

TP1 : Prise en main de SIMULINK

(Partie communication numérique)

1. Objectifs : Les objectifs de ce TP sont :

- Prendre en main le logiciel de simulation Simulink Matlab ;
- Voir quelques blocs essentiels à utiliser dans les TP's de communication numérique pour ce semestre
- Apprendre comment modifier les paramètres des blocs et simuler un modèle Simulink.

2. Présentation du Simulink/Matlab :

MATLAB (MATrix LABoratory) est un logiciel de calculs numériques et de visualisations graphiques destiné aux ingénieurs et scientifiques.

Le Simulink est l'extension graphique de Matlab permettant de représenter les fonctions mathématiques et les systèmes sous forme de diagramme en blocs et de simuler le fonctionnement de ces systèmes. L'utilisation du SIMULINK suit en général les étapes suivantes :

1. Etablir (dessiner) le modèle du système en utilisant les blocs présent en librairie ;
2. Placer des sources de signaux aux entrées du modèle : générateurs numériques ou analogiques
3. Placer des instruments de visualisation en sortie du modèle : Scope, Display, spectrum scope, ...
4. Paramétrer et lancer la simulation du fonctionnement du modèle : double clic sur le bloc puis modifier les paramètres et Simulation/Start pour lancer la simulation ;
5. Observer les résultats à l'aide des instruments de visualisation.

3. Création d'un modèle SIMULINK

Pour ouvrir Simulink taper "simulink" dans la fenêtre de commande du matlab ou cliquez sur l'icône présent dans la fenêtre principale de MATLAB.

Une boîte de dialogue contenant la liste des librairies apparaît à l'écran. Chaque librairie contient un ensemble des blocs qui servent à la création des diagrammes, ou schémas blocs.



The screenshot shows the MATLAB R2013a interface. The Command Window contains the command `simulink`. The Simulink Library Browser window is open, showing a list of libraries on the left and a grid of blocks on the right. Annotations with arrows point to the 'simulink' command, the Simulink Library icon, and the 'Model-Wide Utilities' block.


Ouvrir Simulink

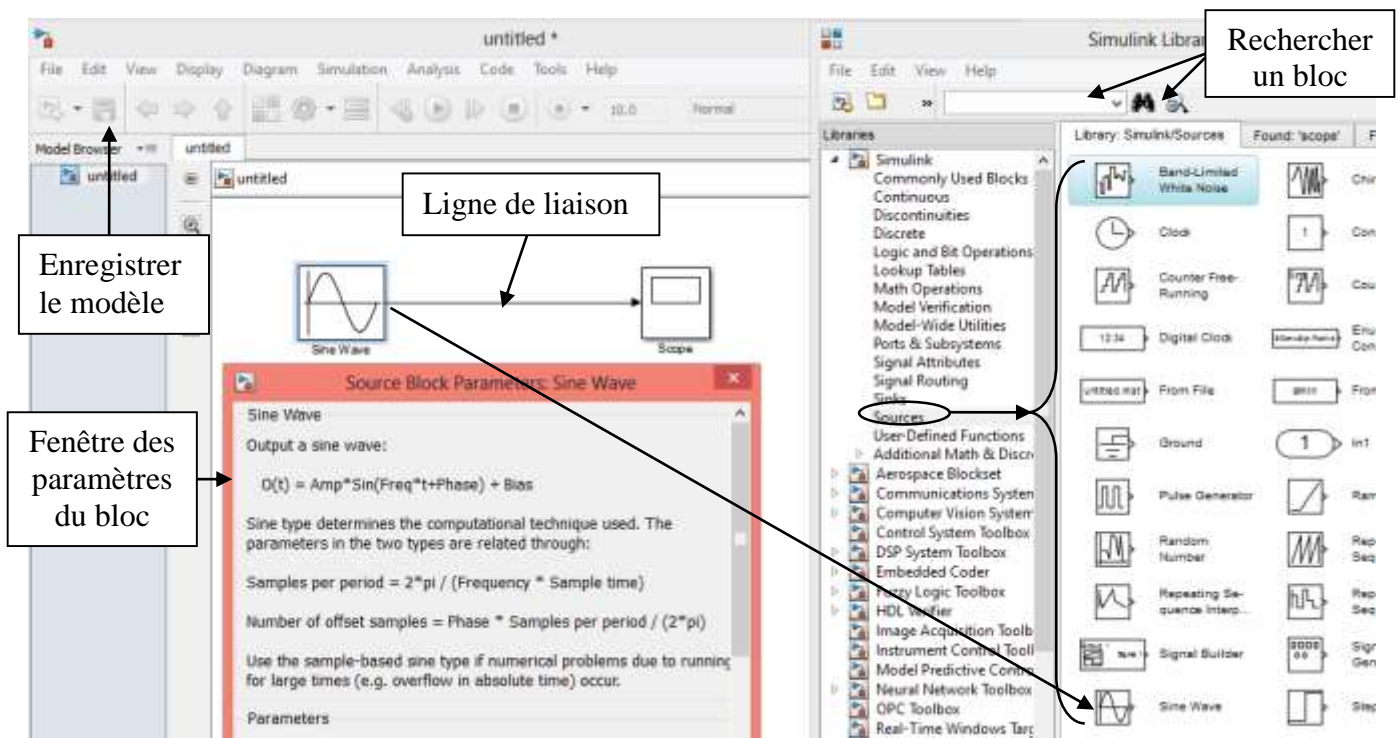
Crée un nouveau modèle Simulink


Chaque librairie contient des blocs

Librairies Simulink

Pour créer un modèle Simulink suivre les étapes suivantes:

1. Dans le menu du Simulink Library Browser sélectionnez **File > New > Model** ou cliquez sur l'icône  , une fenêtre de travail **Untitled** s'ouvrira.
2. Ouvrir les collections de blocs par double clic (Vous pouvez chercher n'importe quel bloc en tapant son nom dans la barre de recherche du Simulink Library Browser) ;
3. Faire glisser dans la fenêtre de travail les blocs dont vous avez besoin pour construire le diagramme (Vous pouvez prendre une copie d'un bloc en cliquant sur le bouton droit de la souris et le faire glisser);
4. Faire des liaisons entre les blocs à l'aide de la souris ;
5. Changer les paramètres de n'importe quel bloc en faisant double clic sur ce bloc, une fenêtre s'ouvrira.
6. Lorsque vous avez terminé le diagramme, enregistrer le modèle dans un fichier: **File > Save** ou l'icône d'enregistrement et donner un nom au modèle (*.mdl).



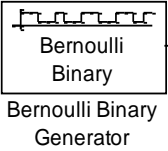
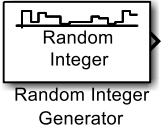

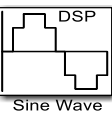
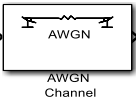
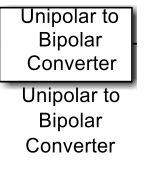
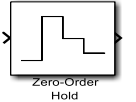
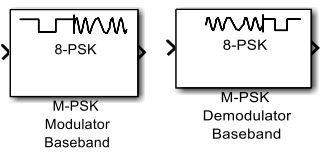
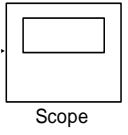
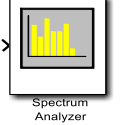
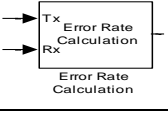
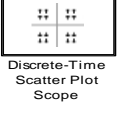
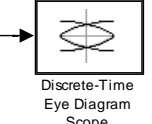
7. Simuler le montage: dans le **menu Simulation/Start** ou en cliquant sur l'icône .
8. Pour régler le temps de simulation aller dans la barre d'outils du menu et écrire sa valeur.



Remarques :

- Pour grouper plusieurs blocs en un seul bloc (sous-système), sélectionnez les blocs dont vous voulez qu'ils soient contenus dans le sous-système, puis sélectionnez **Create subsystem** dans le menu **Edit** (ou bouton droit de la souris)
- Pour écrire un texte double clic sur la ligne puis écrire le texte.

4. Description de quelques blocs :

| Nom | Librairie | Description | Schéma |
|--------------------------------------|--|--|---|
| Bernoulli Binary Generator | Communication System Toolbox/Comm Sources/Random Data Sources | Générateur numérique : génère aléatoirement des 0 et 1 tous les T_b secondes. Probability of zeros = le nombre de zéros par rapport au nombre des 1 pour: Probability of zeros = 0.5 on a le nombre des zero = nombre des 1) et Sample time = le temps d'un bit T_b |  |
| Random Integer Generator | Communication System Toolbox>Comm Sources/Random Data Sources | Générateur numérique génère aléatoirement des entiers réparties uniformément dans l'intervalle $[0, M-1]$ où M (M-ary number) est le nombre de M-matrice dans T_b secondes (Sample time) |  |
| Pulse Generator | Simulink/Sources | Générateur d'horloge |  |
| Sin Wave | DSP System Toolbox/Sources | Générateur des signaux sinusoïdaux (sinus ou cosinus). Ces paramètres essentiels sont Amplitude , Phase offset (phase initial en rad pour pi/2 on a le cosinus), Frequency (Hz) |  |
| AWGN Channel | Communication System Toolbox>Channels | Générateur d'un bruit blanc additif ces paramètres essentiels sont : Eb/No (SNR par bit), Input signal power (puissance du signal en watt), Symbol period (temps d'un bit) |  |
| Unipolar to Bipolar Converter | Communication System Toolbox>Utility Blocs | Génère un signal de M symbole (M-ary number). Si les symboles sont constitués de 'n' bits, $M = 2^n$. Lorsqu'on a deux symboles 0 et 1", $M = 2$, il génère alors un signal analogique de durée T_b qui vaut +1V, si le symbole d'entrée vaut 1, ou -1V, si le symbole d'entrée vaut 0. |  |
| Zero-Order Hold | Simulink/Discrete | Echantillonneur dont la fréquence d'échantillonnage $F_e = 1/T_e$. Simple Time est fixée de façon à observer les signaux dans l'intervalle $[-2/T_e : 2/T_e]$ Hz. |  |
| M-PSK Modulator M-PSK Demodulator | Communication Systeme Toolbox/ Modulation/ Digital Passband Modulation | Blocs de modulateurs et démodulateurs de modulations numériques de phases M_PSK son paramètre M-ary number donne le nombre de bits (exp M=8 on a 8-PSK) |  |
| Scope | Simulink/Sinks | Oscilloscope signal temporel. Pour que le scope vous donne la main pour plusieurs lignes: double clic sur scope, puis cliquer sur l'icône ' parameters ' puis dans ' number of axes ' taper le nombre des axes. |  |
| Spectrum Analyser | DSP System Toolbox/Sinks | Oscilloscope du spectre du signal. Il affiche la puissance ou la densité de puissance (Puissance density or Puissance) (en Watt ou en dBw ou en dBm) d'un signal. |  |
| Error Rate Calculation | Communication System Toolbox> Comm Sinks | Calcul le Taux d'Erreur Binaire, il compare les données d'entrée de l'émetteur avec celles d'un récepteur. Il donne le $BER = \frac{\text{Nombre de bits erronés}}{\text{Nombre total de bit transmis}}$ |  |
| Discrete-Time Scatter Plot Scope | Communication System Toolbox> Comm Sinks | Diagramme de constellation présente le signal modulé dans le plan complexe qui fait correspondre à chaque signal élémentaire un point $C_k = a_k + j b_k$ permet de différencier chaque type de modulation. |  |
| Discrete-Time Eye Diagram Scope | Communication System Toolbox> Comm Sinks | Oscillogramme affiche plusieurs traces d'un signal modulé pour produire un diagramme de l'œil qui permet d'étudier la qualité d'un signal numérique à la sortie du canal de transmission. |  |

TP2 : Modulation en bande de base (Codage en ligne)

1. Objectifs du TP : Les grandes lignes de ce TP sont :

- L'étude des méthodes de codage en ligne les plus utilisées en communication numérique ;
- L'étude des densités spectrales de puissance associées à ces divers codes en ligne

2. Rappels théoriques :

2.1 Introduction

La transmission des données numériques peut se faire de deux manières différentes : **Transmission en bande de base** ou **Transmission sur fréquence porteuse** ou modulation numérique.

La transmission est dite en bande de base si elle ne subit aucune transposition de fréquence par modulation. Donc ce type de transmission ne peut être utilisé que sur support de type métallique (paires torsadées, câble coaxial) et sur de très courtes distances comme les réseaux locaux (exp : transmission sur le câble coaxial dans un réseau local, transmission entre deux équipements informatiques éloignés dans une même entreprise). Cette modulation permet d'obtenir des transmissions à grand débit mais à faible portée..

2.2 Définition :

Le codage en bande de base ou codage en ligne, consiste à faire associer à chaque élément a_k un signal continu (signal électrique, électromagnétique, optique, ...) de durée T_b .

2.3 Choix d'un code : Le choix d'un code par rapport à un autre dépend de plusieurs critères:

- Largeur de sa plage de fréquences : on a la rapidité de modulation maximal $R_{max} = 2B$ donc la largeur de bande du support limite la rapidité de modulation et donc le débit maximal sur la ligne alors la bande passante doit être la plus étroite possible.
- Absence de composante continue (Présence d'énergie à la fréquence nulle)
- Présence d'une raie à la fréquence rythme ce qui donne la possibilité régénération de l'horloge au niveau du récepteur
- Résistance au bruit (espacement des niveaux)
- Complexité du codage (coût et vitesse de codage)
- Absence de polarisation du support (donne une facilité d'installation)
- Limitation de la désynchronisation : Pour que l'horloge du décodeur reste synchrone avec celle du codeur, l'intervalle de temps entre deux changements de niveau de signal doit être le plus court possible.

Aucun des codes ne présentent toutes ces caractéristiques en même temps. Il faudra juste trouver un compromis entre ces caractéristiques suivant l'application.

2.3 Les principaux codes à deux états : Les codages utilisés peuvent être classés selon le nombre de niveaux électriques : 2 niveaux (exp : NRZ, RZ, biphasés, miller, ...), 3 niveaux (exp : bipolaires, bipolaires haute densité, ...) et à multi-niveaux (exp: 2B1Q, ...).

Dans le tableau suivant les caractéristiques et des comparaisons entre les codes NRZ, RZ et Manchester :

| | NRZ | RZ | Manchester |
|---------------|--|---|--|
| Description | $1 \rightarrow +V : T_b$ $0 \rightarrow -V : T_b$ | $1 \rightarrow \begin{cases} +V & : T_b/2 \\ 0V & : T_b/2 \end{cases}$ $0 \rightarrow 0V : T_b$ | $1 \rightarrow \begin{cases} -V : T_b/2 & 1^{er} \text{ moitié} \\ +V : T_b/2 & 2^{ème} \text{ moitié} \end{cases}$ $0 \rightarrow \begin{cases} +V : T_b/2 & 1^{er} \text{ moitié} \\ -V : T_b/2 & 2^{ème} \text{ moitié} \end{cases}$ |
| Signal | | | |
| Spectre | $S(f) = V^2 T_b \left(\frac{\sin(\pi f T_b)}{\pi f T_b} \right)^2$ $= V^2 T_b \text{sinc}^2(\pi f T_b)$ $\text{Max}(S(f)) = V^2 T_b$ | $S(f) = \frac{V^2 T_b}{4} \left(\frac{\sin(\pi f \frac{T_b}{2})}{\pi f \frac{T_b}{2}} \right)^2$ $= \frac{V^2 T_b}{4} \text{sinc}^2(\pi f \frac{T_b}{2})$ $\text{Max}(S(f)) = V^2 T_b / 4$ | $S(f) = V^2 T_b \frac{[\sin(\pi f \frac{T_b}{2})]^4}{[\pi f \frac{T_b}{2}]^2}$ $\text{Max}(S(f)) = V^2 T_b$ |
| Avantages | <ul style="list-style-type: none"> • Simple à mettre en oeuvre • Bande passante : $B = 1/T_b$ • Déterminer la présence ou non du signal • Bon rapport signal sur bruit | <ul style="list-style-type: none"> • Simple (se réalise par AND logique entre le message le signal de l'horloge). • Son spectre contient des raies aux fréquences multiples de la fréquence d'horloge, d'où facilité de récupération de cette horloge à la réception. | <ul style="list-style-type: none"> • DSP centrée autour de $f = 1/T_b$. Ce code est donc la composante continue est nulle et il est bien adapté aux hautes fréquences. • L'horloge peut être récupérer puisque ce code garantie au moins une transition par période • La synchronisation des échanges entre émetteur et récepteur est toujours assurée, • Très peu sensible aux erreurs de transmission |
| Inconvénients | <ul style="list-style-type: none"> • DSP, centrée autour zéro alors composante continue, non nulle, ce code est donc mal adapté aux hautes fréquences. • Les longues séries de 1 ou de 0, entraînent un perte de synchronisation. • Pas des aux fréquences multiples de l'horloge, alors récupération d'horloge non garantie, il faut ainsi utiliser une ligne indépendante pour l'horloge, ce qui limite l'utilisation du NRZ aux courtes distances. • Nécessite de maintenir la polarité | <ul style="list-style-type: none"> • DSP, centrée autour zéro alors composante continue, non nulle, ce code est donc mal adapté aux hautes fréquences. • La bande passante est doublée par rapport à NRZ : $B = 2/T_b$ • La tension nulle (0 Volt) dans le bit '0', entraine une difficulté récepteur d'établir une différence entre le bit 0 et une coupure de transmission. | <ul style="list-style-type: none"> • Bande passante doublée par rapport au codage NRZ : $B = 2/T_b$ • Débit est deux fois plus élevé que pour un codage NRZ, ce qui est bien trop élevé pour être physiquement stable et entrainerait les phénomènes de paradiaphonie. • Nécessite de maintenir la polarité, une inversion de polarité entraîne une inversion des informations binaires |
| Application | <ul style="list-style-type: none"> • Les signaux transmis selon les normes V24, RS232, RS421, RS422, RS48 | <ul style="list-style-type: none"> • Entre l'ordinateur et ses périphériques. | <ul style="list-style-type: none"> • Ethernet (IEEE 802.3) sur câble coaxial, • Profibus (Process Field Bus), • Transmission MBP (Manchester Bus Powered ...) |

Nom & Prénom :

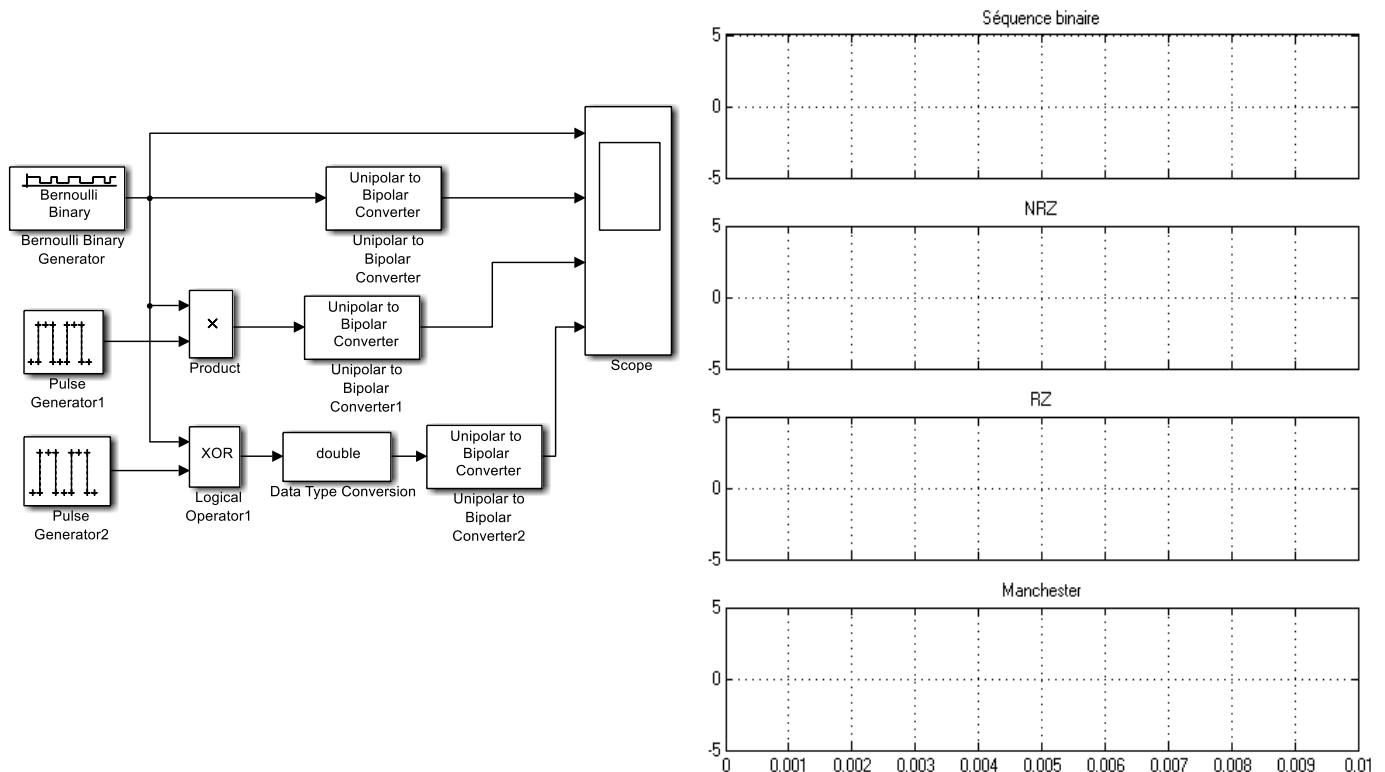
Nom & Prénom :

TP2 : Modulation en bande de base (Codage en ligne)

3. Simulation:

3.1 Représentation temporelle des codes NRZ, RZ et Manchester :

– Réaliser le modèle Simulink des trois codes suivant :



- Régler les blocs comme suit :

- Bernoulli Binary Generator: **Sample time** = $1e-3$ ($T_s=1ms$)
- Unipolar to Bipolar Converter : **M-ary number** = 2
- Générateur de Pulse: **Pulse type** =Sample Based, **Périod** = 2, **Pulse width**= 1, **Sample time**= $5e-4$
- Logical operator (Simulink>Logic and Bit Operations): **Operator** = XOR
- Data Type Conversion (Simulink>Signal Attributes): **Out Put Data** = double.

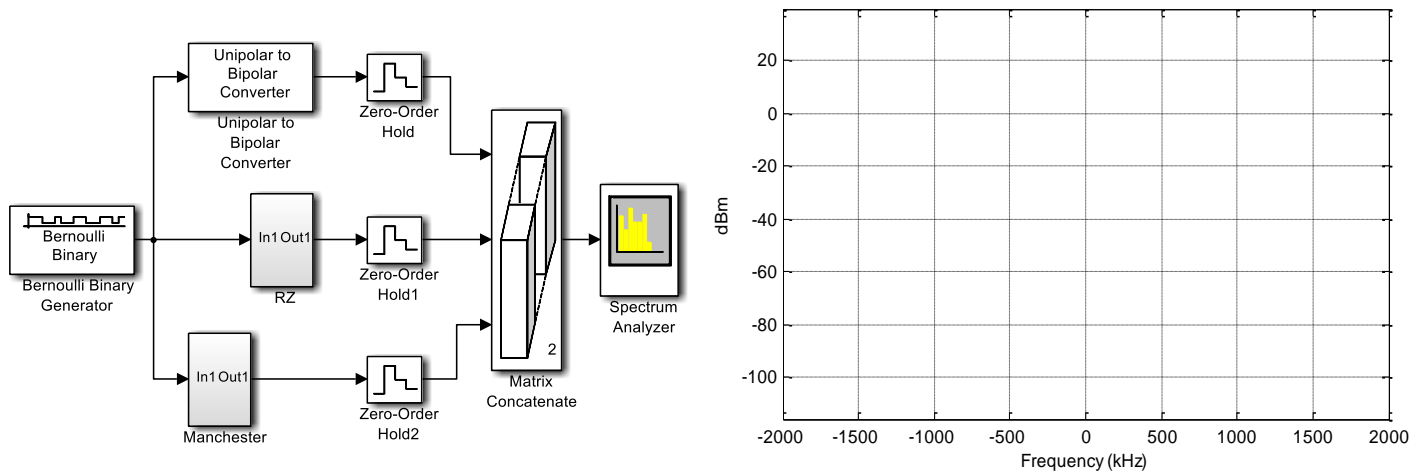
1. Calculer le débit binaire du générateur $D=$

2. Simuler le montage dans 0.01 et tracer les signaux trouvés. Donner la séquence binaire sur le graphe

3.2 Représentation spectrale de trois codes :

– Dans le schéma précédant grouper les blocs du code RZ et Manchester dans des sous-systèmes et ajouter et régler les autre blocs comme suit :

- Zero-Order Hold: **Simulink/Discrete** : **Simple Time**= $2.5e-4$
- Matrix Concatenate: **Simulink> Math Operations**: Number of inputs=3



3. Fixer le temps de simulation à '1' et tracer les représentations fréquentielles des trois codes.

4. Remplir le tableau et faire des comparaisons et des commentaires p les trois codes

| | NRZ | RZ | Manchester | Comparaison et Commentaires |
|----------------|-----|----|------------|-----------------------------|
| Bande passante | | | | |
| Débit binaire | | | | |
| Raie à zéro | | | | |
| Raie à f_b | | | | |

