

## 41-Oscilloscope et spectroscopie

Il y a des moments où voir une image d'un signal est instructif. Dans le monde réel, cela se fait avec un oscilloscope, qui montre un graphique d'amplitude en fonction du temps. Max a deux façons de le faire. La méthode old school utilise un objet *scope~* dédié et la nouvelle méthode qui utilise Jitter. La nouvelle n'a pas remplacé l'ancienne, il y a des applications pour chacune.

Lorsque vous tracez un graphique d'un signal en temps réel, vous devez résoudre deux problèmes. Tout d'abord, pour que votre œil puisse suivre un signal qui varie dans le temps, vous devez capturer et afficher un extrait du signal pendant une certaine période de temps (suffisamment longue pour que vous puissiez vraiment le voir). Par conséquent, le graphique doit être affiché périodiquement et sera toujours légèrement en retard par rapport à ce que vous entendez. Deuxièmement, il n'y a pas assez de pixels sur l'écran pour que vous puissiez voir un graphique de chaque échantillon (du moins, sans que l'affichage soit mis à jour à une vitesse aveuglante), l'affichage doit donc utiliser un seul pixel pour résumer de nombreux échantillons.

### Old School: *scope~*

L'objet *scope~* apparaît sous la forme d'un écran rectangulaire subdivisé en une fine grille. Lorsque le son est activé, une ligne horizontale montre la forme d'onde connectée à l'entrée gauche. (Cette ligne est appelée la *trace*.) Le principe de fonctionnement est simple. Un groupe d'échantillons est rassemblé et une valeur composite est calculée et placée à un endroit dans une mémoire tampon. Le processus se répète jusqu'à ce que le tampon soit plein. Lorsque la mémoire est pleine, des lignes sont tracées d'un point à l'autre de l'écran. Le nombre d'échantillons par point est défini par l'attribut **calccount** (il peut être modifié par un *int* situé dans l'entrée gauche). Le nombre de points est défini par l'attribut **buffer size** (qui peut être modifié par un *int* situé dans l'entrée de droite). Obtenir un affichage utile dépend de la sélection de l'attribut **calccount** et de la taille de **buffer size**, et ceux-ci dépendent de ce que vous essayez de montrer.

### Affichage des formes d'onde

- Sélectionnez **sine** dans l'*umenu* du patch de tutoriel.

Lorsqu'un signal constant est appliqué à *scope~*, ce que vous voyez dépend de la relation entre la fréquence, le taux d'échantillonnage, **calccount** et **buffer size**. **Buffer size** doit être aussi élevée que possible, donc réglez-la sur la largeur de l'objet jusqu'au maximum de 256. En général, plus **calccount** est faible, plus les détails sont visibles (il s'agit en fait d'un sous-échantillonnage de la forme d'onde); il est donc préférable de commencer à 2. Utilisez davantage pour les très basses fréquences, mais 16 permettra de descendre jusqu'à 10 Hz.

Il est fort probable qu'une forme d'onde stable n'ait pas l'air si stable que ça: elle va dériver sur l'écran. La solution à ce problème est l'attribut **trigger**, qui est réglé par défaut sur **0: off** mais doit généralement être réglé sur **1: up**. Cela verrouille l'affichage sur une certaine valeur de la forme d'onde: **triglevel**. Si le déclenchement est activé, l'objet *scope~* attendra que l'onde traverse le **triglevel** dans la direction indiquée avant de commencer un nouveau cycle d'affichage. Ainsi, une forme d'onde simple sera généralement affichée de façon stable. Avec une forme d'onde complexe, **triglevel** peut nécessiter un ajustement pour obtenir de meilleurs résultats.

- Choisissez **triangle** dans le *umenu*.
- Utilisez l'objet *attrui* pour définir le **trigger** sur **up**.
- Utilisez l'objet *attrui* pour ajuster **triglevel**.

Notez que l'onde glisse d'avant en arrière lorsque **triglevel** passe de de -1 à 1.

## Affichage des oscillogrammes

- Connectez un signal musical à votre interface audio et sélectionnez **input** dans l'*umenu* du patch de didacticiel.

Parfois, nous voulons observer le comportement à long terme d'un signal. Cela revient à réduire la vitesse de balayage d'un oscilloscope matériel. Vous voyez l'enveloppe du signal plutôt que les ondes individuelles. Le réglage équivalent sur un *scope~* est une valeur élevée de **calccount**. 128 serait un bon début. La première chose que vous remarquerez est que l'affichage prend une éternité à se mettre à jour. Combien de temps? Eh bien, le nombre d'échantillons nécessaires pour remplir l'affichage est **calccount \* buffer size**; donc divisez-le par le taux d'échantillonnage pour obtenir le nombre de secondes entre les mises à jour. Avec un **calccount** de 128 et une **buffer size** de 256, le temps de mise à jour est de 0,75 seconde. Le même calcul vous aidera à estimer le **calccount** approprié pour afficher une forme d'onde complète; vous voulez que le calcul corresponde à la période de l'onde: un sur la fréquence.

## Que signifient les lignes de la grille?

*scope~* est divisé en une grille par des lignes horizontales et verticales. La ligne horizontale centrale marque le point zéro. Les autres horizontales sont tout simplement à mi-chemin des bords. Les valeurs en haut et en bas sont définies par l'attribut **range**, dont la valeur par défaut est  $\pm 1,0$ . La signification des lignes verticales dépend de **calccount** et de **buffer size**. Comme mentionné ci-dessus, le produit de ces dernières divisé par la fréquence d'échantillonnage détermine la **période d'affichage**. Les lignes verticales divisent cette valeur par 8. Ainsi, si **calccount** est égal à 2 et que **buffer size** 256, la **période d'affichage** sera de 0,0116 à 44,1 kHz. L'inverse de cette valeur donne la fréquence d'une forme d'onde qui remplirait exactement la fenêtre, donc 86,1 Hz devrait convenir. L'intervalle de temps représenté par les lignes verticales est d'environ 1,5 ms.

## Affichage X-Y

- Connectez un signal musical stéréo à votre interface audio et ouvrez le sub-patch X-Y.

Si vous connectez un signal audio à l'entrée droite de *scope ~*, ce signal sera utilisé pour déterminer l'emplacement X (de gauche à droite) de chaque point. Si le même signal est connecté à gauche et à droite, vous obtenez une ligne diagonale allant du bas à gauche au haut à droite. Si le signal à droite est une version déphasée du signal à gauche, la ligne est inclinée dans l'autre sens. Lorsqu'un signal stéréo est connecté, le résultat est une sorte de gribouillage et vous pouvez estimer la différence de phase de la gauche et de la droite par la largeur et la pente de la forme.

Si vous connectez des formes d'onde géométriques aux deux entrées de *scope ~*, vous pouvez générer des formes évolutives intéressantes. Ces formes sont appelées figures de Lissajous et méritent d'être explorées en elles-mêmes. Le fichier d'aide de *scope~* contient quelques exemples.

## New school : *jit.catch ~* et *jit.graph*

Nous pouvons également afficher des formes d'onde dans Jitter avec les objets *jit.catch ~* et *jit.plot*. *Jit.catch~* prend des échantillons d'un signal audio et les regroupe dans une matrice unidimensionnelle. Le nombre d'échantillons par matrice est défini par l'attribut **framesize** et a exactement la même signification que le calcul de **calccount \* buffer size** dans *scope ~*. *Jit.catch~* a plusieurs modes de fonctionnement, mais celui qui nous intéresse est le **mode 3**, qui est déclenché

à la manière d'un oscilloscope. Bien sûr, *jit.catch* ~ nécessite une série de bangs pour déclencher la sortie des matrices.

*Jit.graph* affiche les données dans des matrices à une dimension dans un tableau de caractères (*char*) à deux dimensions. Connectez cela entre *jit.catch*~ et une fenêtre *jit.pwindow*, et vous avez un affichage de forme d'onde. *Jit.graph* a aussi des modes qui déterminent le style d'affichage:

0: points  
1: lignes  
2: zone  
3: zone bipolaire  
4: mesures

Lignes sont les plus proches d'un scope.

La fonction de seuil dans *jit.catch* ~ est un peu étrange dans la mesure où le point de seuil est toujours placé au **milieu** de la matrice. Ainsi, si vous ajustez le **framesize**, la forme d'onde restera centrée dans l'affichage. Cela signifie également que la direction du trigger (**trigdir**) doit être définie dans le mode down (1) pour correspondre à ce que *scope*~ fait avec la même onde. L'affichage peut être encore modifié avec les attributs **rangelo** et **rangehi** qui définissent les valeurs qui touchent le bas ou le haut de l'écran. Il existe également un attribut **height**, qui définit la dimension verticale de la matrice. Il doit être défini pour correspondre à la hauteur de la fenêtre *jit.pwindow* afin d'éviter les artefacts de conversion de taille.

Le choix entre l'utilisation de *scop* ~ ou de Jitter pour afficher l'audio dépend de ce que vous voulez faire. Si un affichage audio est nécessaire dans le patch, *scope*~ est généralement le meilleur choix. Si vous souhaitez traiter l'image plus à la manière d'iTunes, Jitter est la solution.

## Tracer un spectre

Deux mots à la mode dans la communauté DSP sont le **domaine temporel** et le **domaine fréquentiel**. Le **domaine temporel** signifie simplement que nous représentons graphiquement l'évolution des valeurs du signal, comme nous l'avons montré ci-dessus. Le **domaine fréquentiel** montre le son à un moment donné dans le temps, mais montre le contenu de la FFT (fréquence de Fourier). Ainsi, le graphique (appelé **spectrogramme**) présente la fréquence en bas. Vous vous souvenez que la FFT stipule que toute forme d'onde peut être décomposée en une somme d'ondes sinusoïdales. Un spectrogramme représente l'amplitude de ces ondes sinusoïdales. Nous entrerons dans les détails au chapitre 4 de l'analyse: *Analyse de Fourier simple*. Ici, nous allons simplement explorer une manière simple d'afficher un spectrogramme.

## Spectrogramme

- Sélectionnez **triangle** dans l'*umenu* et réglez la fréquence sur 440 Hz.

L'objet *spectroscope*~ affiche un beau spectrogramme dès sa sortie de la boîte. Les pics de la trace indiquent la fréquence et l'amplitude des partiels de la forme d'onde triangulaire. Dans un système idéal, ces pics seraient des lignes fines, mais l'algorithme requis pour afficher un spectre en temps réel produit des "jupes" au bas des pics. Le principal attribut à prendre en compte est la mise à l'échelle de l'axe des fréquences, qui est choisie par l'attribut **logfreq** et indiquée par **Afficher l'échelle des axes de fréquence** dans l'*attrui*. En **mode linéaire**, toute la plage audible de 20 à 22 kHz est uniformément répartie de gauche à droite. L'affichage du *spectroscope*~ doit montrer un bel ensemble d'harmoniques uniformément espacées. En fait, il s'agit d'une harmonique sur deux,

comme vous pouvez le voir en sélectionnant **rectangle**. Si vous choisissez *input*, la plupart de l'action sera concentrée sur le côté gauche. C'est parce que la musique est principalement confinée à une gamme de fréquences assez basse.

- Sélectionnez **Afficher l'échelle des axes des fréquence** dans l'*attrui* attaché à l'objet *spectroscope~* et choisissez **Échelle logarithmique**.
- Sélectionnez sinus dans l'*umenu* et réglez la fréquence sur 43 Hz.

En mode logarithmique, les *octaves* sont uniformément réparties sur l'affichage. L'affichage doit montrer un large pic à environ un tiers de l'échelle. (Ignorez ce qui se trouve à gauche de la crête: la résolution de l'affichage n'est que de 43 Hz, donc tout ce qui se trouve en dessous est sans intérêt.) Changez maintenant la fréquence à 86 Hz et notez comment le pic se déplace. Au fur et à mesure que vous augmentez les octaves: 129, 172, 215, etc., le pic se déplace d'autant. Le pic se rétrécit également. La bande de résolution de 43 Hz représente une part moins importante du total au fur et à mesure que l'on monte. Maintenant, sélectionnez **input**. La musique remplit joliment l'écran.

Les deux modes d'affichage sont très utiles: le mode linéaire montre la structure harmonique des formes d'onde stables et le mode logarithmique montre l'équilibre tonal de la musique de manière intuitive.

## Sonogramme

Nous pouvons obtenir les informations du domaine temporel et du domaine fréquentiel sur un seul affichage en choisissant le mode sonogram.

- Sélectionnez le mode d'affichage dans *attrui* et choisissez Sonogram.
- Sélectionnez l'entrée dans *umenu*.

Vous devriez maintenant voir une image dessinée avec des bandes de noir, de gris et de blanc. Un sonogramme est un tracé bidimensionnel indiquant le temps en bas et la fréquence de bas en haut. L'amplitude à chaque point est représentée par l'intensité de la couleur. En écoutant de la musique, vous verrez comment les notes graves et les notes aiguës apparaissent et vous verrez même la structure harmonique du son. (Le mode **logfreq** affecte également les sonogrammes.) Pour un plaisir encore plus grand, sélectionnez **couleur de sonogramme** dans *attrui*, puis choisissez **couleur**. Maintenant, l'intensité des composantes de fréquence s'affiche alors en rouge, puis en jaune, en vert, et en violet.

L'objet *spectroscope~* peut être personnalisé de diverses manières, notamment l'orientation verticale, le mode dessin ou défilement, le sens inverse et les couleurs. Vous pouvez également restreindre la plage d'affichage pour obtenir plus de détails dans un petit espace.

## Résumé

L'objet *scope ~* donne une vue d'oscilloscope d'un signal, en traçant l'amplitude en fonction du temps. Les formes d'onde peuvent également être affichées dans le format de la matrice Jitter par *jit.catch ~* et *jit.graph*. L'objet *spectroscope ~* peut afficher le contenu fréquentiel sous forme de spectrogramme dynamique ou de sonogramme mobile.