

# Notice d'utilisation

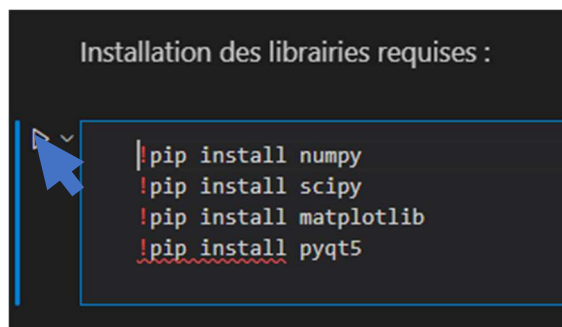
De l'application de simulation pour l'enseignement des télécommunications numériques

*L'application permet de représenter l'influence de différents paramètres sur la transmission en bande de base d'une suite binaire pseudo aléatoire. Toutes les instructions nécessaires pour son bon fonctionnement sont présentées ci-après.*

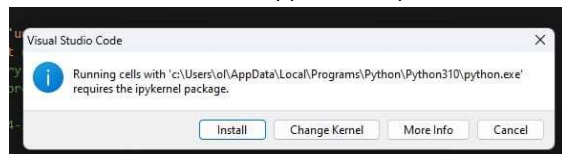
## 1. Instructions d'installations

1 – Ouvrez le fichier '**TeleSimU.ipynb**' avec un logiciel d'exécution comme Visual Studio Code ou Jupyter notebook d'Anaconda. A défaut, installez-le : <https://code.visualstudio.com/Download> ou <https://www.anaconda.com/download>

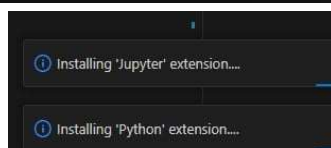
2 – Exécutez la première cellule, pour installer toutes les librairies nécessaires.



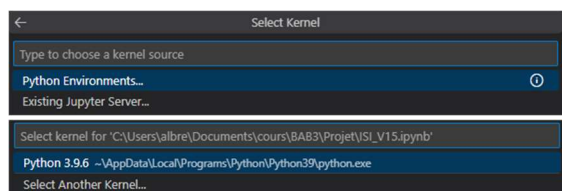
Si une fenêtre similaire apparaît, cliquez sur '**Install**'



et l'extension Python Jupyter s'installera à Visual studio.

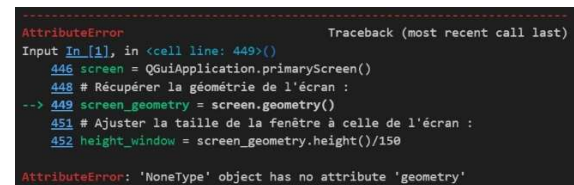


Lors du choix du noyau, choisissez '**Python Environments**' puis une version de Python supérieure à 3.9.

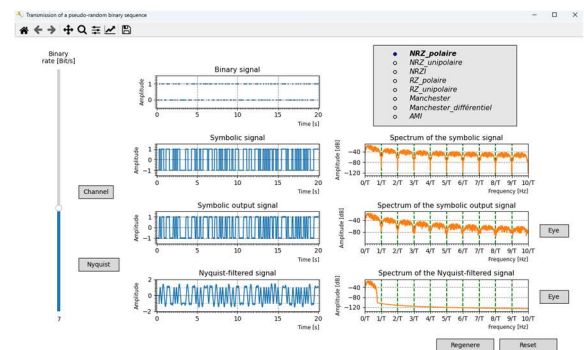


Si l'exécution n'aboutit à rien, installez ces librairies dans l'invité de commande.

3 – Faites de même pour la cellule de code suivante. Si vous obtenez cette erreur, réexécutez la cellule. La première exécution du script a permis d'obtenir les dimensions de votre écran.



4 – Si une fenêtre comme celle-ci s'ouvre, c'est que tout s'est bien déroulé. Sinon, voir Section 5.



## 2. Fonctionnalités générales

Cette fenêtre principale montre l'évolution du signal à 4 moments clés de sa transmission.

Le graphe en haut à gauche représente la suite de bits. Ensuite, par-dessous, se trouve le signal représentant l'attribution des symboles d'un code précis (ici : **NRZ\_polaire**) pour chaque bit ainsi qu'à sa droite la transformée de Fourier de ce signal. Le troisième moment clé illustrant le signal en sortie du canal de transmission. Celui-ci est représenté juste en-dessous avec encore une fois une évolution temporelle à gauche et fréquentielle à droite. De même pour le dernier, à la sortie du filtre de Nyquist. Il y a également 6 boutons, une glissière, un cadre avec 8 sélections possibles ainsi qu'une "toolbar".



#### Toolbar :



Si vous laissez votre souris immobile sur un icône, une explication de son utilité apparaîtra. Cette barre permet de modifier l'affichage des graphes mais en aucun cas les données. Elle est incluse lors de l'exécution de l'affichage d'un graphe sur une fenêtre extérieure à l'environnement du script.

#### Cadre de sélection :

Ce cadre représente tous les codes possibles pour affecter une forme de symbole au signal binaire. Pour changer de code, il suffit de cliquer sur le nom du code, il se mettra alors en gras pour indiquer qu'il a bien été sélectionné.

#### Glissière :

Elle permet de modifier la valeur du paramètre renseigner au-dessus (ou à gauche). En cliquant sur la glissière, cela modifie la valeur qui est alors affichée en dessous (ou à droite).

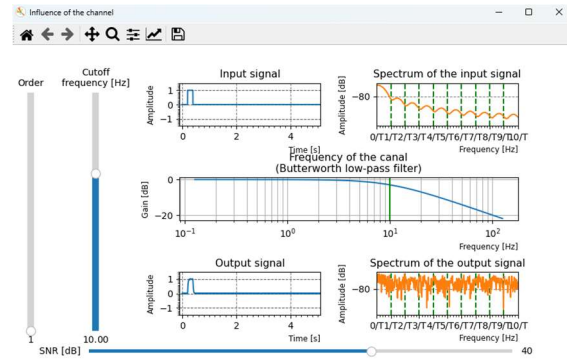
#### Bouton :

Lorsque l'on clique sur un bouton cela ouvre une seconde fenêtre avec d'autres graphes et d'autres éléments d'interactions. Les seules exceptions sont ; le bouton '**Regenere**' qui comme son nom l'indique régénère une autre suite pseudo aléatoire de bits et le bouton '**Reset**' qui permet de remettre toutes les glissières à leur valeur initiale.

### 3. Fonctionnalités avancées

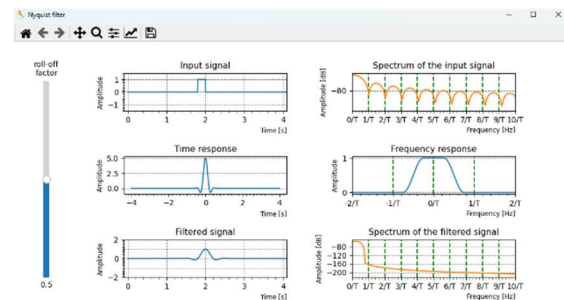
Il est possible d'ouvrir en tout 4 fenêtres supplémentaires interconnectées entre-elles. Lorsque vous changez le débit binaire ou bien le choix du code utilisé sur la fenêtre principale, la modification se répercute également sur toutes les autres fenêtres.

En cliquant sur '**Channel**', il est possible d'ouvrir une première fenêtre pour modifier les paramètres du canal de transmission. Tels que l'ordre, la fréquence de coupure et le rapport signal sur bruit.

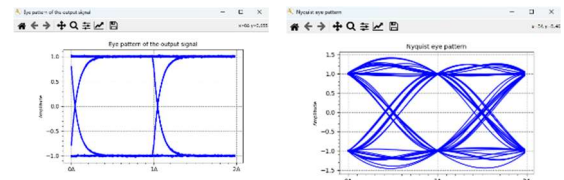


Les modifications engendrées seront automatiquement mises à jour pour toutes les fenêtres et cela vous permettra de voir l'impact de vos changements.

La seconde fenêtre est ouvrable par le bouton '**Nyquist**' et permet de modifier le facteur de roll-off du filtre de Nyquist.



Les deux derniers boutons '**Eye**' permettent de représenter les yeux des signaux à leur gauche.



#### Animation :

Il est possible de voir la construction du diagramme de l'œil pas à pas. Pour ce faire, rendez-vous à nouveau dans l'environnement du script et exécutez la dernière cellule de code.

Animation de l'oeil de Nyquist :

```
from matplotlib import rc
%matplotlib qt

delta_test=1/symbolic_rate
delta=1/bit_rate_slider.val
ratio_bi_sy_rate = delta_test/delta
```



Vous devriez obtenir un invité de saisie similaire :

Faites votre choix (1 ou 2) : (Press 'Enter' to confirm or 'Escape' to cancel)

Signal à représenter :

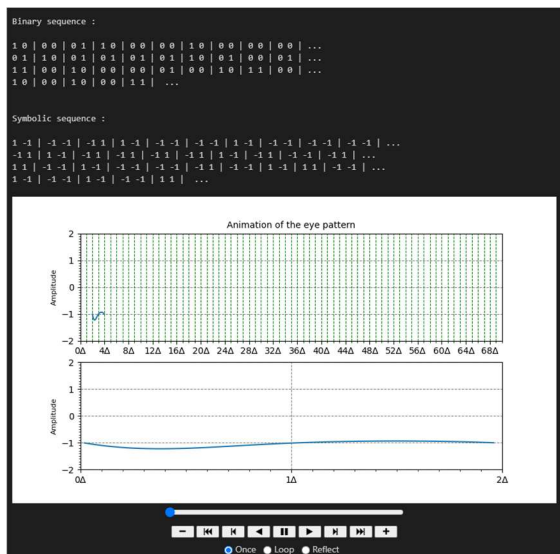
1 : Symbolic output signal

2 : Nyquist-filtered signal

Il ne vous reste alors qu'à faire votre choix et l'animation se créera.

Le résultat s'affiche dans le script en-dessous de la cellule de code.

Un rappel des premiers bits est affiché ainsi que les symboles correspondants, ce qui vous permet de comprendre chaque courbe une à une.



Les boutons en dessous du graphe vous permettent de gérer l'animation.

- / + : Régler la vitesse,

|<< / >>| : Aller au tout début/fin,

|< / >| : Afficher la courbe précédente/suivante,

< / > : Démarrer l'animation à l'envers/à l'endroit,

|| : Faire pause.

La glissière permet de voyager à différents moments de l'animation. Et les sélecteurs permettent de choisir entre une animation unique, en boucle ou bien en réflexion.

#### 4. Limitations

Chaque glissière a un minimum et un maximum, ce qui limite les observations possibles. Néanmoins, seules les glissières de débit binaire, de fréquence de coupure et du rapport signal sur bruit, ont un impact non-négligeable sur la qualité d'une transmission.

En effet, l'ordre du filtre est un paramètre plutôt fixe qui reste unitaire dans la majeure partie des cas. Quant au facteur de roll-off, il n'affecte que la réception du signal.

Le débit binaire doit être réglé pour assurer un certain nombre de transitions différentes de bits, il est donc conseillé de le régler entre 5 et 7 [bit/s] et entre 3 et 5 [bit/s] pour les 2 codes RZ et les 2 codes Manchester.

La fréquence de coupure du filtre passe-bas représentant le canal est dépendant de cette même valeur. Pour observer de l'interférence conséquente entre symboles, il faut placer la valeur en deçà de la moitié du débit binaire. Par ailleurs, pour que l'interférence disparaisse, il faut placer la valeur au-dessus de 2 fois la valeur du débit binaire (ou 3 fois pour les 4 codes particuliers mentionnés ci-avant).

Enfin, pour le rapport signal sur bruit, il faut le laisser dans la plage des 20 à 40 [dB], en-dessous le bruit sera fort impactant et au-dessus négligeable.

#### 5. Support

Aide pour l'installation de Visual Studio :

<https://code.visualstudio.com/docs#vscode>

Aide pour l'installation de l'extension Jupyter :

<https://code.visualstudio.com/docs/datascience/jupyter-#notebooks#:~:text=You%20can%20create%20a%20Jupyter,picker%20in%20the%20top%20right.>

#### 6. Contact

Pour tout autre problème et/ou défaut veuillez nous contacter par mail :

Denis Albrecq : [221084@umons.ac.be](mailto:221084@umons.ac.be)

Tom Fontaine : [221537@umons.ac.be](mailto:221537@umons.ac.be)

Ou bien notre promotrice :

Véronique Moeyaert :

[veronique.moeyaert@umons.ac.be](mailto:veronique.moeyaert@umons.ac.be)

