56-Ajuster la compression

Il y a plus de messages utilisés pour contrôler l'objet *omx.comp ~* que de contrôles sur de nombreux compresseurs. L'utilisation de ces messages vous donne un accès sans précédent au fonctionnement interne de l'objet et vous permet de régler son comportement pour une grande variété de sons. En plus de Threshold, Ratio, Attack et Release, les messages suivants sont disponibles: (Essayez-les dans le patcher *C3mTweakingCompression*.)

.**Couplage de canaux**: les signaux stéréo ont naturellement des niveaux différents de chaque côté. Si un canal déclenche la compression mais pas l'autre, les niveaux relatifs des canaux seront modifiés. Cela se traduira par un déplacement de l'image sonore. C'est pourquoi, une compression égale est appliquée aux deux canaux, même si un seul en a besoin. Normalement, la quantité de compression est déterminée par le canal le plus fort à un moment donné. Si **channelCoupling** est réglé sur 1, la compression suivra le canal gauche. Sur 2, c'est le canal droit qui détermine l'action.

.**SmoothGain**: Comme le détecteur de niveau fonctionne par étapes discrètes, l'application de la mesure de niveau directement à la commande de gain entraînerait une action en escalier ou un «bruit de fermeture éclair». Pour éviter cela, une enveloppe (similaire à *line ~*) est appliquée au signal de contrôle. **SmoothGain** contrôle le temps de cette enveloppe.

.**Delay**: Ce message retarde l'application de la meusre à la commande de l'amplificateur. Cela permet d'obtenir des crêtes rapides pour un son plus percutant. Combiné avec un temps d'attaque rapide, cela produira des coups de grosse caisse forts mais très courts, parfaits pour le hip hop.

.**Sidechain**: un filtre Sidechain applique l'EQ vocal inverse décrit dans le prochain tutoriel au circuit de mesure. Cela réduira l'effet des voix sur le niveau de sortie, utile si vous compressez un mixage complet. Il empêchera la voix de pousser vers le bas les parties soutenues telles que l'orgue.

.**NoiseGate**: Le Noise Gating est presque toujours nécessaire lorsque vous compressez d e l'audio provenant du monde réel. Par exemple, si vous enregistrez un ampli de guitare, le compresseur va accentuer le ronflement de l'ampli entre les notes. Pour éviter cela, *omx.comp ~* dispose d'une noise gate intégrée. Lorsqu'il est activé, il applique une expansion descendante à tout signal inférieur au **ngThreshold**. Il n'y a pas de commandes d'attaque et de relâchement car la *gate* est normalement appliquée à des signaux très faibles.

.**Release**: les trois messages **gatingLevel**, **freezeLevel** et **progressiveRelease** effectuent des ajustements automatiques du délai de relâchement en fonction du signal d'entrée. Les signaux inférieurs au niveau de gel ne déclenchent aucune amplification. Les signaux supérieurs à ce niveau mais inférieurs au niveau de déclenchement auront un relâchement lent. Le relâchement progressif accélère le temps de relâchement sur les entrées les plus fortes. Cela signifie que les signaux plus doux (où l'oreille est moins sensible aux changements dynamiques) ne seront pas aussi compressés que les signaux forts.

.**LimEnabled**: Ce message fixe une limite absolue à la force du signal de sortie. Cela permet d'éviter une surcharge du signal lorsque les commandes sont configurées pour le punching.

Messages de paramètres

Les objets OMX envoient des messages depuis la troisième sortie lorsque des paramètres sont modifiés. Ceux-ci se présentent sous forme de listes de paramètres contenant 7 champs:

1. Portée 0 = globale, 1 = prédéfinie, 2 = fin de liste.

2. Nom du paramètre.

3. Valeur actuelle.

4. Valeur maximale.

5. Valeur minimale

6. Valeur d'affichage.

7. Unités - cela est ***arbitraire*** pour la plupart des éléments.

Les valeurs des paramètres sont mises à l'échelle de façon arbitraire, la plupart de 0 à 100. Certaines d'entre elles peuvent être converties en unités familières. Tout paramètre dB peut être converti en regardant la valeur affichée lorsqu'elle est réglée sur 0 - c'est la valeur minimale (100 correspond à 0 dB). Divisez 100 par cette valeur pour obtenir le facteur d'échelle. Ensuite, pour définir une valeur, ajoutez l'absolu du minimum et multipliez-le par l'absolu du facteur de mise à l'échelle, comme ceci:

> .Pour convertir agcThreshold (de –36 à 0), utilisez un objet *expr* avec les arguments suivants: (36 + $ i1) * 2.77.
>
> Pour convertir ngThreshold (de –90 à 0), utilisez un objet *expr* avec les arguments suivants: (90 + $ i1) * 1.11. Il est probablement plus facile de capturer les nombres qui sortent dans les messages de paramètres et de les afficher.
>
> Le ratio utilise des valeurs exponentielles. Pour convertir les nombres donnés en ratio, utilisez un objet *expr* avec les arguments suivants: ln ($ f1) / ln (1.04).

Préréglages

Avec autant de paramètres, il est essentiel de disposer d'une méthode permettant de rappeler des combinaisons de réglages. Les compresseurs ont quelques préréglages intégrés (sélectionnables à l'aide du message **ChoosePreset**), mais la solution ultime sera de sauvegarder la configuration complète dans un objet *coll* ou *pattr*. Le message **SaveSettings** vous donnera une copie de tous les paramètres.

Indicateurs de niveau

La sortie de droite permet de mesurer le gain si elle est activée avec le message **meters**. Les sorties sont une liste prête à être connectée à un objet *multislider*. La liste pour *omx.comp* ~ montre les valeurs gauche et droite pour le gain AGC, le noise gate et le limiteur. Le message **meterRate** détermine la fréquence d'envoi de ces valeurs. Vous pouvez également obtenir les valeurs à l'aide du message **meterData**. Les vu-mètres pour lrd compresseurs à 4 et 5 bandes sont assez élaborés.