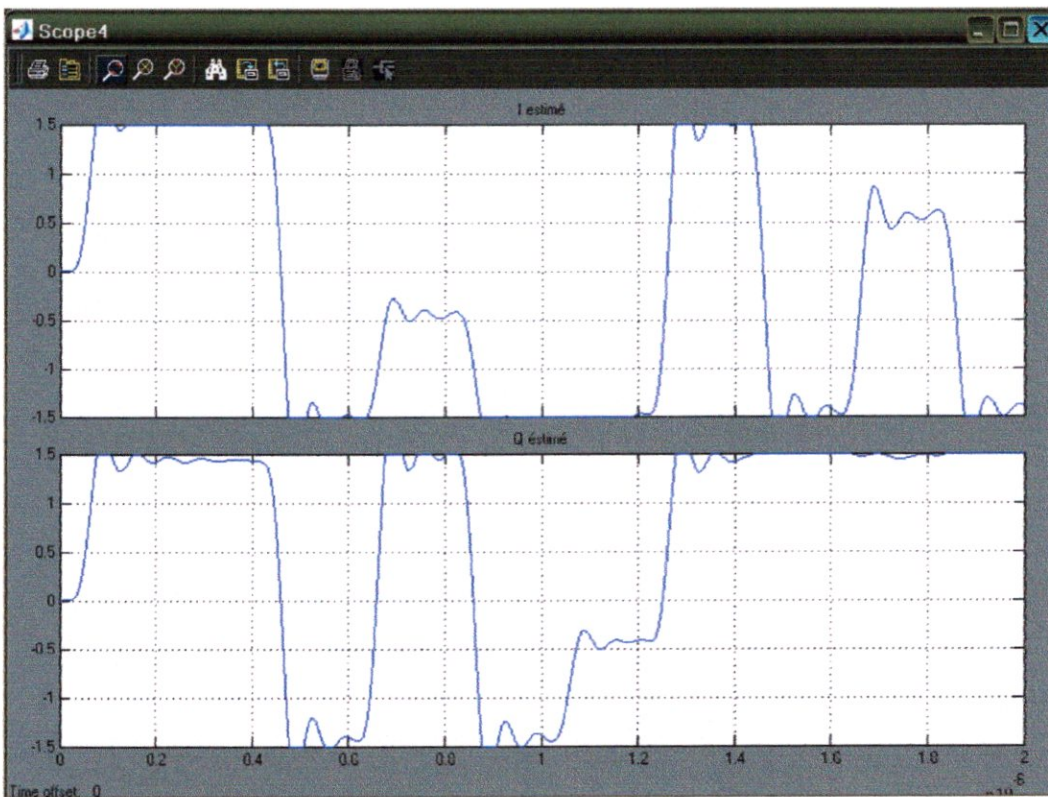


Figure IV.13 : les signaux I et Q sortants du démodulateur

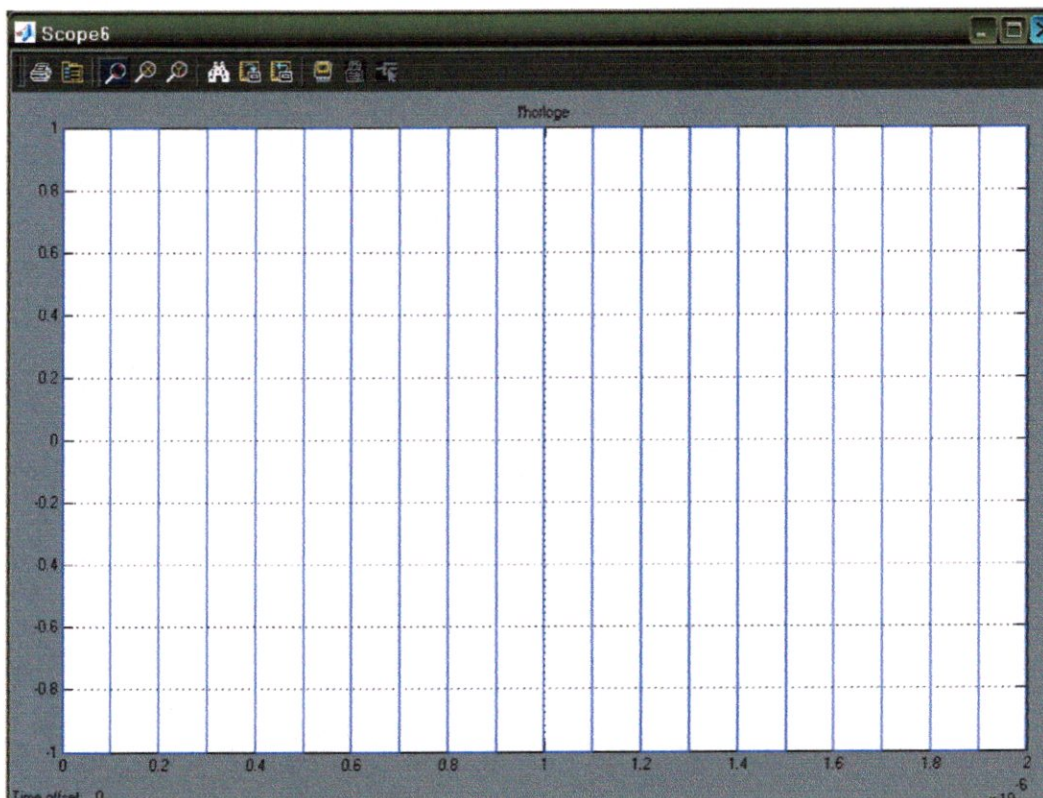


Figure IV.14 : le signal d'horloge

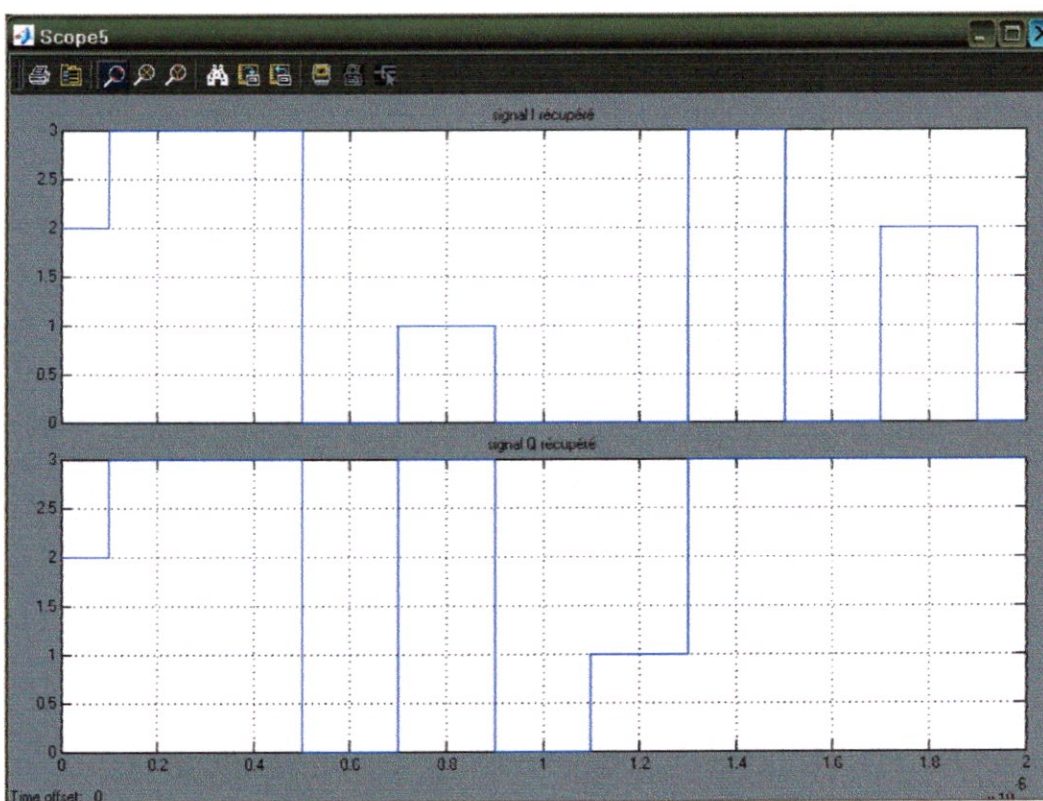


Figure IV.15 : les signaux I et Q reconstitués au récepteur

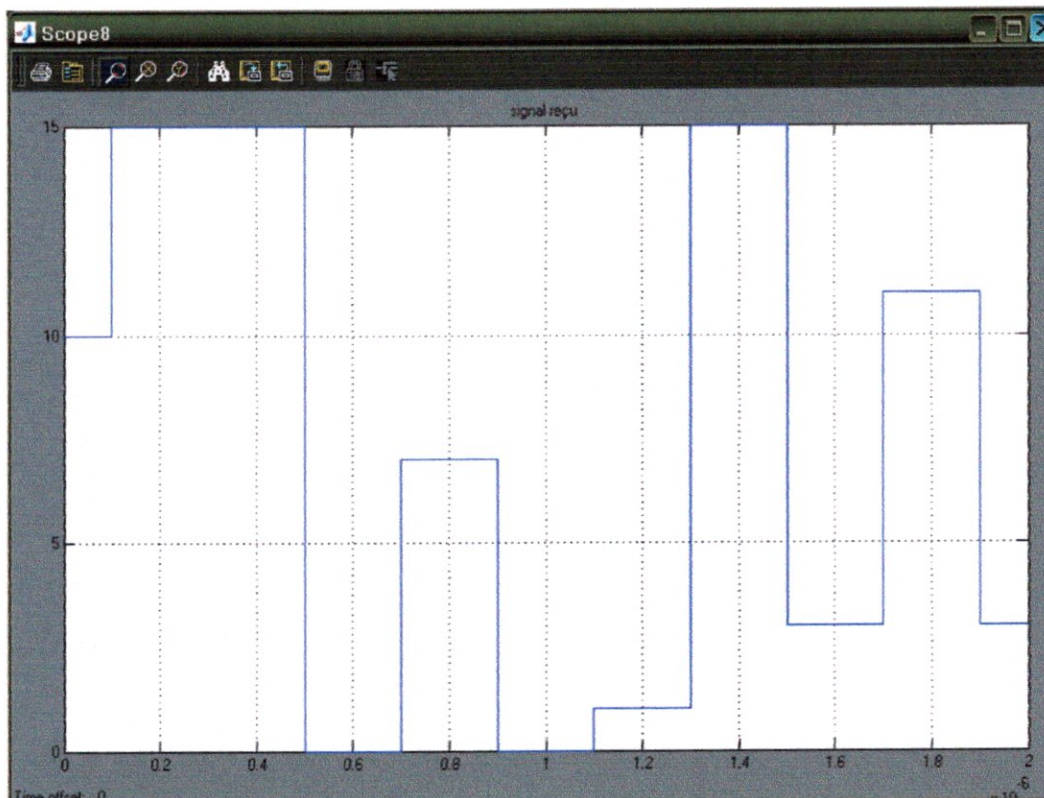


Figure IV.16 : le signal reçu

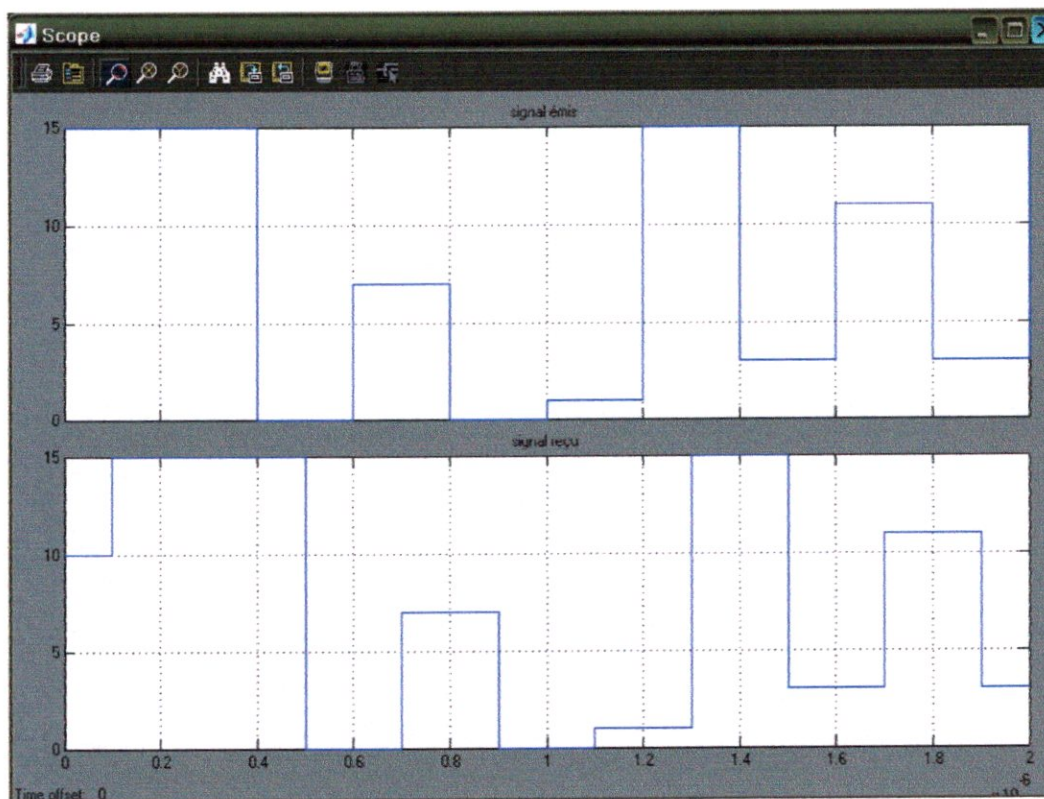


Figure IV.17 : le signal émis et le signal reçu

A partir de la figure IV.17, nous remarquons que le signal émis est reçu avec un retard, pour cela nous avons utilisé le bloc de retard à l'émission.

Nous remarquons aussi que les deux signaux (émis et reçu) sont identiques si nous annulons le décalage entre les deux signaux.

Nous pouvons comparer les résultats à l'aide d'un autre moyen qui est le calculateur de l'erreur. (Figure IV.18).



Figure IV.18 : le calculateur de l'erreur

Cette figure montre les paramètres du calculateur où :

- ❖ Le nombre des symboles émis est égal à 10 symboles.
- ❖ Le nombre des symboles erronées est égal à zéro indique que les deux signaux (émis et reçu) sont identiques.
- ❖ Le taux d'erreur par symbole est nul, où :

$$\text{Taux d'erreur} = \frac{\text{nombre des symboles erronées}}{\text{nombre des symboles total}}$$

Afin de comparer les résultats numériquement, nous tapons le nom de la variable « message » dans la fenêtre de commande du Matlab (>> message), Ensuite nous appuyons sur « entrer ». Nous avons les résultats suivants :

Le signal émis	Le signal reçu
10	10
15	15
15	15
0	0
7	7
0	0
1	1
15	15
3	3
11	11
3	3

Le tableau précédent représente les données enregistrées dans le workspace, où la première ligne indique un décalage temporel entre les signaux émis et reçu (voir figure IV.17), que nous omettons de la comparaison.

Il est à noter que la valeur 10 se trouvant à la première ligne résulte de l'addition d'une constante au signal sortant du démodulateur, de la conversion entier - binaire et binaire - entier (voir figures III.23 et III.24).

Il est clair que les deux signaux (émis et reçu) sont identiques, donc la réception est effectuée sans erreurs.

IV.5 Conclusion

Nous avons testé la simulation de la chaîne de transmission numérique, et nous avons visualisé les résultats en utilisant trois moyens différents. Nous avons vu que les signaux sont identiques, ce qui signifie que le modèle fonctionne très bien avec les paramètres choisis dans le fichier Matlab « parametres ».

Ces paramètres ont une influence sur les résultats de la simulation. Pour ce faire, nous prenons comme exemples :

- ❖ la fréquence de coupure f_c des filtres du démodulateur, si $f_c = 2 \cdot f_s$, le nombre de symboles erronés va augmenter comme l'indique la figure IV.19.



Figure IV.19 : le calculateur d'erreur quand f_c diminue

- ❖ Le changement de l'amplitude de la porteuse ou l'amplitude du VCO de la PLL dans le bloc de récupération de la fréquence porteuse la figure IV.20.



Figure IV.20 : le calculateur d'erreur après un changement de l'amplitude de la porteuse

Conclusion générale

Afin d'augmenter le débit binaire dans la transmission numérique, nous avons utilisé la technique de modulation numérique QAM 16. Cette technique consiste à diviser le signal de source en deux trains (trains modulateurs I et Q). Ces deux trains sont ensuite modulés par deux porteuses en quadrature.

A la réception, nous récupérerons les deux trains modulateurs. Ces derniers sont assemblés pour construire le signal de source.

Nous avons utilisé le logiciel Matlab/Simulink pour simuler un modèle réduit d'une chaîne de transmission. Chaque bloc du modèle a été testé et validé.

Les résultats présentés au chapitre 4, sont pour le moins encourageant ! Il est donc primordial de continuer dans cette voie.

Ce projet nous a permis de voir une application immédiate et utile des notions abordées dans les cours de traitement du signal et de télécommunications. En étudiant le modèle de transmission, nous avons appris à utiliser Simulink avec une toolbox appropriée pour simuler le système. Nous pensons qu'il pourrait être intéressant de réaliser des TP simples en utilisant ce modèle.

Cette étude pourra servir de base pour de futurs développements, la structure du logiciel utilisée permettant de faire évoluer aisément les travaux effectués. Pour améliorer le modèle, on pourra ainsi ajouter un correcteur d'erreur à la chaîne de transmission ou en complexifier certains.

Abréviations

- AM:** amplitude modulation.
- ASK:** amplitude shift keying.
- AWGN:** additive white Gaussian noise.
- DSP:** digital signal processing.
- DVB-C:** digital video broadcasting cable.
- DVB-S:** digital video broadcasting satellite.
- DVB-T:** digital video broadcasting terrestrial.
- FM:** frequency modulation.
- FSK:** frequency shift keying.
- KHz:** kilohertz.
- MAQ:** modulation d'amplitude de deux porteuses en quadrature.
- MDA:** modulation par déplacement d'amplitude.
- MDF:** modulation par déplacement de fréquence.
- MDP:** modulation par déplacement de phase.
- MHz:** megahertz.
- OOK:** on off keying.
- PM:** phase modulation.
- PLL:** phased locked loop.
- PSK:** phase shift keying.
- QAM:** quadrature amplitude modulation.
- SSR:** surveillance secondary radar.
- TEB:** taux d'erreur par bit.
- TES:** taux d'erreur par symbole.
- VCO:** voltage controlled oscillator
- VDL:** VHF digital links.
- VHF:** very high frequency.

bibliographie

- [1] BERNARD sklar, [2001]
digital communications
“ fundamentals and applications”
U.S.A
Editions : Prentice Hall
- [2] AGILENT, [2001]
digital modulation in communications
systems-an introduction
U.S.A
Editions : Agilent technologies
- [3] Patrice KADIONIK, [2005]
base de transmission numérique,
ENSEIRB Bordeaux.
www.enseirb.fr.
- [4] OACI [1997]
annexe 10 (télécommunications aéronautiques)
volume III
- [5] Cours de télécommunication [2004-2005]
institut de l'aéronautique, option : installation,
quatrième année, université de Blida.
- [6] عبد الكريم البيكو [2003]
الدليل المرجعي و التعليمي Matlab 6.5
سوريا
الناشر: شعاع
- [7] MATHWORKS, [2002]
Matlab 6.5 help.
- [8] le site web : WWW.bensoire.
free.fr/expe/simul_matlab.htm