

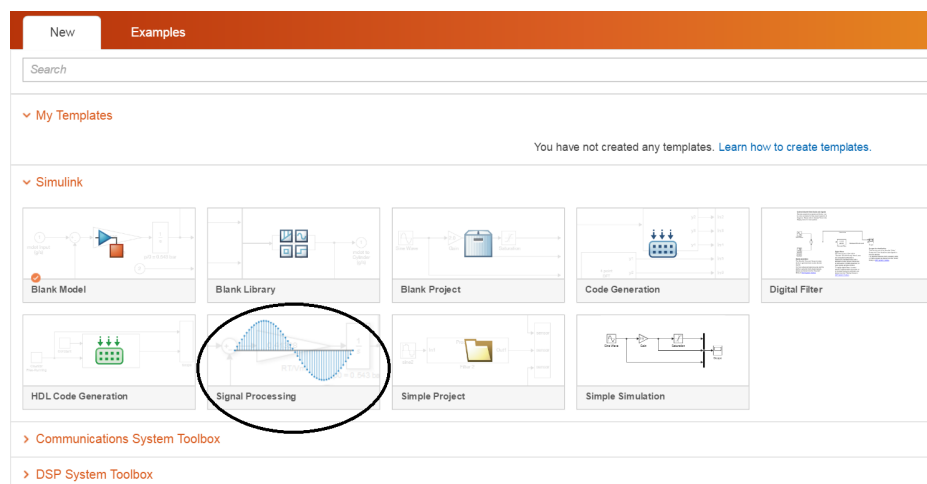
TP1 : USRP, Analyse spectrale et Modulation AM

Simulink : 3 bonnes pratiques à retenir

Lancer Matlab, puis Simulink (en tapant `simulink` dans la fenêtre de commande ou avec le bouton de la barre d'outils).

1. pour créer un nouveau fichier, **ouvrir un template**. En général, choisir type **Signal Processing**; ceci permet de régler par défaut Simulink pour les systèmes à temps discret¹. C'est indispensable quand on travaille avec l'USRP.

En ce qui nous concerne, en TC, dans certains cas particuliers seulement, quand on est amenés à simuler du temps continu, on configure pour des systèmes à temps continu.

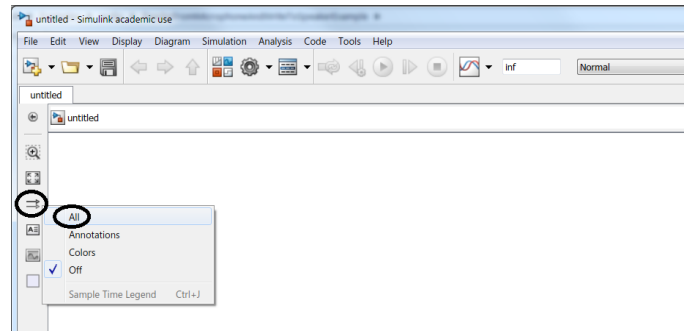


Pour ouvrir un nouveau modèle depuis les menus, faire **File > New** et ensuite **Model**, ET PAS **Blank Model**, pour avoir accès au template.

On construit ensuite un modèle en connectant des blocs qu'on trouve dans la bibliothèque. On accède à cette dernière depuis le menu **View > Library Browser**, ou avec **CTRL+Shift+L** (ou encore, l'icône correspondante dans la barre d'outils). Si l'on connaît le nom, on peut simplement faire un double-clic sur le modèle (dans le blanc) et taper le nom du bloc recherché.

2. **Toujours faire afficher sur votre modèle les couleurs** correspondant aux sample times.

1. Plus précisément, on choisit ainsi un *solver*. Qu'est-ce que le solver? Le mieux est encore de consulter l'aide de Matlab : "To simulate a dynamic system, you compute its states at successive time steps over a specified time span. This computation uses information provided by a model of the system. Time steps are time intervals when the computation happens. The size of this time interval is called step size. The process of computing the states of a model in this manner is known as solving the model. No single method of solving a model applies to all systems. Simulink® provides a set of programs called solvers. Each solver embodies a particular approach to solving a model."



Une fois que le modèle est construit, pour afficher les couleurs et vérifier si le modèle ne contient pas d'erreurs, faire **CTRL+E** (Update Diagram); et pour avoir la légende, faire **CTRL+J**.

3. **Toujours faire afficher les dimensions et types des signaux de votre modèle**; pour cela, menu **Display > Signals & Ports > Signal Dimensions** et **Display > Signals & Ports > Port Data Types**. Une fois que le modèle est construit, pour afficher les types et dimensions, faire **CTRL+E** (Update Diagram).

1 Transmission AM (simulation)

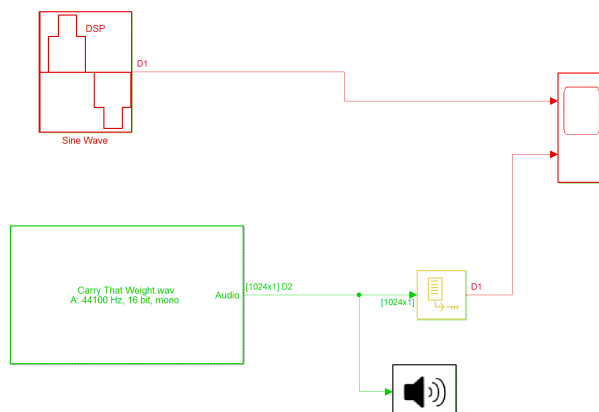
1.1 Les sources

Créer un nouveau fichier (avec le template DSP). On vous demande d'utiliser deux sources :

- une sinusoïde, de fréquence 2 kHz; pour cela utiliser le bloc **Sine Wave** que vous trouverez dans la **DSP System Toolbox > Sources**. Attention il y a plusieurs sources générant des sinusoïdes, il faut bien prendre celle de cette toolbox. Régler sa fréquence, puis le **Sample time** afin qu'elle soit échantillonnée à 44.1 kHz;
- un signal audio lu depuis un fichier .wav, que vous trouverez sous Moodle : Pour pouvoir le lire, chercher le bloc **From Multimedia File**. Dans les paramètres, il faut donner le chemin du fichier. Puis tirer un fil à la sortie, et faire **CTRL+E**. Quelle est la taille des données? Est-ce du mono ou du stereo? Faire **CTRL+J** et justifier la valeur de la fréquence d'échantillonnage.

Vous, pouvez ensuite ajouter un bloc **Audio Device Writer** pour jouer le son.

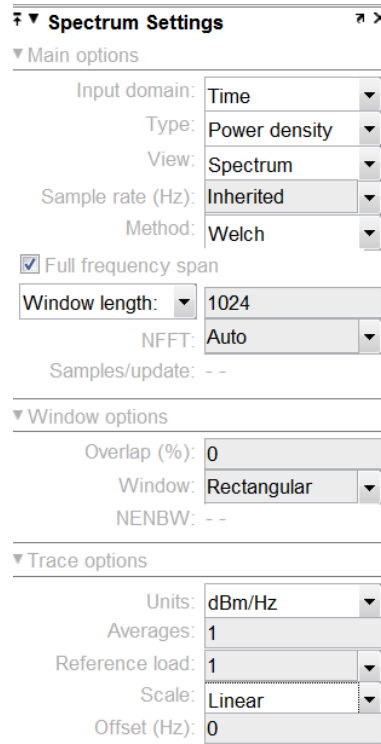
Il faut connaître l'amplitude du signal audio : vous en aurez besoin pour régler l'indice de modulation. Pour cela, mettre un bloc **Unbuffer** à la sortie pour mettre les échantillons en série; puis un bloc **scope**.



Simuler quelques secondes. Relever la valeur maximale atteinte.

L'analyse spectrale sous simulink

Ajouter un bloc Spectrum Analyser (toolbox DSP System > Sinks). Ouvrir l'analyseur en double-cliquant, et faire View > Spectrum Settings. Le mieux est de commencer par mettre les réglages ci-dessous :



Ce bloc calcule la *Densité Spectrale de Puissance* (DSP) d'un signal qui se trouve sur son entrée.

Soit $x(t)$ un signal à temps continu, déterministe à *énergie finie* ; cela signifie que son support temporel, qu'on peut appeler T , est borné : le signal est nul, puis non nul pendant une durée T , puis nul à nouveau, pour toujours.

Dans ce cas, sa transformée de Fourier $X(f)$ existe², et représente la *répartition de l'amplitude et de la phase* en fonction de la fréquence. Le carré du module de $X(f)$, noté $|X(f)|^2$, représente la *répartition de l'énergie* en fonction de la fréquence. Enfin, si l'on divise par T , on obtient la *répartition de la puissance* en fonction de la fréquence : c'est la DSP. On la note $S_{xx}(f)$. On a donc, pour $x(t)$ déterministe, à énergie finie :

$$S_{xx}(f) = \frac{|X(f)|^2}{T} \quad (1)$$

Si l'on fait le même raisonnement, mais en partant d'un signal à temps discret à énergie finie, $x(n)$, contenant N échantillons, on obtient :

$$S_{xx}(f) = \frac{|X(f)|^2}{N} \quad (2)$$

$X(f)$ est cette fois la transformée de Fourier à temps discret de $x(n)$.

2. Pour quelle existe, il faut que l'intégrale converge $\forall f$

Dans notre cas, comment le bloc **Spectrum Analyser** calcule-t-il la DSP ?

L'entrée de ce bloc est forcément un signal à temps discret, $x(n)$ (simulink n'accepte pas de temps continu à l'entrée du bloc). Le calcul est donc forcément le (2) ; à une différence près : le spectre $X(f)$ ne peut être calculé que par transformée de Fourier discrète, c'est-à-dire en pratique par FFT.

Rappel : les différentes Transformées de Fourier

Signal à temps continu : on calcule une transformée de Fourier à temps continu ; on obtient $X(f)$, défini pour toutes les valeurs de f

Signal à temps discret : on calcule

— une transformée de Fourier à temps discret ; on obtient $X(f)$, défini pour toutes les valeurs de $f \in [-f_e/2, f_e/2]$;

— ou une transformée de Fourier discrète ; on obtient $X(k\Delta f)$, défini pour

$$k\Delta f \in [-f_e/2, f_e/2]$$

Seule la seconde est réalisable par un ordinateur numérique (sans quoi on obtiendrait une infinité de valeurs, qu'il faudrait stocker) ; un algorithme permet d'accélérer le calcul quand le nombre d'échantillons est une puissance de 2, et on parle alors de FFT (Fast Fourier Transform).

Ce qu'il faut avoir en tête, pour le réglage de l'analyseur, c'est que le spectre est calculé sur la bande $[-f_e/2, f_e/2]$. La fréquence d'échantillonnage f_e étant fixée par le signal d'entrée à analyser (ici : 44.1 kHz), on fait varier l'écart entre deux points, Δf , par le nombre de points de la FFT, noté N_{FFT} .

En général, N et N_{FFT} sont égaux. On fixe N par la **Window length** et on met N_{FFT} sur **Auto** pour qu'il soit identique.

On choisit donc N plutôt grand, pour que la résolution soit acceptable. Une valeur de départ peut être 1024 ; si le spectre est vraiment mal résolu, essayer, 2048, 4096, etc.

Un autre réglage intéressant à connaître est celui de la *fenêtre*. Avec une fenêtre rectangulaire comme dans la capture ci-dessus, on n'a en fait *pas de fenêtre*, c'est-à-dire qu'on calcule réellement la FFT du signal. Mais le fait de tronquer le signal analysé sur N_{FFT} échantillons va induire une déformation du spectre. L'utilisation d'une fenêtre – par exemple la fenêtre de Hanning (hann) – permet en général d'améliorer les résultats. Une bonne explication de ce phénomène se trouve à l'adresse :

<https://fr.tek.com/blog/window-functions-spectrum-analyzers>

Enfin, si l'on analyse pas un signal déterministe, mais un processus aléatoire, on ne calcule que la DSP d'une *réalisation*. Pour obtenir la DSP, il faut en prendre l'espérance ; en pratique, on moyenne plusieurs traces avec le paramètre **Averages** (au moins 5)) pour obtenir un spectre plus stable.

Observer le spectre des deux signaux. Est-ce le résultat attendu ? Quelle est la bande occupée par le signal audio ?

1.2 La modulation

On sait que pour transmettre un signal par modulation d'amplitude, il faut utiliser une porteuse à une fréquence nettement plus élevée que le signal à transmettre (modulant). La modulation d'amplitude se fait simplement par multiplication.

Regarder la vidéo 1, puis ouvrir le fichier `AM_simulation`. Au niveau de la source permettant de lire le fichier audio, penser à modifier le chemin. Compléter le cadre à l'aide des blocs `Gain`, `Constant`, `Sum`, afin d'obtenir un signal modulé en amplitude, double bande avec porteuse, d'indice 0,5.

Valider votre modèle en observant le scope de gauche : le signal doit être toujours positif (indice inférieur à 1).

Une fois validé, décommenter (clic droit > `Uncomment`) les blocs qui simulent l'USRP, pour faire la multiplication. A partir de là, la simulation sera beaucoup plus lente (simulink doit traiter 500k échantillons par seconde).

Simuler pour obtenir quelques secondes de temps simulé et observer le scope de droite ; zoomer pour mettre en évidence la modulation (autrement dit, il faut voir la porteuse sur la voie du bas).

1.3 Analyse spectrale du signal modulé

Décommenter, en sortie, le bloc `Spectrum Analyser`.

Chercher à obtenir le spectre le plus satisfaisant, pour les deux entrées, et noter vos réglages. Est-ce l'allure attendue ?

Dans le cas où le modulant est le signal audio, relever approximativement les fréquences minimale et maximale, et la bande occupée.

1.4 L'USRP en transmission

On remplace l'encadré de droite par un bloc permettant de configurer l'USRP. Bien sûr, vous ne pouvez pas le faire ici.

La seule différence notable avec la simulation concerne la conversion analogique numérique. Le composant qui s'occupe de cela est un DAC (Digital to Analog Converter). On trouve ensuite l'étape de transposition en fréquence ; je transmettrai à 433 MHz environ (bande libre).

Pour vérifier que vous sauriez faire cette configuration, faire le quizz, puis regarder la vidéo 2.

2 Démodulation AM

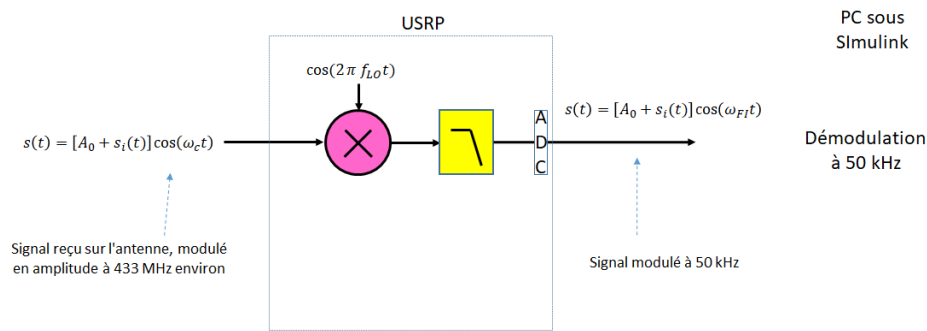
2.1 Principe du passage en fréquence intermédiaire

On propose, côté récepteur, de transposer le signal vers une fréquence porteuse plus faible avant démodulation, fréquence que nous appellerons *fréquence intermédiaire*, f_{FI} . Cette transposition se fait avec des composants RF ; évidemment, c'est possible facilement avec une USRP.

la fréquence intermédiaire doit être suffisamment faible pour que le signal modulé puisse être numérisé et observé sous Simulink. On choisit :

$$f_{FI} = 50 \text{ kHz}$$

Le schéma utilisé est le suivant :



Quelle valeur allez-vous choisir pour f_{LO} , si l'on choisit $f_{LO} > f_c$?

A quoi sert le filtre passe-bas? Comment doit être réglée sa fréquence de coupure? (noter que celle-ci est configurée dans l'USRP, sans que l'on puisse la modifier).

2.2 L'USRP en réception

On met le f_{LO} calculé à la partie précédente dans le paramètre **Center frequency** pour effectuer la transposition à $f_{FI} = 50$ kHz.

Un autre choix important concerne la fréquence d'échantillonnage de l'ADC qui suit le filtre passe-bas.

Pour respecter Shannon, quelle valeur minimale doit-on choisir pour cette fréquence? Donner une valeur numérique.

On prend une certaine marge, et on choisit 441 kHz. La raison est qu'au bout de la chaîne, après démodulation, il faudra envoyer à la carte son des échantillons à 44.1 kHz; cette réduction de f_e sera plus facile si le facteur de réduction est entier.

Regarder la vidéo 3 pour voir comment configurer l'USRP en réception.

2.3 Démodulation

Créer un nouveau fichier, avec le template **DSP System**. Vous allez utiliser comme source les échantillons qui sont sortis de mon USRP, et que j'ai enregistrés. Sachant que le sample rate à la sortie de l'USRP est de 441 kHz, et que chaque échantillon est codé sur 128 bits (on récupère la sortie des deux mixers que contient l'USRP, chacune est codée en format **double** sur 64 bits), estimer la taille de données nécessaire pour stocker 1 minute environ.

Vous avez dû télécharger le fichier **mystery_music.mat**. Ajouter un bloc **From File** pour lire les échantillons depuis ce fichier. Attention au paramètre **sample time**: il faut mettre celui que l'on avait à la sortie de l'USRP; encore une fois, on était à 441 kHz, et les échantillons sortaient par paquets de 128; soit un sample time de $1/441e3*128$. (Vérifier la valeur en faisant **CTRL+J**).

L'USRP, en émission comme en réception, contient en fait *deux* mixers, et ici, on récupère deux signaux à la fréquence intermédiaire, appelés I et Q :

$$I = [A_0 + s_i(t)] \cos(\omega_{FI} t)$$

$$Q = [A_0 + s_i(t)] \sin(\omega_{FI} t)$$

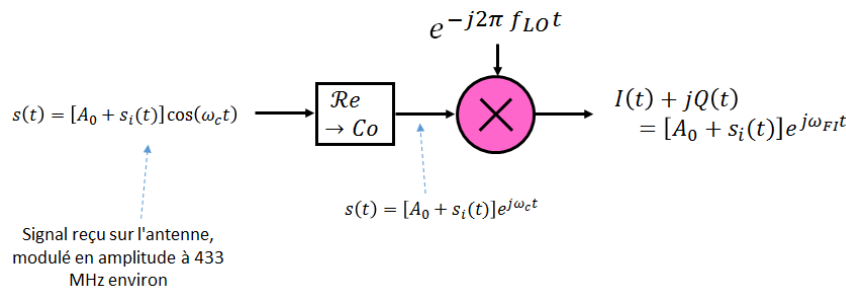
On les récupère sous la forme d'un signal complexe $I + jQ$. On pourrait conserver seulement I et utiliser le schéma classique, qui consiste à extraire l'enveloppe du signal modulé. Toutefois,

on peut, en gardant à la fois I et Q , encore simplifier le schéma (c'est la magie de la radio logicielle).

On obtient à la sortie de l'USRP (ou du bloc `From File`) :

$$I + jQ = [A_0 + s_i(t)] e^{j\omega_{FI}t}$$

On aurait pu trouver cette expression rapidement en utilisant le modèle complexe de l'USRP en réception, que nous reverrons dans le TP3 :



C'est le *module* de ce signal complexe qui contient l'information.

Un seul bloc, `Complex to Magnitude-Angle` permet donc de démoduler.

L'ajouter derrière le bloc `From File`. Connecter aussi au `From File` une `Constellation`. C'est intéressant car la distance à l'origine d'un échantillon placé sur la constellation représente l'amplitude.

Exécuter votre modèle quelques secondes, et observer la constellation ; le rayon doit "vibrer" en permanence, puisque l'amplitude est modulée par le son ; en observant sa valeur moyenne, on peut savoir si le signal reçu est trop bas, ou au contraire s'il sature. On ajuste le gain en conséquence³.

2.4 Sortie sur la carte son

Pour jouer le son, il faut commencer par ôter la composante continue du signal démodulé à l'aide d'un bloc `DC blocker`.

La carte son attend des échantillons avec un sample rate prédéfini, typiquement 44,1 kHz. Le sample rate à la sortie de l'USRP est de 441 kHz ; il faut donc le réduire d'un facteur 10 pour passer à 44,1 kHz. Pour cela, utiliser un bloc `FIR Decimation`, et indiquer que le taux est 10. Dans `Input processing`, choisir `columns as Channels (frame based)` et dans `Rate options`, `Allow multirate processing` (sinon le sample rate ne change pas, c'est la taille de la frame qui varie). Exécuter votre modèle : vous devez récupérer le son. Pour valider, utiliser par exemple `shazam`, et envoyer le titre sous Discord.

3 Connaissances, Capacités

Une connaissance est un concept théorique (qu'on pourrait parfois chercher sur Wikipedia). Une capacité est associée à un verbe d'action : il s'agit de savoir faire quelque chose. Vérifier que vous avez acquis les connaissances et capacités visées.

3. Dans un vrai système, cet ajustement du gain est automatique : on parle de Automatic Gain Control, AGC

Connaissance (Co)/ Capacité (Ca)	Acquise ?
Co : contenu des chaînes d'émission et de réception de l'USRP	
Co : correspondance entre paramètres des blocs SDRU Transmitter / Receiver et l'USRP	
Ca : calculer la puissance de sortie de l'USRP en fonction du gain (en transmission)	
Ca : utiliser l'USRP et simulink pour émettre un signal modulé en amplitude, avec ou sans porteuse	
Ca : Savoir utiliser l'analyseur de spectre de simulink pour observer un spectre	
Co : Principe de la transposition en fréquence intermédiaire	
Ca : utiliser l'USRP et simulink pour démoduler un signal modulé en amplitude, avec porteuse	
Ca : choisir le sample rate sur un modèle simulink basique (le connaître et justifier en tout point)	