

35-données de contrôle de la fréquence audio

Oscillateurs basse fréquence

Ce didacticiel traite de l'utilisation de l'objet *poly ~* pour générer des oscillateurs basse fréquence (LFO) afin de générer des données de contrôle pour les objets Max. Il existe de nombreuses tâches dans Max (comme les graphiques) qui ne nécessitent pas de mises à jour de contrôle à un taux audio (44 100 par seconde ou plus) et qui ont simplement besoin de recevoir régulièrement de nouvelles valeurs à une vitesse inférieure. Par exemple, un patcheur de traitement vidéo fonctionnant à 30 images par seconde n'aurait besoin de nouvelles informations pour tous ses paramètres que toutes les 33 millisecondes. En même temps, les générateurs de signaux MSP sont d'excellents choix pour concevoir des sources périodiques d'informations de contrôle. L'objet *poly ~* dispose d'un moyen simple de *sous-échantillonner* le traitement audio dans une abstraction, de sorte qu'il peut être utilisé pour créer des données de contrôle dans Max sans la surcharge CPU de travailler au taux audio.

- Jetez un coup d'œil au patcheur du tutoriel. Activez le *dac ~* (à droite) et le *metro* (à gauche) en utilisant les objets *toggle*. Vous devriez voir une ligne se dessiner dans l'objet *lcd* à droite du patcheur. Vous pouvez à tout moment appuyer sur la barre d'espace pour effacer le dessin.
- Dans les boîtes de *nombres X* et *Y*, tapez **1.** et **1.3**, respectivement. Regardez la courbe dessinée dans le *lcd*. Effacez l'écran à l'aide de la barre d'espace et modifiez les valeurs en **1** et **4** .. Remarquez que lorsque les valeurs sont dans un rapport entier, le modèle commence à se redessiner après un certain temps. Essayez vos propres nombres ou modifiez-les au fur et à mesure.

L'utilisation d'ondes cosinusoïdales de fréquences différentes comme entrées **X** et **Y** d'une surface de dessin (comme un oscilloscope) crée ce qu'on appelle des *courbes de lissajou*. Ces courbes décrivent visuellement un mouvement harmonique complexe, de sorte que les motifs **X** et **Y** dans des rapports simples créent des formes répétitives prédéterminées.

Utilisation de *snapshot ~*

- Double-cliquez sur l'un des objets *poly ~* pour voir l'intérieur de l'abstraction **polylfo1 ~**.

Remarquez à quel point notre abstraction semble simple, avec un objet *in* permettant de régler la fréquence d'un objet *cycle ~*. Un deuxième objet *in* fournit des messages **bang** à un *snapshot ~*. L'objet *snapshot ~* nous permet de capturer des valeurs uniques de signaux MSP sous forme de nombres à virgule flottante générés comme des événements Max. A chaque fois qu'un *snapshot ~* reçoit un **bang**, il échantillonne le signal MSP courant à son entrée et sort cet échantillon sous forme de valeur à virgule flottante. Dans notre tutoriel, cette valeur est ensuite envoyée à un objet *out*, puis renvoyée à notre patcheur principal.

Changer les abstractions

- Fermez le **polylfo1~** d'abstraction et revenez au patcheur principal du didacticiel. Cliquez sur les boîtes de *messages* du milieu qui commencent par le message **patchername**. Le message **patchername recharge** un objet *poly ~* avec une nouvelle abstraction. Remarquez l'effet sur le dessin, selon ce que vous sélectionnez et selon qu'il s'applique à l'objet *poly~ X* ou **Y**. Sélectionnez **polylfo2 ~** et double-cliquez sur l'objet *poly ~* pour le voir.

L'abstraction de **polylfo2 ~** fait la même chose que **polylfo1 ~**, mais génère sa sortie sous forme d'une onde en dents de scie plutôt qu'en cosinus. L'objet *phasor ~* génère des rampes de **0** à **1**, nous

mettons donc à l'échelle la sortie pour qu'elle aille de **-1** à **1** avec les objets `* ~` et `- ~` avant de l'échantillonner avec l'objet `snapshot ~`.

- Changez l'un des objets `poly ~` en `polylfo3 ~` et regardez à l'intérieur.

L'objet `polylfo3 ~` génère une sortie triangulaire en utilisant un objet `triangle ~`. L'objet `triangle ~` enveloppe un `phasor ~` autour d'un point central, le transformant essentiellement en une forme d'onde triangulaire. Cet objet produit des valeurs comprises entre **-1** et **1**, il ne nécessite donc pas de mise à l'échelle.

Arguments de rééchantillonnage

- Fermez toutes les abstractions ouvertes et retournez au patcheur principal. Regardez les arguments des objets `poly ~`.

Lorsque vous travaillez avec une abstraction à l'intérieur d'un objet `poly ~`, vous pouvez décider de travailler à un taux d'échantillonnage inférieur ou supérieur à celui de votre patcheur principal MSP. Ce taux n'est pas spécifié directement, mais comme un diviseur ou multiplicateur du taux principal. Les arguments **up 2** feront fonctionner le `poly~` à deux fois la fréquence d'échantillonnage du patcheur principal. Les arguments **down 8**, comme indiqué ci-dessous, font tourner l'audio dans les objets `poly ~` à un *huitième* de la vitesse normale (c'est-à-dire 5,5125 kHz si le patcheur principal tourne à 44,1 kHz). Puisque nous n'échantillons les valeurs de notre oscillateur qu'une fois par image de dessin (fixé par la vitesse du `metro` à 50Hz ou toutes les 20 ms), cela constitue une résolution plus que suffisante pour nos données de contrôle.

- Mélangez et associez différentes courbes LFO en choisissant différentes abstractions `poly~` exécutées à différentes fréquences. Remarquez comment les courbes de Lissajous se comportent avec les différentes formes d'onde.

Résumé

L'objet `poly ~` peut travailler à une fréquence d'échantillonnage différente de celle de son patcheur MSP hôte. Les arguments **up** et **down** vous permettent de spécifier un multiplicateur ou un diviseur pour la fréquence d'échantillonnage principale. Lors de la conception de LFOs (oscillateurs basse fréquence), il est souvent utile d'utiliser des patchs MSP massivement sous-échantillonnés à l'intérieur d'objets `poly~` pour générer efficacement les courbes. L'objet `snapshot ~` vous permet d'échantillonner une seule valeur d'un signal MSP en réponse à un **bang**.