

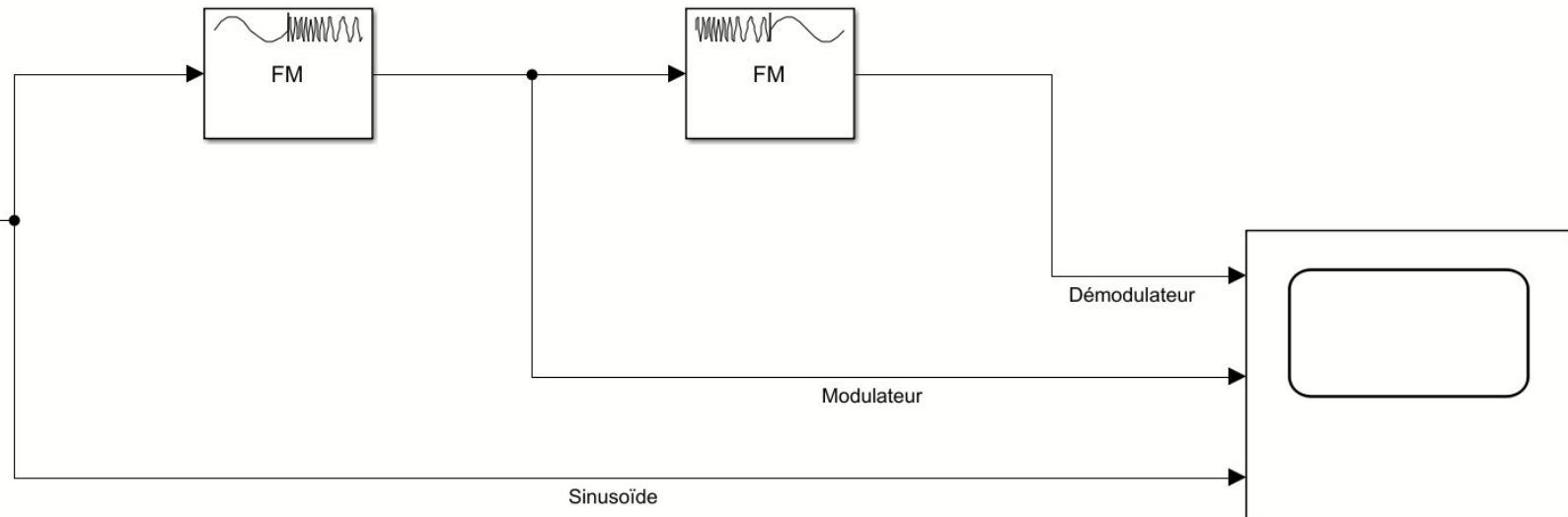


Présentation de la SAE2.02

Par ALAMELOU Rohan, BOUGHLEM Bilel et
GRONDIN Emmanuel

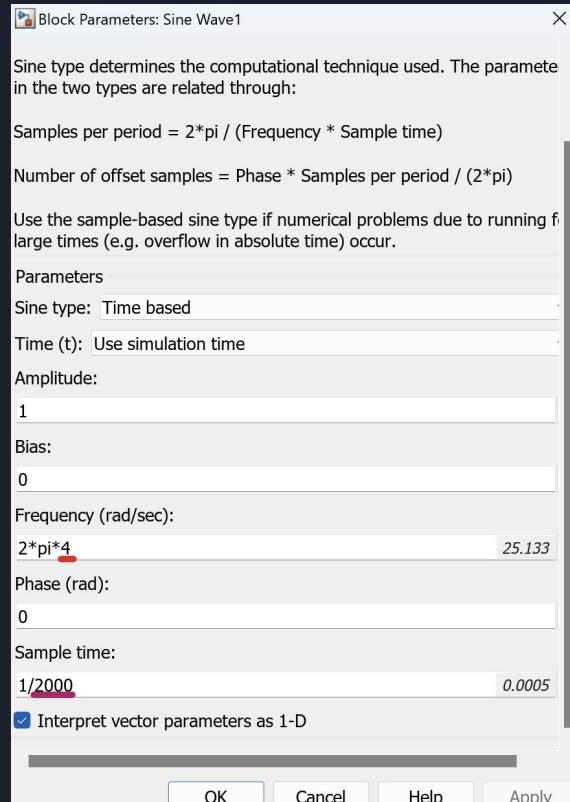
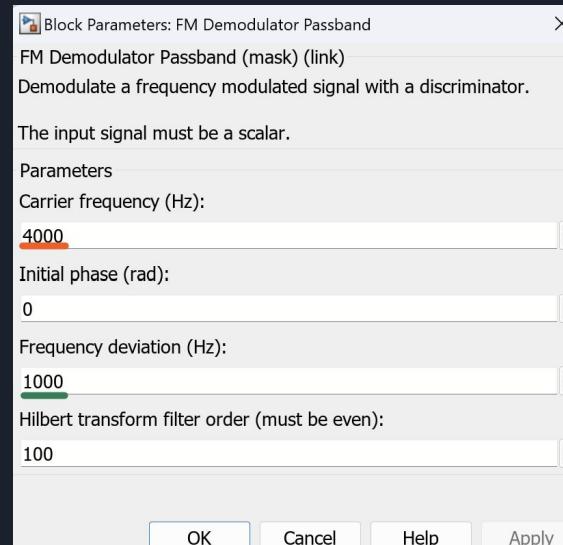
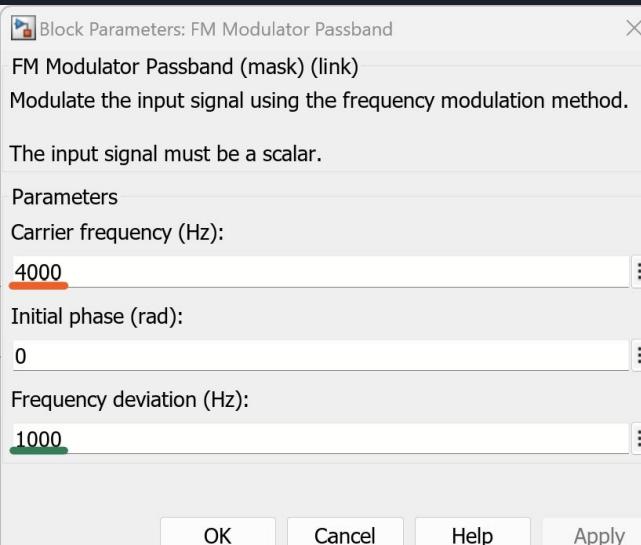
Etape 1 : Simulation d'une modulation puis démodulation

On commence par créer le modèle conformément aux instructions :



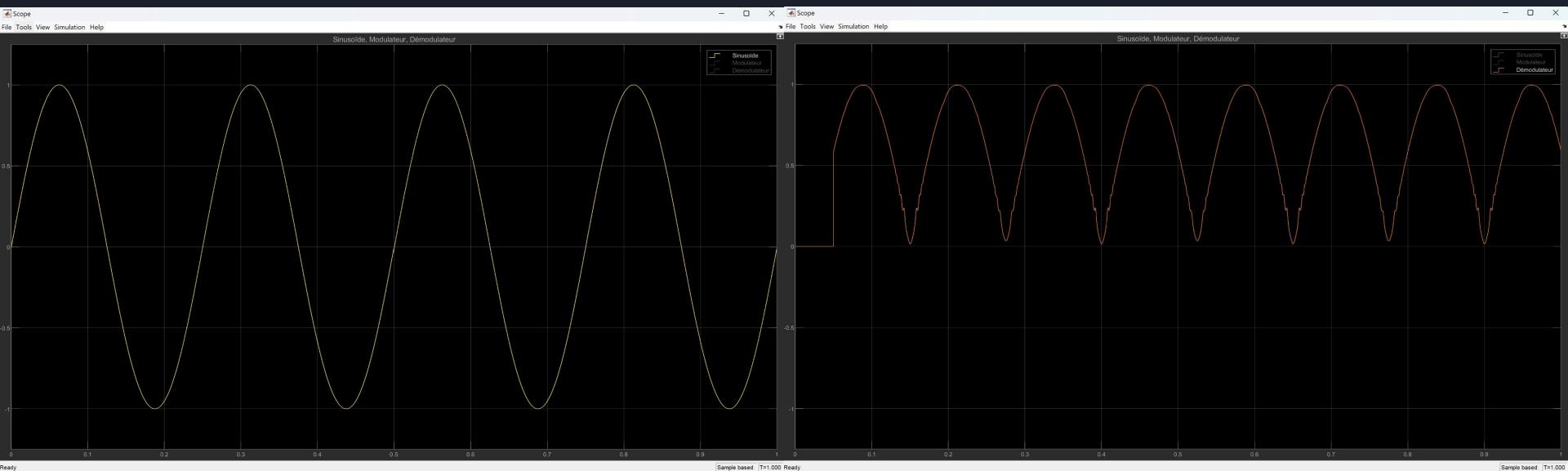
Etape 1 : Simulation d'une modulation puis démodulation

Puis on configure correctement la **fréquence de la sinusoïde à 4 Hz**, la **fréquence d'échantillonnage à 2 kHz** la **fréquence de la porteuse comme étant de 4 kHz** et l'**excursion de fréquence comme étant de 1 kHz**



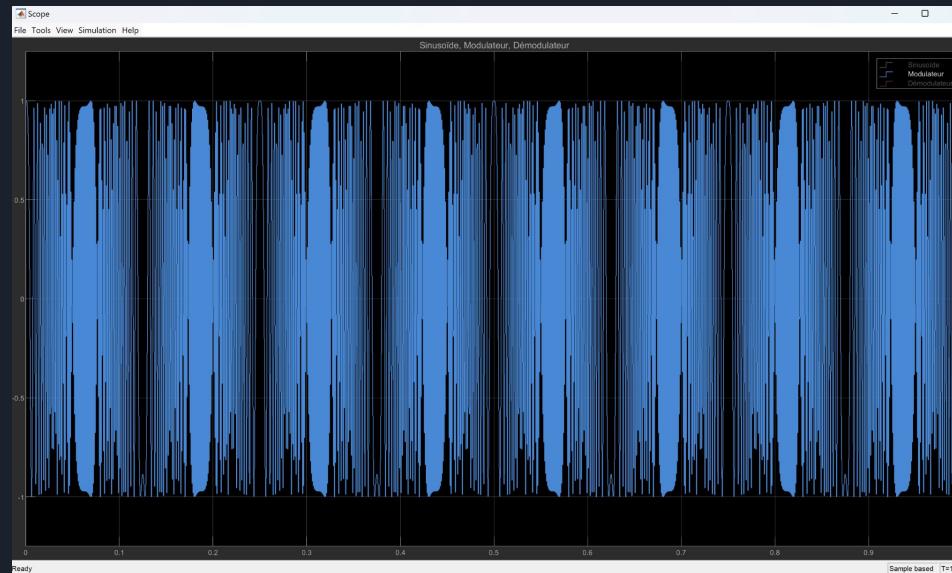
Etape 1 : Simulation d'une modulation puis démodulation

On retrouve donc logiquement la **sinusoïde d'entrée** du système modulateur-démodulateur et la **sinusoïde de sortie** du système modulateur-démodulateur, on peut d'ailleurs observer une modification de la fréquence et de l'amplitude qui sont toutes normales car le système modifie ces valeurs puis ne les ramène que partiellement sur ce qu'elles étaient de base, le déphasage lui aussi est normal puisque le signal subit un délai de propagation entre chaque transformation :



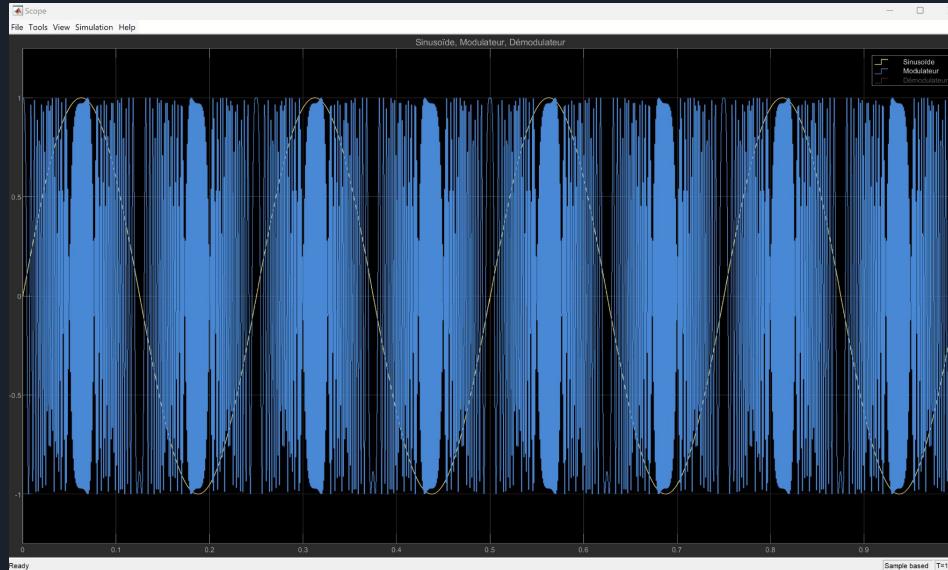
Test de l'Etape 1 : Caractérisation de notre signal au travers des blocs

Le premier signal que l'on va caractériser est celui ci-dessous, celui émit par le modulateur en direction du démodulateur appelé D* :



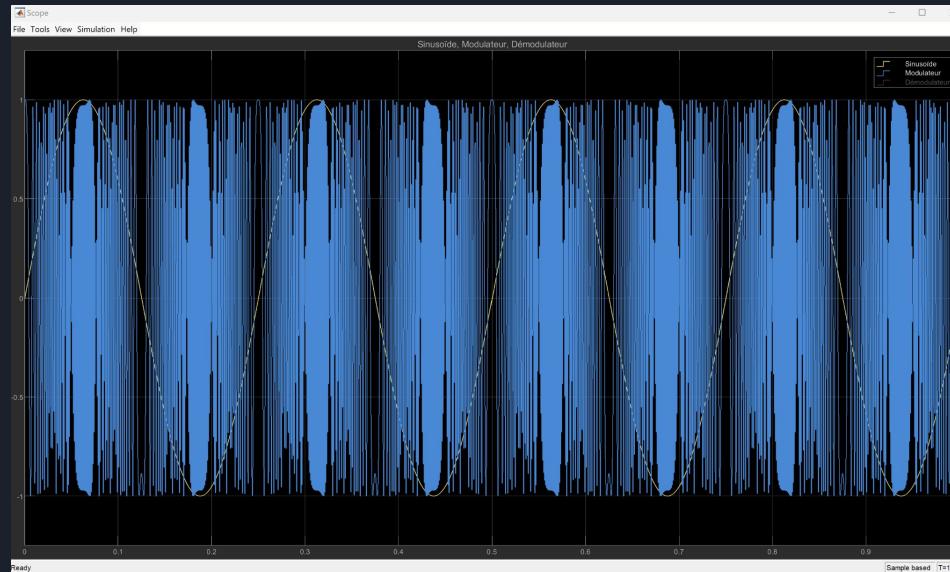
Test de l'Etape 1 : Caractérisation de notre signal au travers des blocs

Notre signal D* possède la même amplitude que notre signal sinusoïdal d'entrée qui est donc de 1 V pour une plage d'amplitude allant de [1;-1] V :



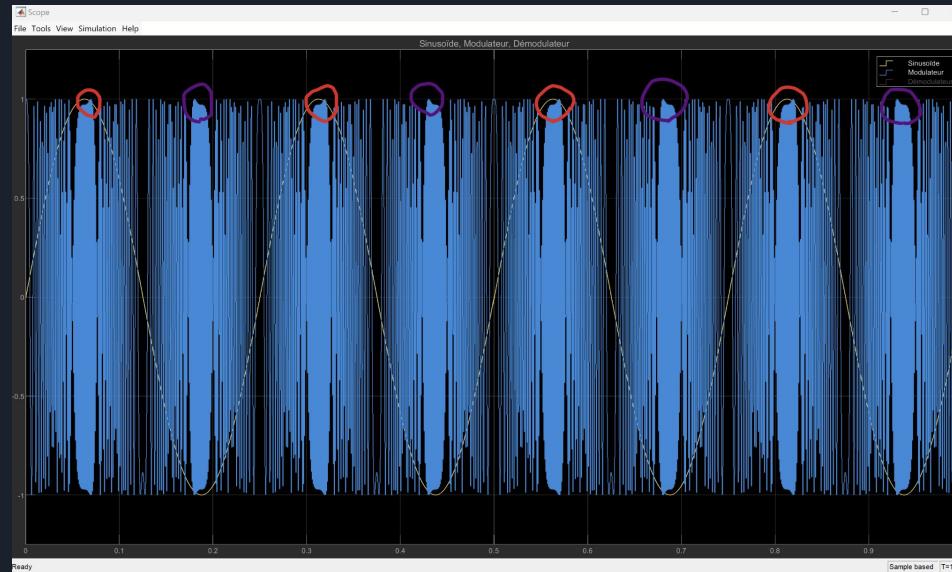
Test de l'Etape 1 : Caractérisation de notre signal au travers des blocs

L'on voit bien que chaque période du signal sinusoïdal d'entrée correspond à une période du signal D* qui a donc une fréquence de 4 Hz :



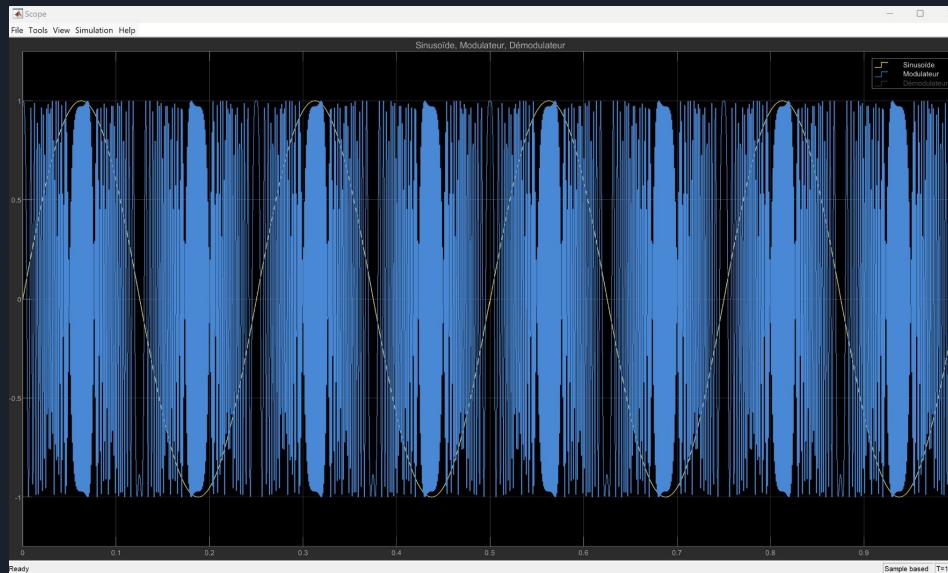
Test de l'Etape 1 : Caractérisation de notre signal au travers des blocs

L'on pourrait croire qu'il y a 8 périodes et non 4 mais c'est faux et les éléments entourés en rouge sont différents de ceux entourés en violet :



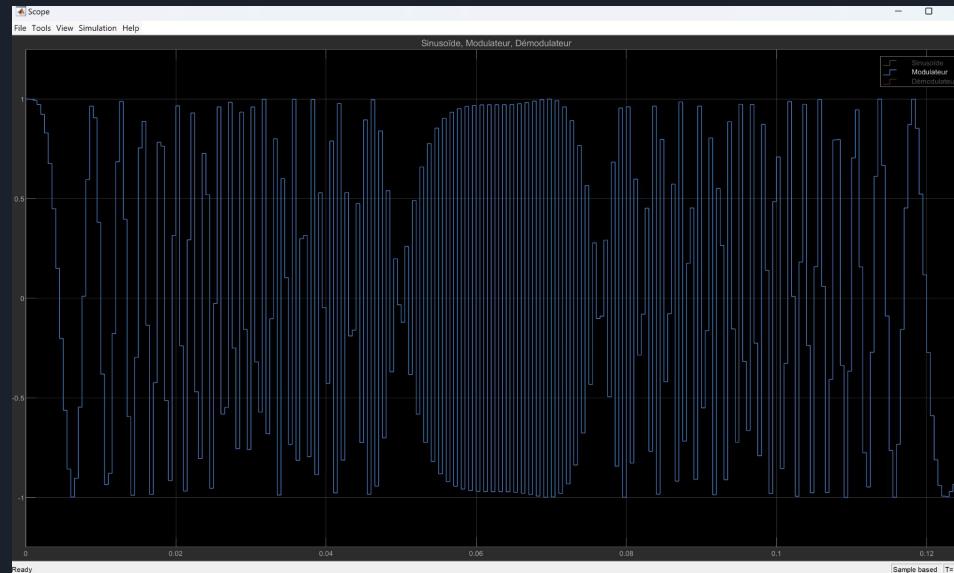
Test de l'Etape 1 : Caractérisation de notre signal au travers des blocs

L'on remarque également que le signal D^* atteint des valeurs proches de 1 et -1 V beaucoup plus souvent lorsque que le signal sinusoïdal d'entrée a une valeur proche de 1 et -1 V et, à l'inverse, atteint des valeurs proche de 1 et -1 V beaucoup moins souvent lorsque que le signal sinusoïdal d'entrée à une valeur proche de 0 V. C'est ce qui donne l'impression erronée que la fréquence du signal D^* est plus élevée pour les valeurs proches de 1 et -1 V et moins élevée pour les valeurs proches de 0 V :



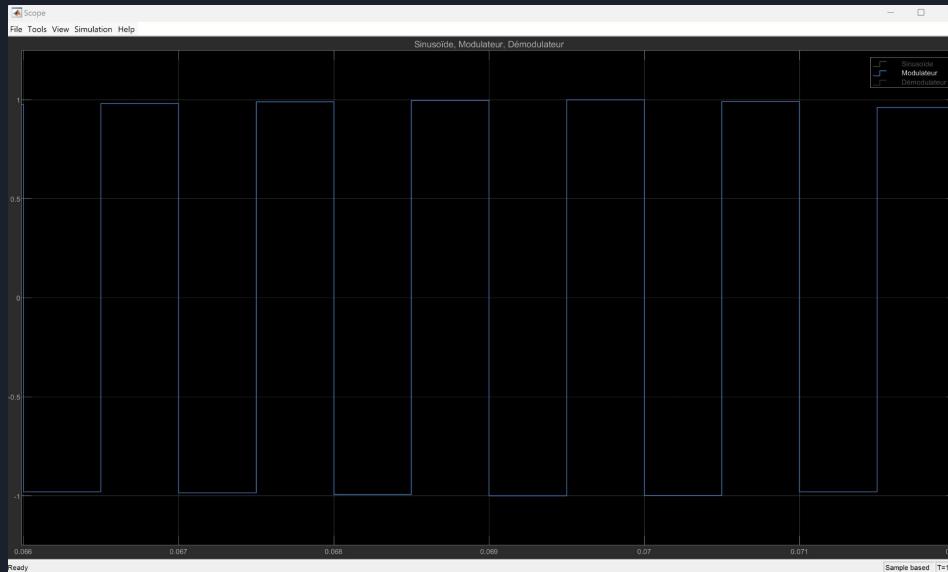
Test de l'Etape 1 : Caractérisation de notre signal au travers des blocs

Voilà à quoi ressemble une demi-période du signal D* :



Test de l'Etape 1 : Caractérisation de notre signal au travers des blocs

En zoomant sur les hautes amplitudes pour avoir une meilleure visibilité, on voit bien qu'un échantillon mesure 0,0005 s soit 0,5 ms, on obtient donc une fréquence d'échantillonnage que l'on va nommer F_e de $1/0,0005 = 2$ kHz, un résultat cohérent :



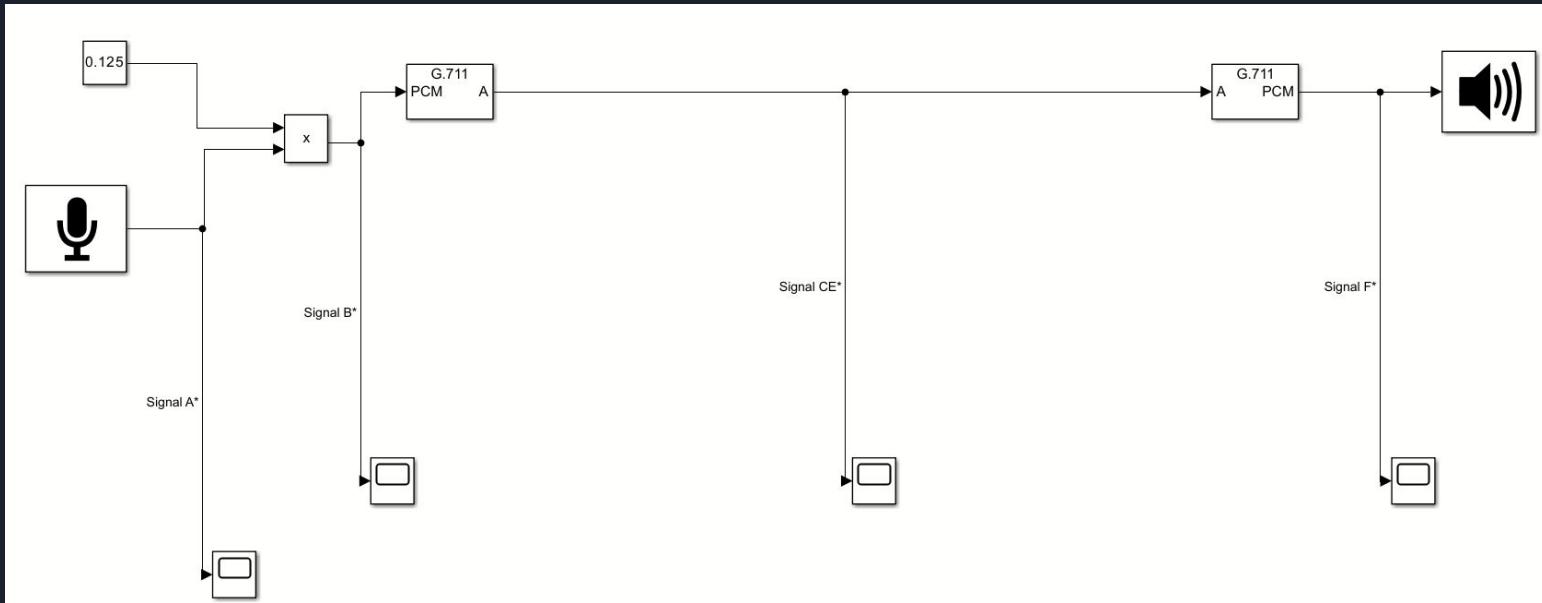


Test de l'Etape 1 : Caractérisation de notre signal au travers des blocs

Pour résumer, le signal D* est un signal périodique de fréquence 4 Hz, d'amplitude crête à crête 2 V et de fréquence d'échantillonnage 2 kHz, en bref tout comme notre signal sinusoïdal d'entrée à la différence que c'est un signal numérique et non analogique comme le signal sinusoïdal d'entrée. De plus la forme n'est clairement pas celle d'une sinusoïde.

Etape 2 : Création du modèle final

On commence par changer de topologie pour celle ci-dessous, on a une sinusoïde et une constante dont le produit sera envoyé dans un codec configuré en encodeur puis un codec configuré en décodeur. L'haut parleur sert à tester le fonctionnement :



Etape 2 : Crédit du modèle final

Observons en détail les configurations de nos différents blocs en commençant par le bloc micro :

The screenshot shows two instances of the 'Block Parameters: Audio Device Reader' dialog box. Both dialogs have the same title bar and basic configuration settings.

Title Bar: Block Parameters: Audio Device Reader

Description: Record audio stream from your computer's audio device.

Main Tab: Selected tab.

Parameters:

- Driver: DirectSound
- Device: Default
- Sample rate (Hz): 44100
- Outputs
 - Number of channels: 1
 - Samples per frame: 256

Advanced Tab: Not selected.

Right Dialog (Advanced Tab):

- Device bit depth: 16-bit integer
- Use default channel mapping
- Outputs
 - Output number of samples overrun
- Output data type: int16

Buttons at the bottom: OK, Cancel, Help, Apply.

Etape 2 : Création du modèle final

Puis le bloc de la constante 0,125 :

Block Parameters: Constant

Constant

Output the constant specified by the 'Constant value' parameter. If 'Constant value' is a vector and 'Interpret vector parameters as 1-D' is on, treat the constant value as a 1-D array. Otherwise, output a matrix with the same dimensions as the constant value.

Main Signal Attributes

Constant value:

0.125

Interpret vector parameters as 1-D

Sample time:

inf

X Block Parameters: Constant

Constant

Output the constant specified by the 'Constant value' parameter. If 'Constant value' is a vector and 'Interpret vector parameters as 1-D' is on, treat the constant value as a 1-D array. Otherwise, output a matrix with the same dimensions as the constant value.

Main Signal Attributes

Output minimum: []

Output maximum: []

Output data type: Inherit: Inherit from 'Constant value'

Lock output data type setting against changes by the fixed-point tools

OK Cancel Help Apply

OK Cancel Help Apply

Etape 2 : Création du modèle final

Puis le bloc produit :

The image shows two identical 'Block Parameters: Product' dialog boxes side-by-side, illustrating the configuration of a Product block in a modeling environment.

Main Tab Content:

- Product**: The block type is set to 'Product'.
- Description**: 'Multiply or divide inputs. Choose element-wise or matrix product and specify one of the following:'
 - a) * or / for each input port. For example, **/* performs the operation ' $u1*u2/u3*u4$ '.
 - b) A scalar value specifies the number of input ports to be multiplied. For example, 2 performs the operation ' $u1*u2$ '.
- Notes**: If there is only one input port and the Multiplication parameter is set to Element-wise(.*) , a single * or / collapses the input signal using the specified operation. However, if the Multiplication parameter is set to Matrix(*), a single * causes the block to output the matrix unchanged, and a single / causes the block to output the matrix inverse.

Main Tab Options:

- Number of inputs:** Set to 2.
- Multiplication:** Set to Element-wise(.*) .

Signal Attributes Tab Options:

- Require all inputs to have the same data type:** Unchecked.
- Output minimum:** []
- Output maximum:** []
- Output data type:** int16
- Lock output data type setting against changes by the fixed-point tools:** Unchecked.
- Integer rounding mode:** Floor
- Saturate on integer overflow:** Unchecked.

Buttons at the bottom:

- OK, Cancel, Help, Apply

Etape 2 : Création du modèle final

Puis le premier bloc codec configuré en encodeur :

Block Parameters: G.711 Codec

G.711 Codec (mask) (link)

Implements the ITU-T G.711 recommendation for encoding, decoding, or converting speech signals.

The block encodes int16 PCM signals using A-law or mu-law into uint8 codewords. The input is assumed to be a 13-bit (A-law) or 14-bit (mu-law) PCM signal.

The block decodes uint8 codewords into 13-bit (A-law) or 14-bit (mu-law) PCM signals of type int16.

The block converts between A-law and mu-law uint8 codewords.

Parameters

Mode: Encode PCM to A-law

Overflow diagnostic: Warning

OK Cancel Help Apply

Etape 2 : Création du modèle final

Puis le second bloc codec configuré en décodeur : G.711 Codec (mask) (link)

Block Parameters: G.711 Codec1 X

G.711 Codec (mask) (link)

Implements the ITU-T G.711 recommendation for encoding, decoding, or converting speech signals.

The block encodes int16 PCM signals using A-law or mu-law into uint8 codewords. The input is assumed to be a 13-bit (A-law) or 14-bit (mu-law) PCM signal.

The block decodes uint8 codewords into 13-bit (A-law) or 14-bit (mu-law) PCM signals of type int16.

The block converts between A-law and mu-law uint8 codewords.

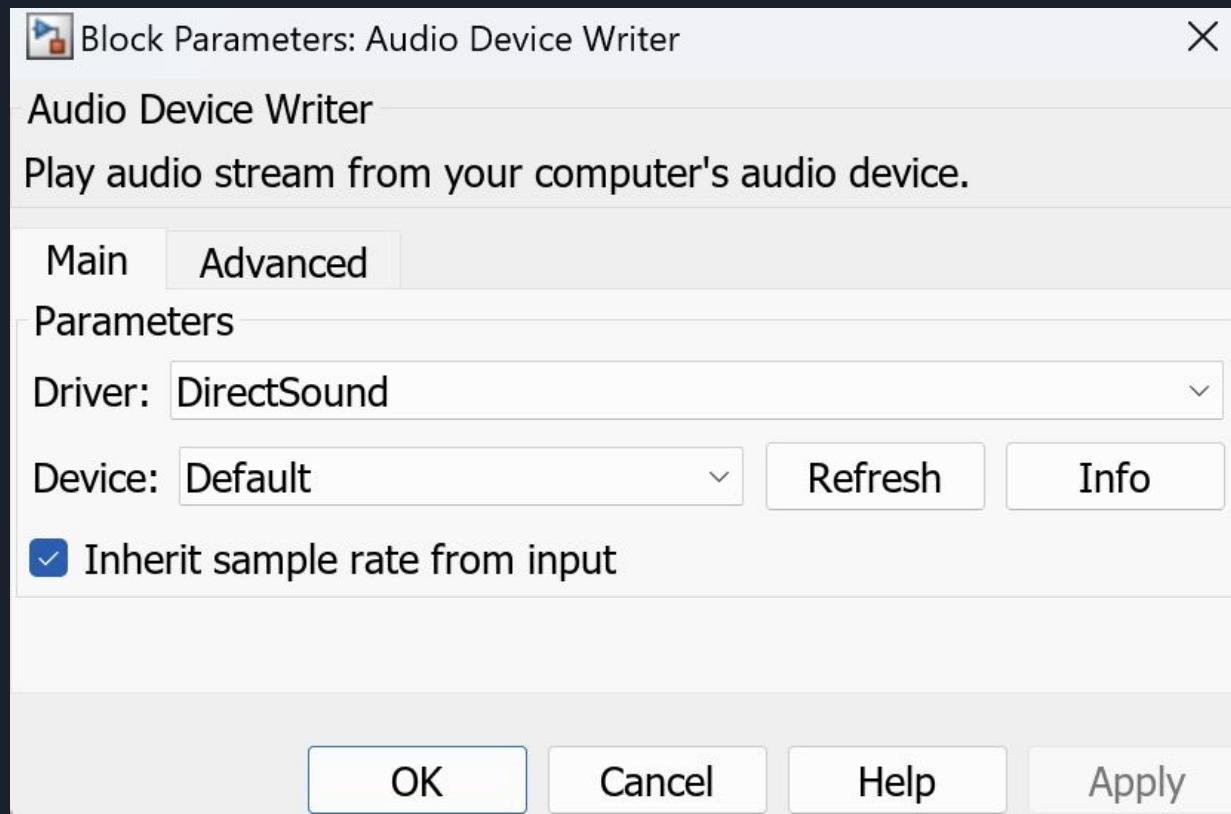
Parameters

Mode: Decode A-law to PCM ▼

OK Cancel Help Apply

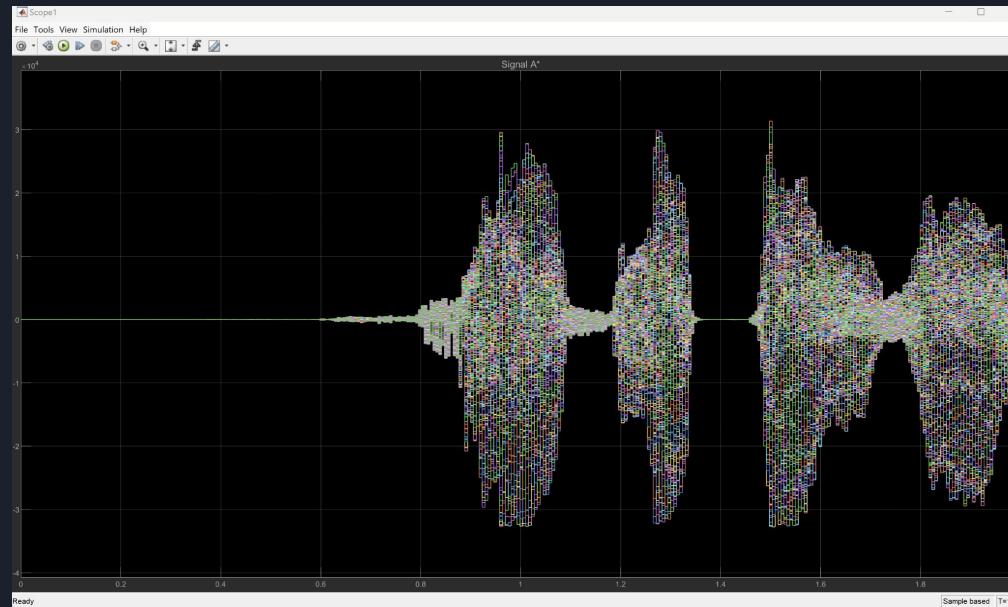
Etape 2 : Création du modèle final

Et enfin, le bloc haut parleur :



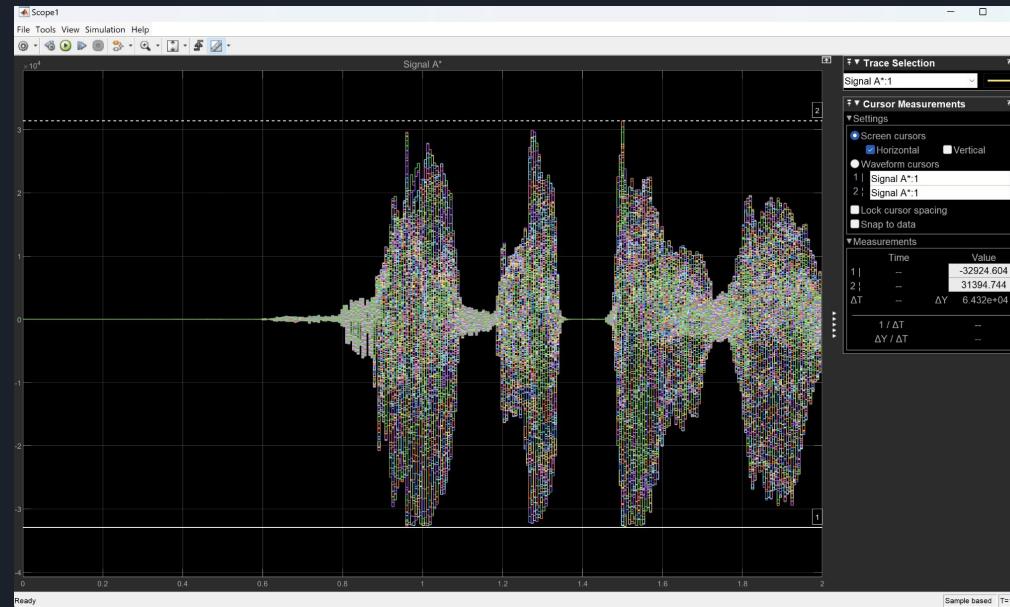
Test de l'Etape 2 : Caractérisation de notre signal au travers des blocs

Voici notre signal A*, le signal d'origine interprété par le microphone :



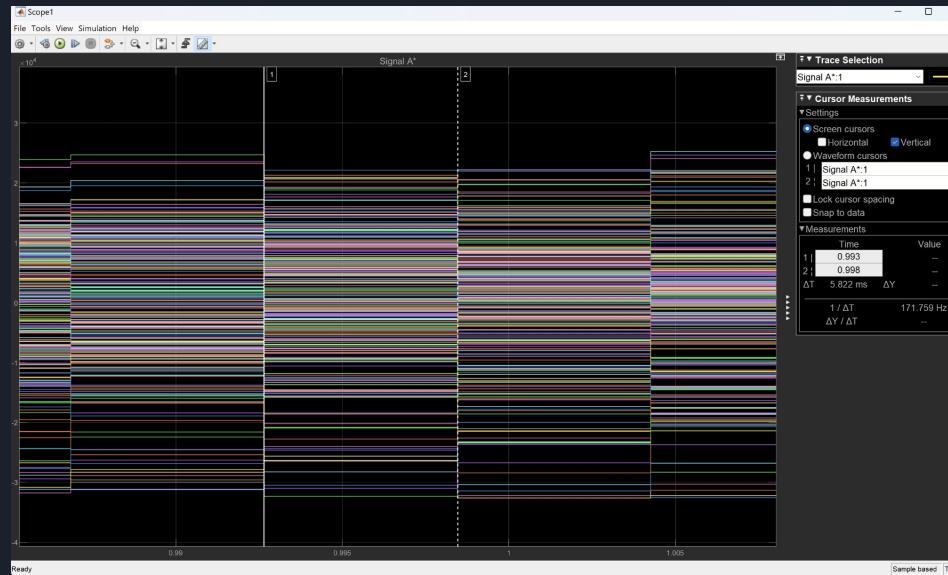
Test de l'Etape 2 : Caractérisation de notre signal au travers des blocs

On a une plage d'amplitudes allant de [31394;-32924] ce qui est équivalent à une amplitude crête à crête de 64318 :



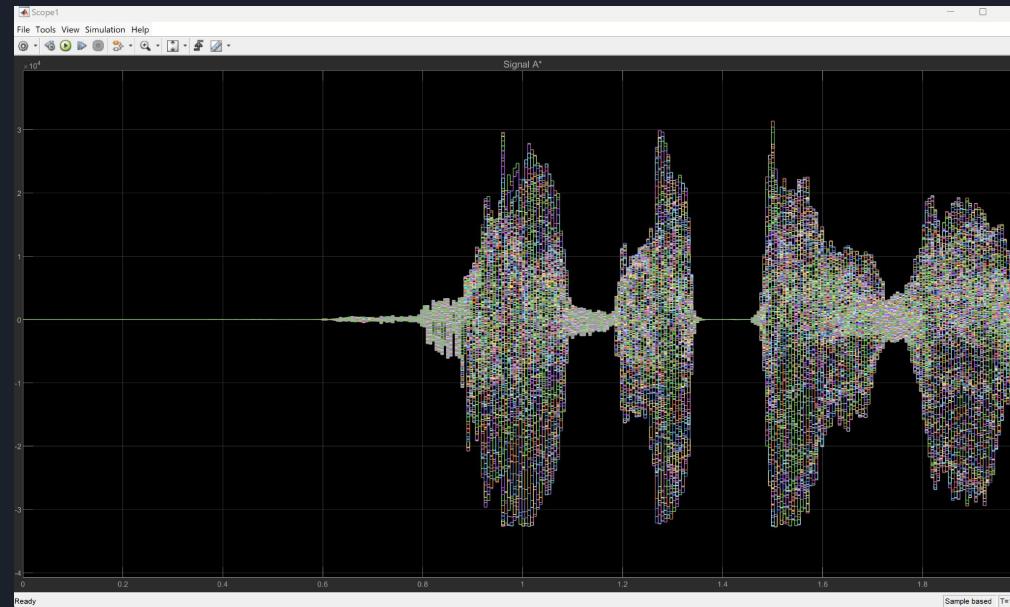
Test de l'Etape 2 : Caractérisation de notre signal au travers des blocs

On a une tranche de 256 traces, cette tranche a une durée ΔT de 5.822 ms. Cela nous donne donc une période d'échantillonnage de $\Delta T / 256 = (5.822 \times 10^{-3}) / 256 = 0.00002274218$. On a donc une fréquence d'échantillonnage de $1 / \text{période d'échantillonnage} = 1 / 0.00002274218 = 43971$ Hz ce qui est approximativement équivalent à notre fréquence d'échantillonnage de 44100 Hz :



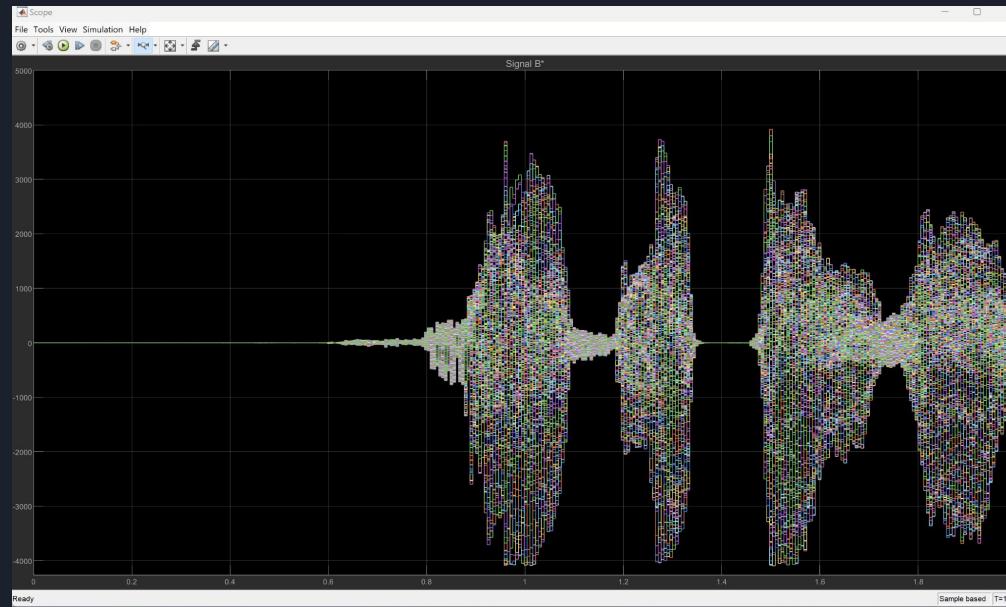
Test de l'Etape 2 : Caractérisation de notre signal au travers des blocs

Le signal A* est donc un signal analogique non périodique d'amplitude crête à crête de 64320 et de fréquence d'échantillonnage de 43971 Hz. De plus, chaque tranche contient 256 traces.



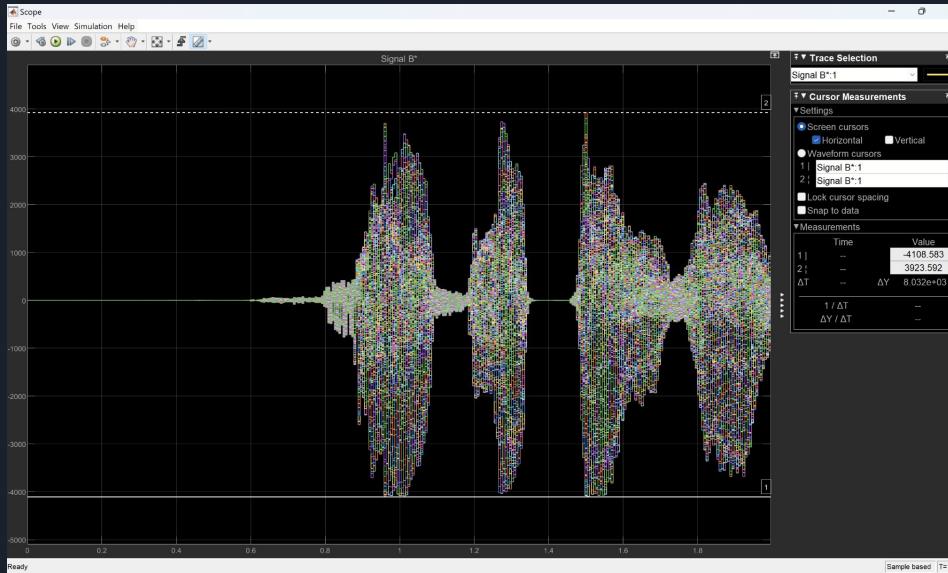
Test de l'Etape 2 : Caractérisation de notre signal au travers des blocs

Voici notre signal B^* , le produit du signal d'origine et de la constante 0.125 :



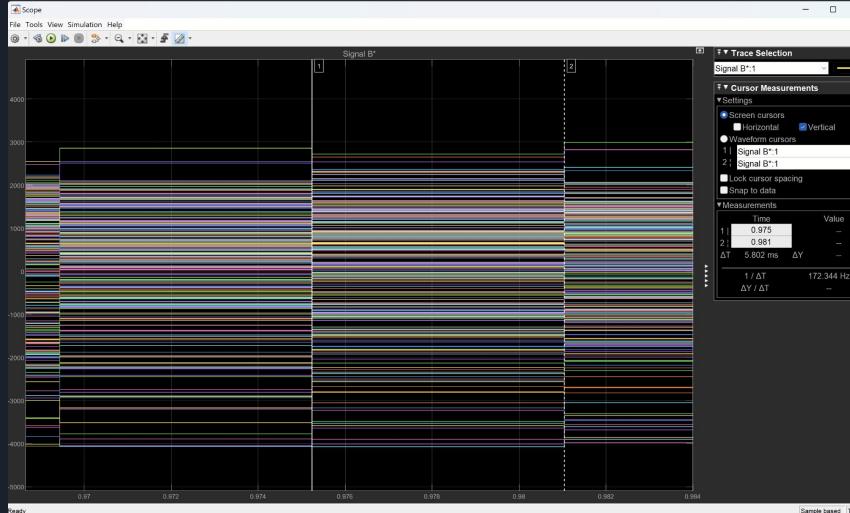
Test de l'Etape 2 : Caractérisation de notre signal au travers des blocs

On a une plage d'amplitudes allant de [3923;-4108] ce qui est équivalent à une amplitude crête à crête de 8032 :



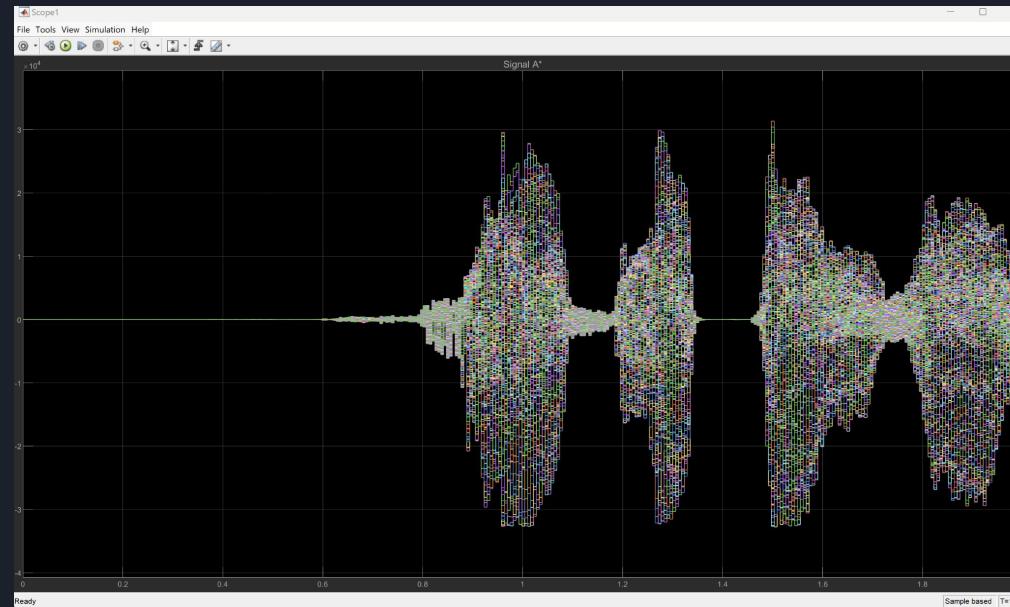
Test de l'Etape 2 : Caractérisation de notre signal au travers des blocs

On a une tranche de 256 traces, cette tranche a une durée DeltaT de 5.802 ms. Cela nous donne donc une période d'échantillonnage de $(5.802 \times 10^{-3}) / 256 = 0.00002266406$. On a donc une fréquence d'échantillonnage de $1 / 0.00002266406 = 44122$ Hz ce qui est approximativement équivalent à notre fréquence d'échantillonnage de 44100 Hz :



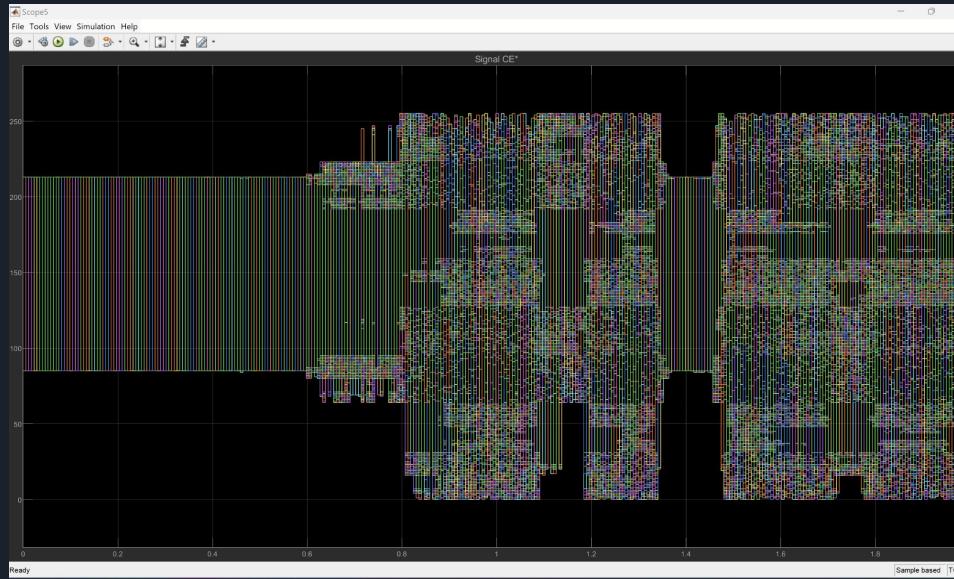
Test de l'Etape 2 : Caractérisation de notre signal au travers des blocs

Le signal B^* est donc un signal analogique non périodique d'amplitude crête à crête de 8032 et de fréquence d'échantillonnage de 44122 Hz. De plus, chaque tranche contient 256 traces.



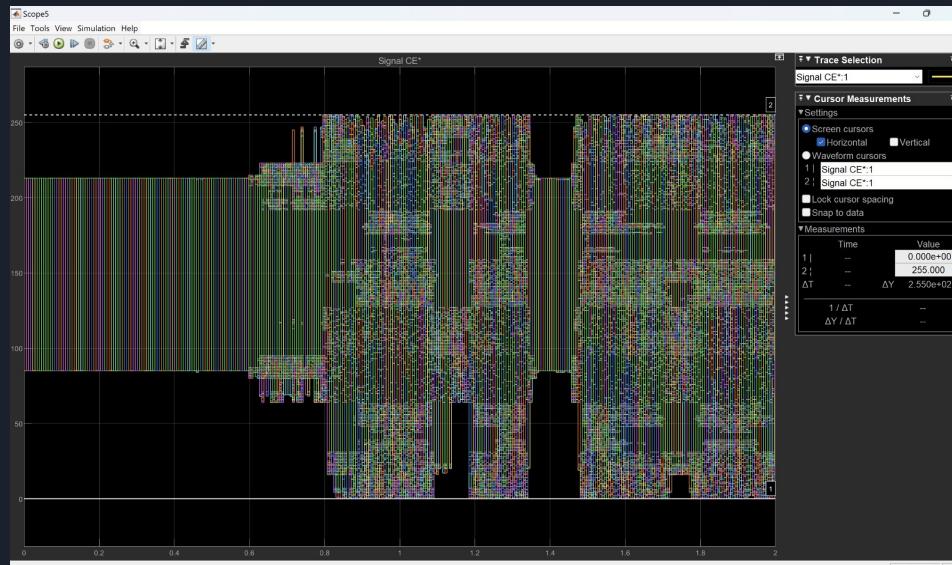
Test de l'Etape 2 : Caractérisation de notre signal au travers des blocs

Voici notre signal CE*, le signal de sortie du convertisseur configuré en tant que encodeur :



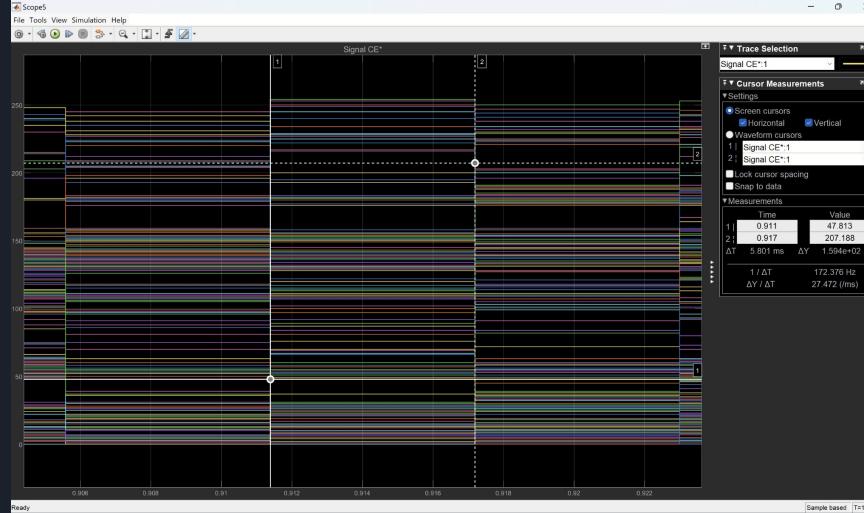
Test de l'Etape 2 : Caractérisation de notre signal au travers des blocs

On a une plage d'amplitudes allant de [0;255] ce qui est équivalent à une amplitude crête à crête de 255, en comptant 0, on a donc 256 valeurs possibles (N) par trace :



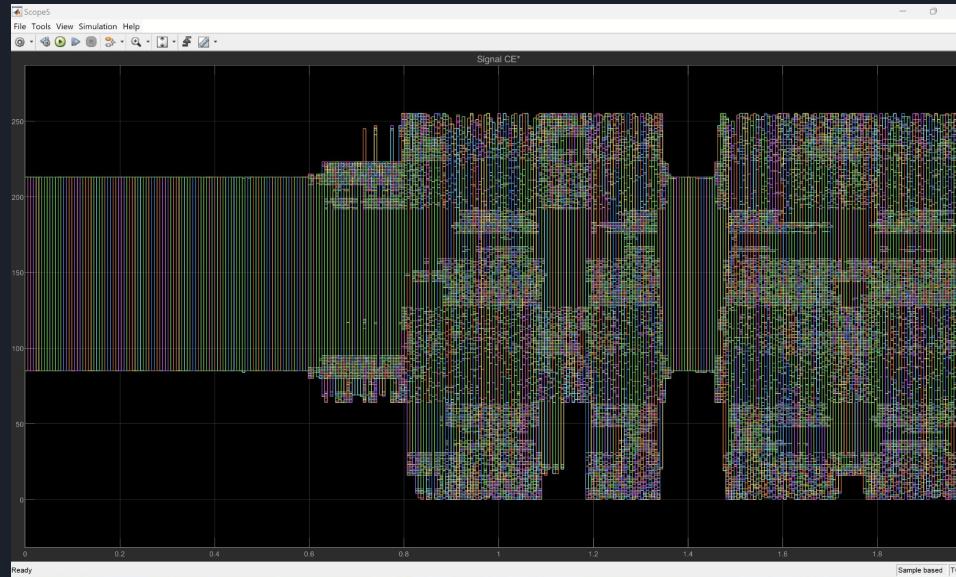
Test de l'Etape 2 : Caractérisation de notre signal au travers des blocs

On a une tranche de 256 traces, cette tranche a une durée DeltaT de 5.801 ms. Cela nous donne donc une période d'échantillonnage de $(5.801 \times 10^{-3}) / 256 = 0.00002266015$. On a donc une fréquence d'échantillonnage de $1 / 0.00002266015 = 44130$ Hz ce qui est approximativement équivalent à notre fréquence d'échantillonnage de 44100 Hz :



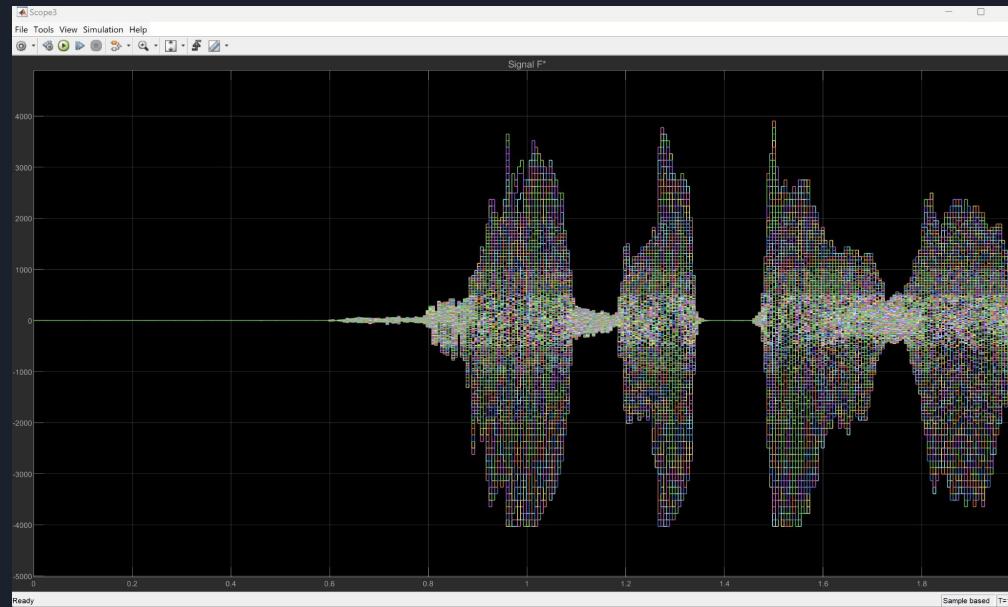
Test de l'Etape 2 : Caractérisation de notre signal au travers des blocs

Le signal CE* est donc un signal numérique non périodique d'amplitude crête à crête de 255 et de fréquence d'échantillonnage de 44130 Hz. De plus, chaque tranche contient 256 traces.



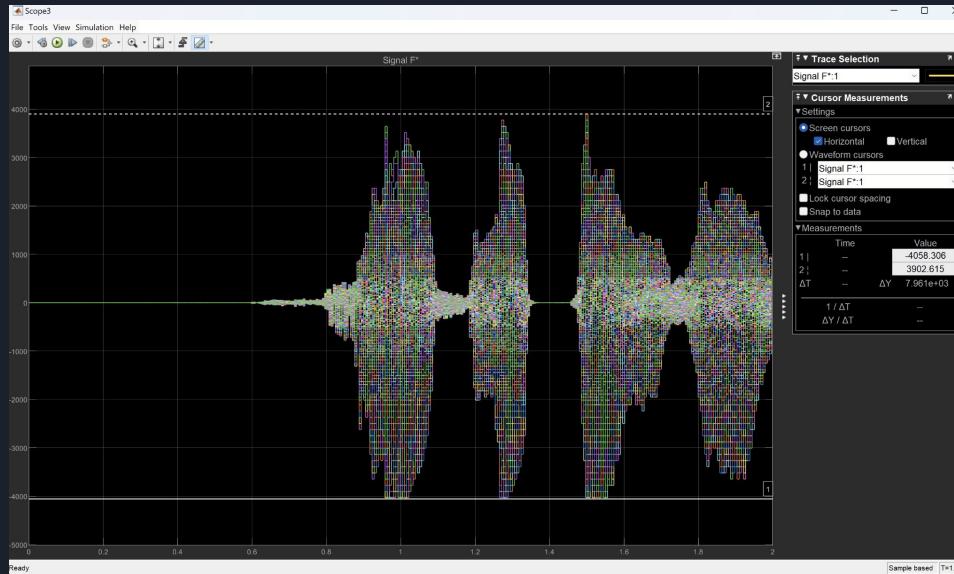
Test de l'Etape 2 : Caractérisation de notre signal au travers des blocs

Voici notre signal F^* , le signal de sortie du convertisseur configuré en tant que décodeur :



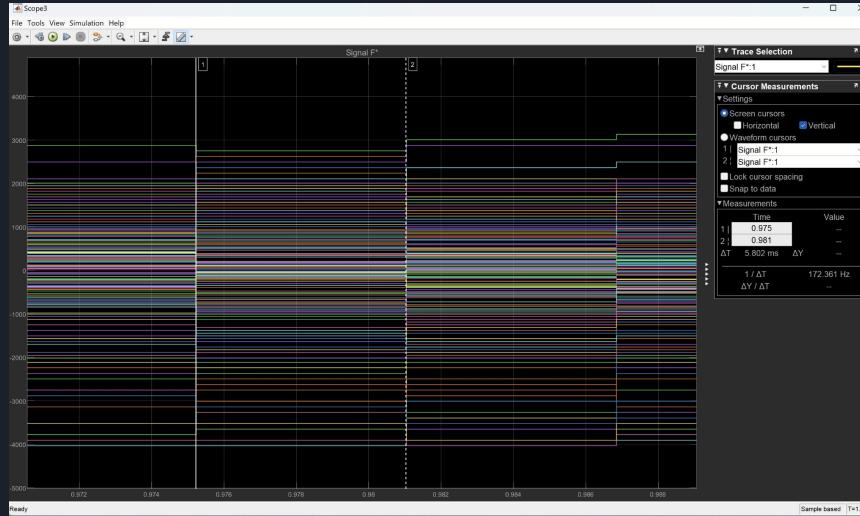
Test de l'Etape 2 : Caractérisation de notre signal au travers des blocs

On a une plage d'amplitudes allant de [3902;-4058] ce qui est équivalent à une amplitude crête à crête de 7960 :



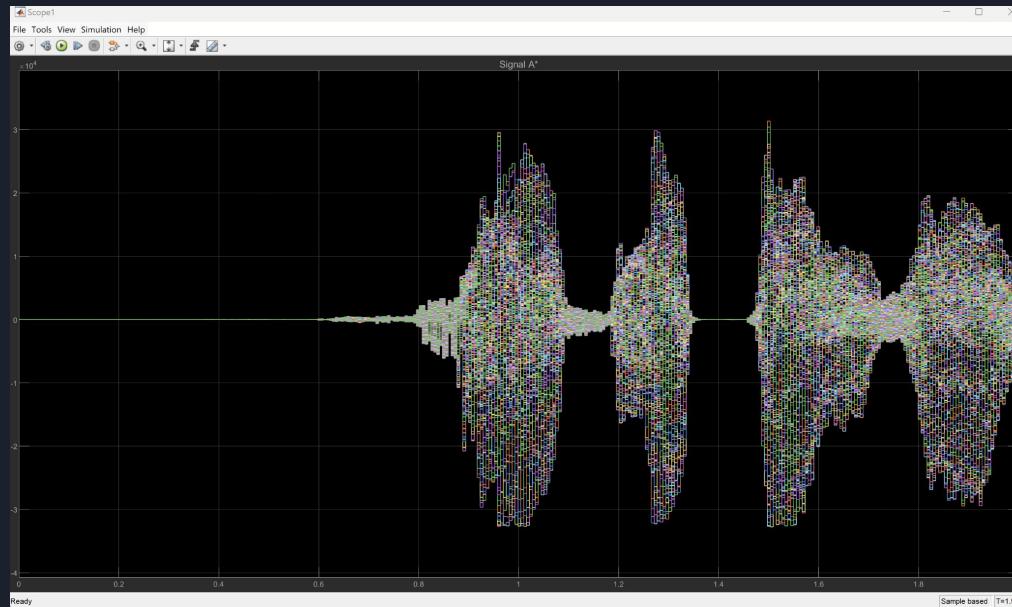
Test de l'Etape 2 : Caractérisation de notre signal au travers des blocs

On a une tranche de 256 traces, cette tranche a une durée DeltaT de 5.802 ms. Cela nous donne donc une période d'échantillonnage de $(5.802 \times 10^{-3}) / 256 = 0.00002266406$. On a donc une fréquence d'échantillonnage de $1 / 0.00002266406 = 44122$ Hz ce qui est approximativement équivalent à notre fréquence d'échantillonnage de 44100 Hz :



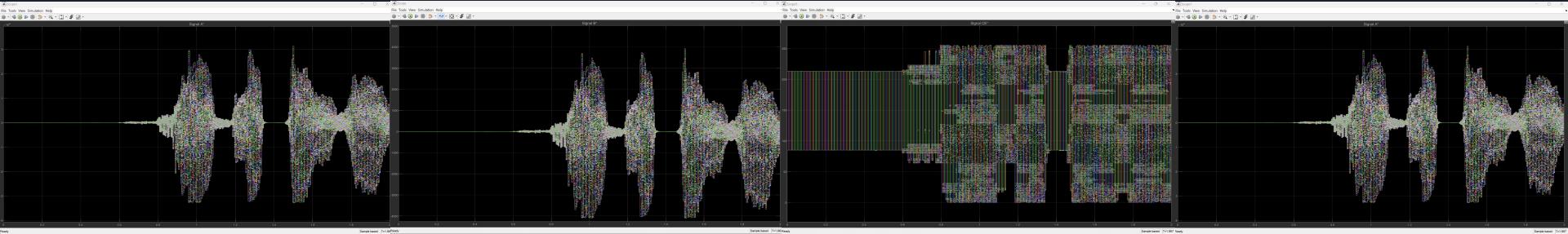
Test de l'Etape 2 : Caractérisation de notre signal au travers des blocs

Le signal F^* est donc un signal analogique non périodique d'amplitude crête à crête de 7691 et de fréquence d'échantillonnage de 44122 Hz. De plus, chaque tranche contient 256 traces. On note qu'il est quasiment identique au signal B^* ce qui est logique car un signal encodé puis décodé devrait rester presque identique :



Test de l'Etape 2 : Caractérisation de notre signal au travers des blocs

On a donc un signal A* qui est multiplié par une constante de valeur 0.125 qui est ensuite encodé et enfin décodé :





Test de l'Etape 2 : Caractérisation de notre signal au travers des blocs

Et voici le tableau récapitulatif des caractéristiques des signaux au travers des blocs :

	Signal A*	Signal B*	Signal CE*	Signal F*
Plage d'amplitudes	[31394;-32924]	[3923;-4108]	[0;255]	[3902;-4058]
Fréquence d'échantillonnage	43971 Hz	44122 Hz	44130 Hz	44122 Hz