

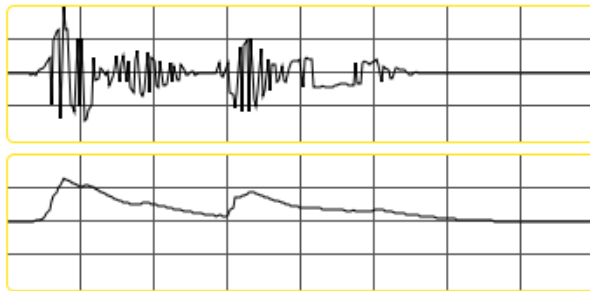
## 27-Suivi d'enveloppe

Dans ce groupe de didacticiels, nous allons examiner différentes façons de travailler avec la dynamique des signaux audio. La capacité d'utiliser et de contrôler l'amplitude d'une forme d'onde audio est importante pour de nombreuses applications, du niveau macro (compression audio) au niveau micro (distorsion). Dans ce premier tutoriel, nous allons apprendre à dériver des valeurs de contrôle à partir des paramètres d'amplitude d'un signal audio que nous pouvons utiliser pour contrôler des paramètres ailleurs dans la chaîne du signal. Cette technique s'appelle le *suivi d'enveloppe*.

### L'enveloppe, s'il vous plaît ...

En travaillant avec MSP, nous avons appris que l'audio numérique est représenté par un flux d'échantillons discrets, chacun d'entre eux représentant *l'amplitude* d'un signal à un moment donné. Cependant, lorsque nous pensons (et utilisons) le terme amplitude, nous n'y pensons pas nécessairement sur une base échantillon par échantillon. Au contraire, nous discutons de l'amplitude d'un son en fonction de *l'intensité* qu'il nous semble avoir pour nous, auditeurs. Par ailleurs, nous créons des systèmes de mesure objectifs pour tenter de quantifier cette intensité sonore (subjective), comme les indicateurs de niveau dans les logiciels de mixage audio. Ces deux systèmes examinent l'amplitude d'un signal audio non pas échantillon par échantillon, mais *moyennée* dans le temps. Cette technique consistant à dériver la macro-amplitude de l'intensité sonore ou du volume d'un son (par opposition à la micro-amplitude des valeurs d'échantillon) est appelée *suivi d'enveloppe*.

Nous utilisons le terme enveloppe dans la conception des synthétiseurs pour désigner la forme dynamique globale du son. Par exemple, on peut dire qu'un son qui s'ouvre en douceur puis s'estompe progressivement dans le même laps de temps a une enveloppe triangulaire. Un son avec une attaque courte, un long sustain et un court fade peut ressembler à un trapèze. De même, nous pouvons prendre n'importe quelle source sonore et, avec un peu de travail, abstraire son enveloppe:



"Is that you?": Forme d'onde (en haut) et enveloppe (en bas)

Pour créer l'enveloppe ci-dessus, nous avons examiné les échantillons au sein de la forme d'onde et suivi leur amplitude moyenne dans le temps. Il existe plusieurs stratégies pour cela dans MSP, impliquant un trio d'objets qui nous permettent de lisser les signaux audio en fonction de différents paramètres.

- Dans le didacticiel, familiarisez-vous avec la zone étiquetée **1**. Il s'agit d'un patcheur de lecture d'échantillons de base qui charge un fichier audio dans un objet *buffer* ~ et le lit dans une boucle sans fin en utilisant l'objet *groove* ~. Lancez l'audio, montez le curseur de *gain* ~ et assurez-vous que vous pouvez entendre la boucle de batterie en cours de lecture.

## Suivi d'enveloppe dans MSP

- Regardez les deux objets en jaune attachés à la sortie de l'objet *groove* ~.

La première chose à faire pour obtenir l'amplitude lissée d'un son est de convertir le signal en une onde *redressée*, c'est-à-dire d'ignorer la différence entre les valeurs négatives et positives de l'échantillons et de se concentrer uniquement sur leur distance par rapport à 0. Nous pouvons le faire en prenant la valeur absolue d'un signal, ce qui peut être obtenu à l'aide de l'objet *abs* ~ dans MSP.

L'objet *snapshot* ~ nous permet de convertir un échantillon d'un signal MSP en un nombre à virgule flottante sorti de l'objet en tant qu'événement Max. Chaque fois que l'objet reçoit un **bang**, il prend l'échantillon audio actuel et le sort dans Max sous la forme de nombre qui peut être visualisé, mis à l'échelle ou utilisé pour déclencher des événements Max. Dans le cas de notre patcheur, la valeur absolue de la sortie de notre objet *groove* ~ est utilisée pour piloter la hauteur d'un objet *multislider*, avec le *snapshot* ~ fournissant la traduction intermédiaire d'un signal MSP à un message Max. Comme nous pouvons le constater, le rebondissement du *multislider* nous donne une vague impression de l'amplitude de la boucle de batterie. Cependant, une impression plus précise de la façon dont nous *entendons* le son peut être obtenue en lissant ces valeurs.

- Regardez la logique du patcher intitulée **2** dans notre tutoriel. Dans la boîte de *nombre* «Up», tapez la valeur **1**. et appuyez sur Retour. Dans la boîte de *nombre* 'Down', tapez la valeur **100**. et appuyez sur Retour. Observez les résultats tels qu'ils sont tracés dans les objets *multislider* ci-dessous. Faites maintenant l'inverse (tapez **1**. comme valeur 'Down' et **100**. comme valeur 'Up'). Vous remarquez la différence?

La sortie de l'objet *send* ~ (qui consiste en notre boucle de batterie) est convertie en valeur absolue, passe par un objet appelé *rampsmooth* ~, et est tracée dans une boîte de *nombres* et deux objets *multislider*. Le *multislider* de droite est dans l'un de ses modes de défilement, ce qui nous permet de voir les valeurs telles qu'elles se déroulent dans le temps. L'objet *rampsmooth* ~ effectue une opération de lissage linéaire sur un signal d'entrée. Ses deux arguments (ou nombres donnés aux entrées centrale et droite) contrôlent la manière dont le lissage se comporte. L'entrée du milieu contrôle combien d'échantillons il nous faut pour qu'un signal diminue. Ces deux valeurs nous permettent de lisser indépendamment les valeurs d'échantillons *croissantes* et *décroissantes*, de sorte que les montées brusques puissent être maintenues dans l'enveloppe mais aussi que des silences brusques génèrent une fin.

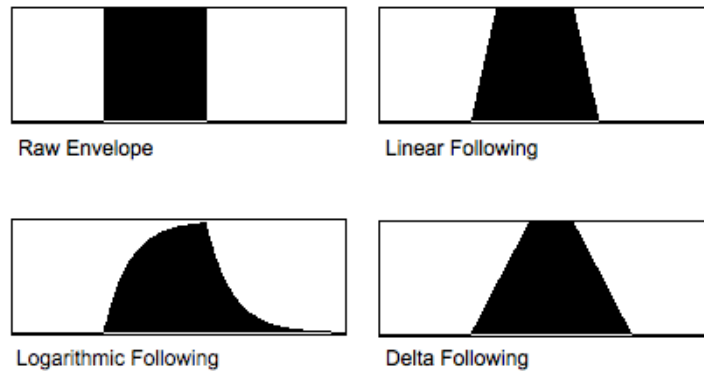
- Regardez la zone du didacticiel étiquetées **3**. Comme précédemment, entrez **1**. dans la boîte de *nombre* étiquetée «Up» et **100**. dans la boîte de *nombre* étiquetée «Down». Observez l'enveloppe dessinée dans le *multislider*. Maintenant, faites l'inverse (tapez **1**. en tant que valeur 'Down' et **100**. en tant que valeur 'Up'). Tapez **100**. pour les deux valeurs.

L'objet *slide* ~ lisse les signaux audio de manière logarithmique. Les deux valeurs fournies fonctionnent comme des dénominateurs pour la nouvelle énergie montante ou descendante entrant dans le signal (lissé). Par exemple, une valeur "Up" de **1000** signifie qu'il faudra 1000 échantillons pour qu'une transition de 0 à 1 se produise. Plus les valeurs "Up" et "Down" sont grandes, plus la courbe est lisse.

- Regardez la zone du patcheur **4**. Dans la boîte de *nombre* intitulée «Delta», tapez **0,01** et appuyez sur Retour. Observez les résultats. Tapez un nombre encore plus petit, tel que **0.0001**.

L'objet *deltaclip* ~ fonctionne d'une manière différente de *rampsmooth* ~ et *slide* ~. Il définit un seuil pour la quantité de changement d'un signal audio, et écrête le signal sortant à cette quantité. En réglant les valeurs min et max pour le delta sur **-1** et **1**, les échantillons passeront normalement à condition qu'ils ne changent pas de plus de cette quantité. Des valeurs plus petites lisseront le signal audio de manière *relative*, de sorte que l'enveloppe résultante trace un ensemble de valeurs beaucoup plus étroit que le signal entrant.

Si nous faisons passer une rafale de bruit par les différentes techniques de suivi d'enveloppe, présentées dans notre didacticiel de patcher, nous pouvons voir la différence entre les courbes qu'elles génèrent:



*Différents suivis d'enveloppe sur une explosion de bruit.*

Chacune de ces techniques de suivi d'enveloppe a des applications musicales et sonores différentes et possède des signatures différentes. Un suiveur d'enveloppe logarithmique contrôlant le volume d'un synthétiseur, par exemple, aurait un son très différent d'un suiveur linéaire.

## Utiliser des enveloppes comme signaux de contrôle

- Examinez la logique du patcheur étiquetée **5** dans le didacticiel. Montez le curseur de *gain* ~ pour entendre la sortie de l'objet *cycle* ~. En utilisant un menu, sélectionnez les différents suiveurs d'enveloppe comme entrée de contrôle. Si vous le souhaitez, modifiez également leurs paramètres pour voir quels types d'effets différents vous pouvez obtenir.

Les trois zones de patcheur suivi d'enveloppe envoient leurs signaux de sortie par le biais d'objets *send*~ à tout *receive*~ réglé sur le nom approprié. Dans le cas de la logique du patcheur **5**, ce signal (allant de **0** à **1**) contrôle la fréquence d'un objet *cycle* ~, mappant le signal de contrôle entre **100** et **1100** Hz. La différence entre les trois types d'enveloppes devrait être apparente si elles sont réglées à des valeurs de lissage assez élevées.

- Baissez le curseur de *gain* ~ dans la zone de patcheur **5** et regardez la logique du patcheur étiquetée **6**. Montez le curseur de *gain* ~ et sélectionnez différentes sources pour un signal de contrôle dans l'*umenu*.

Dans cet exemple, nous utilisons notre signal de contrôle d'enveloppe pour moduler la fréquence de coupure d'un filtre *lores* ~, créant ainsi un effet wah-wah dynamique sur une forme d'onde mélangée et désaccordée.

- Baissez la zone de patcheur **6** et regardez la logique de patch étiquetée **7**. Montez le curseur de *gain* ~ et sélectionnez différentes enveloppes dans l'*umenu*.

Dans cet exemple, nous utilisons notre enveloppe pour créer un «auto-wah» sur la boucle de batterie elle-même. Il s'agit d'une utilisation très courante du suivi d'enveloppe dans la production musicale, où un signal de contrôle dérivé d'un signal audio est utilisé pour contrôler un paramètre d'un effet appliqué à ce même clip audio plus loin dans la chaîne. En utilisant cette logique, nous pourrions utiliser notre suiveur d'enveloppe pour produire des signaux de contrôle pour n'importe quelle procédure de traitement de signal que nous voulons dans MSP.

## Résumé

Lorsque nous parlons de *l'amplitude* d'un signal audio, nous faisons la distinction entre l'amplitude d'échantillon par échantillon et l'intensité sonore globale d'un son. Cette deuxième forme est appelée *l'enveloppe* d'un son et le traitement du signal qui permet de la dériver est appelé le *suivi d'enveloppe*. Les enveloppes des signaux audio peuvent être créées en regardant la forme d'onde absolue (rectifiée) (créé avec l'objet *abs~*) et en la lissant à l'aide d'objets MSP tels que *rampsmooth~*, *slide~* ou *deltaclip~*. Ces signaux de contrôle lissés peuvent ensuite être mis à l'échelle et utilisés pour contrôler les paramètres de n'importe quelle opération de synthèse ou de traitement du signal de votre choix. L'objet MSP *snapshot ~* est utile pour convertir les signaux audio MSP en nombres Max à virgule flottante qui peuvent être visualisés en utilisant des objets tels que *multislider*.