

49-Filtre en peigne

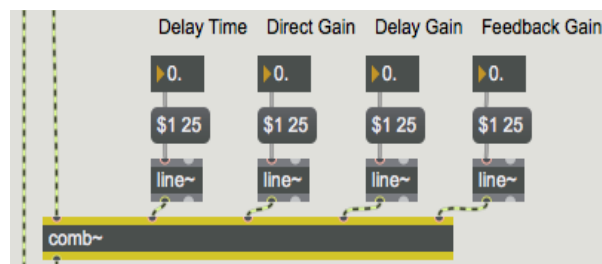
Filtre en peigne: *comb ~*

Le temps de retard minimum qui peut être utilisé pour le feedback d'une ligne à retard en utilisant *tapin ~* et *tapout ~* est déterminé par la taille du vecteur de signal. Cependant, de nombreuses formules de filtrage intéressantes nécessitent un retour utilisant des temps de retard de seulement un ou deux échantillons. De tels processus de filtrage doivent être programmés dans un seul objet MSP.

Un exemple d'un tel objet est *comb ~*, qui implémente une formule de **filtrage en peigne**. De manière générale, un filtre audio est un amplificateur dépendant de la fréquence; il augmente l'amplitude de certaines composantes de fréquence d'un signal tout en réduisant d'autres fréquences. Un filtre en peigne accentue et atténue le signal d'entrée à des intervalles de fréquence régulièrement espacés, c'est-à-dire à des multiples entiers d'une certaine fréquence fondamentale.

Détail technique: la fréquence fondamentale d'un filtre en peigne est l'inverse du temps de retard. Par exemple, si le temps de retard est de 2 millisecondes (1/500 de seconde), l'accentuation se produit à des intervalles de 500 Hz (500, 1000, 1500, etc.) et l'atténuation se produit entre ces fréquences. L'extrémité de l'effet de filtrage dépend du facteur (entre 0 et 1) par lequel le feedback est mis à l'échelle. Lorsque le facteur d'échelle s'approche de 1, l'accentuation et l'atténuation deviennent plus extrêmes. Cela provoque l'effet sonore de résonance (un son «résonant») aux harmoniques de la fréquence fondamentale.

L'objet *comb ~* émet un signal qui est une combinaison a) du signal d'entrée, b) du signal d'entrée qu'il a reçu il y a un certain temps et c) du signal de sortie qu'il a envoyé il y a le même temps (qui aurait inclus des retards antérieurs). Dans les entrées de *comb ~* nous pouvons spécifier la quantité désirée de chacun de ces trois éléments (a, b et c), ainsi que le temps de retard (nous l'appellerons d).



Vous pouvez ajuster tous les paramètres du filtre en peigne

Détail technique: À un instant donné (nous appellerons cet instant t), le *comb~* utilise la valeur du signal d'entrée (x_t) pour calculer la sortie y_t de la manière suivante:

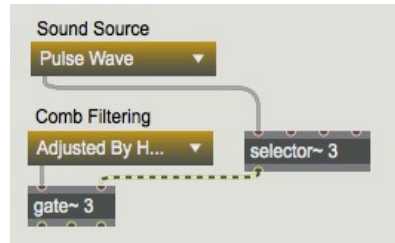
$$y_t = a x_t + b x_{(t-d)} + c y_{(t-d)}$$

Le terme $b x_{(t-d)}$ dans l'équation est appelé **feedforward** et le terme $c y_{(t-d)}$ est le **feedback**.

La fréquence fondamentale du filtre en peigne dépend du temps de retard, et l'intensité du filtrage dépend des trois autres paramètres. Notez que le facteur d'échelle pour la rétroaction (l'entrée de droite) ne doit généralement pas dépasser 1, car cela entraînerait une augmentation constante de la sortie du filtre au fur et à mesure qu'un signal de plus en plus grand est réinjecté.

Essayer le filtre en peigne

Le patch du didacticiel vous permet d'essayer le filtre en peigne en l'appliquant à différents sons. Le patch vous fournit trois sources sonores possibles pour le filtrage - l'entrée audio de votre ordinateur, une onde pulsée à bande limitée ou un bruit blanc - et trois options de filtrage: non-filtré, filtre en peigne avec paramètres ajustés manuellement ou filtre en peigne avec paramètres modulés en continu par d'autres signaux.



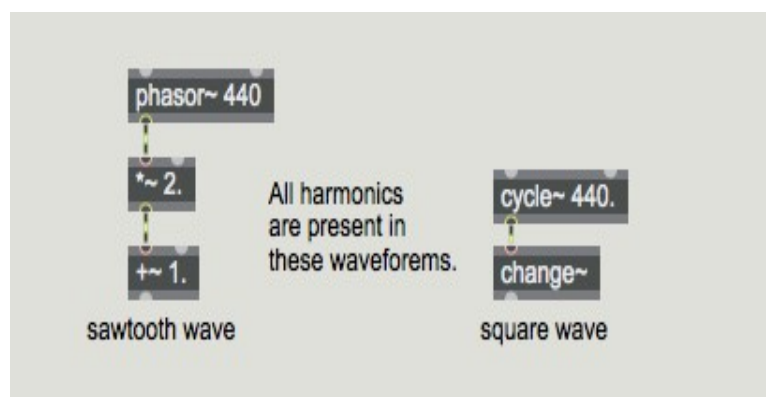
Choisissez une source sonore et dirigez-la vers le filtrage souhaité à l'aide des menus contextuels.

- Cliquez sur les boutons du *preset* pour essayer différentes combinaisons, avec des exemples de réglages de paramètres. Écoutez l'effet du filtre, puis expérimentez en modifiant vous-même les paramètres. Vous pouvez utiliser les messages de note MIDI de votre synthé pour fournir des informations de hauteur et de vélocité (fréquence et amplitude) pour l'onde d'impulsion, et vous pouvez utiliser la molette de modification pour modifier le temps de retard du filtre.

Un filtre en peigne a une résonance harmonique caractéristique en raison des fréquences équidistantes de ses crêtes et de ses creux d'amplification. Ce trait est particulièrement efficace lorsque le filtre en peigne est balayé vers le haut et vers le bas en fréquence, soulignant ainsi différentes parties du son source. Nous pouvons provoquer ce balayage de fréquence simplement en faisant varier le temps de retard.

Impulsion à bande limitée

Les effets d'un filtre sont particulièrement perceptibles lorsque le son source contient de nombreuses fréquences différentes, qui peuvent être modifiées par le filtre. Si nous voulons appliquer un filtre en peigne à un son aigu avec un spectre harmonique, il est logique d'utiliser un son comportant de nombreux partiels, comme une onde en dents de scie ou une onde carrée.

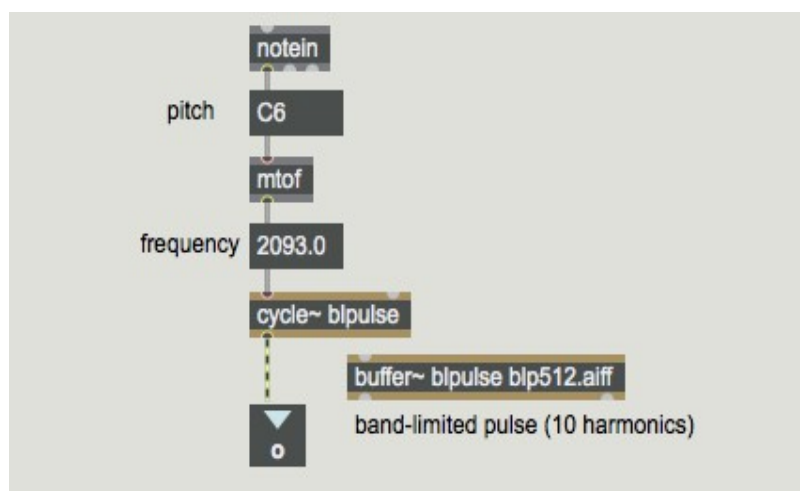


Ces ondes mathématiquement idéales sont peut-être trop "parfaites" pour être utilisées comme ondes sonores

Le problème avec ces formes d'onde dérivées mathématiquement, cependant, est qu'elles peuvent être trop riches en partiels élevés. Elles peuvent avoir des partiels au-dessus du taux de Nyquist qui sont suffisamment forts pour provoquer un aliasing inharmonique. (Ce problème est traité plus en détail dans le *didacticiel de filtrage 3*.)

Pour ce tutoriel, nous utilisons une forme d'onde appelée **impulsion à bande limitée**. Une impulsion à bande limitée a un spectre harmonique avec une énergie égale à tous les harmoniques, mais a un nombre limité d'harmoniques afin d'éviter les repliements. La forme d'onde utilisée dans ce patch de tutoriel a dix harmoniques d'énergie égale, de sorte que sa composante de fréquence la plus élevée a dix fois la fréquence de la fondamentale.

Cela signifie que nous pouvons l'utiliser pour jouer des fréquences fondamentales jusqu'à 2 205 Hz si notre taux d'échantillonnage est de 44 100 Hz. (son harmonique la plus élevée aurait une fréquence de 22 050 Hz, ce qui est égal au taux de Nyquist.) Comme la touche la plus élevée d'un clavier MIDI à 61 touches joue une fréquence de 2 093 Hz, cette forme d'onde ne provoquera pas de repliement si nous l'utilisons comme limite supérieure.

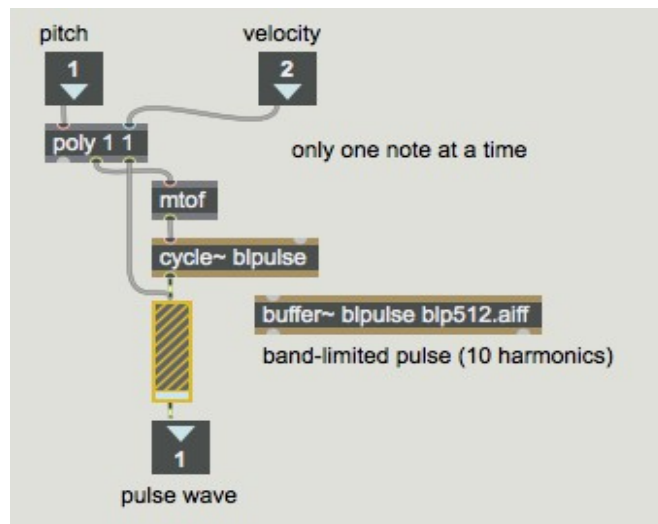


Jouer une onde d'impulsion limitée en bande avec MIDI

Détail technique: dans une onde pulsée idéale (optimalement étroite), chaque cycle de la forme d'onde consisterait en un seul échantillon de valeur 1, suivi de tous les échantillons à 0. Cela créerait un spectre harmonique avec toutes les harmoniques à la même amplitude, continuant vers le haut à l'infini. Il est possible de créer un réseau de signaux MSP qui calcule – à partir de la fréquence fondamentale et de la fréquence d'échantillonnage - un signal impulsif à bande limitée contenant le nombre maximal d'harmoniques possibles sans repliement. Dans ce cas, cependant, nous avons choisi d'utiliser simplement une forme d'onde stockée contenant dix partiels.

Conversion vitesse/amplitude: *gain* ~

Le sub-patch **pPulse_Wave** contient un moyen simple mais efficace de jouer un son dans MSP via MIDI. Il utilise un objet *poly* pour implémenter le vol de voix, limitant les notes MIDI entrantes à une note à la fois. (Il désactive la note précédente en l'envoyant avec une vitesse de 0 avant de jouer la note entrante.) Il utilise ensuite *mtof* pour convertir le numéro de note MIDI en valeur de fréquence correcte pour MSP, et il utilise l'objet MSP *gain* ~ pour mettre à l'échelle l'amplitude du signal en fonction de la vitesse MIDI.



Conversion des données MIDI de hauteur et de vélocité en informations de fréquence et d'amplitude pour MSP

L'objet *gain* ~ prend à la fois un signal et un nombre dans son entrée gauche. Le nombre est utilisé comme un facteur d'amplitude par lequel le signal est mis à l'échelle avant d'être l'envoyé. Une caractéristique particulière de *gain* ~ (en dehors de son utilité en tant qu'objet d'interface utilisateur pour la mise à l'échelle d'un signal) est qu'il peut convertir les nombres entrants d'une progression linéaire en une courbe logarithmique ou exponentielle. Ceci est très approprié dans ce cas, car nous voulons convertir la plage de vélocité linéaire (0 à 127) en une courbe d'amplitude exponentielle (0 à 1) qui correspond à peu près à la façon dont nous entendons l'intensité sonore. Chaque changement de vélocité de 10 correspond à un changement d'amplitude de 6 dB. L'autre caractéristique utile du *gain* ~ est que, plutôt que de changer brusquement d'amplitude lorsqu'il reçoit un nouveau nombre dans son entrée gauche, il prend quelques millisecondes pour progresser graduellement vers le nouveau facteur d'amplitude. Le temps nécessaire à cette progression peut être spécifié en envoyant un temps, en millisecondes, dans l'entrée droite. Dans ce patch, nous utilisons simplement le temps par défaut de 20 ms.

- Choisissez l'un des paramètres d'exemple prédéfinis, et choisissez 'Pulse Wave' dans le menu déroulant 'Sound Source'. Jouez de longues notes avec le clavier MIDI. Vous pouvez également obtenir un son continu à n'importe quelle amplitude et fréquence en envoyant des nombres à partir des objets de boîte de *nombres* de hauteur et de vélocité (d'abord la vélocité, puis la hauteur) dans les entrées du sub-patch **pPulse_Wave**.

Varier les paramètres du filtre

Comme l'illustre ce patch, il est généralement préférable de modifier les paramètres d'un filtre en utilisant un signal changeant progressivement plutôt que d'effectuer un changement abrupt avec un seul chiffre. Ainsi, les changements de paramètres effectués sur l'objet *comb* ~ 'Ajusted By Hand' sont tout d'abord envoyés à un objet *line* ~ pour interpolation sur un temps de 25 ms.

L'objet *comb* ~ «Modulated» voit son temps de retard varier à basse fréquence en fonction de la forme de l'onde pulsée à bande limitée (juste parce que c'est une forme plus intéressante qu'une simple sinusoïde). La modulation pourrait en fait être effectuée par un signal variable de n'importe quelle forme. Vous pouvez faire varier le taux de cette modulation en utilisant la molette de modulation de votre synthé (ou tout simplement en faisant glisser la boîte de *nombre*). Le gain des retards x et y (les deux entrées les plus à droite) est modulé par une onde sinusoïdale allant de 0,01 à 0,99 (pour le gain de rétroaction) et une onde cosinusoïdale allant de 0,01 à 0,49 (pour le gain de

feedforward). Lorsque l'amplitude de l'un augmente, l'autre diminue.

- L'expérimentation de différentes combinaisons de valeurs de paramètres peut vous donner des idées pour d'autres types de modulation que vous pourriez vouloir concevoir dans vos propres patches.

Résumé

L'objet *comb* ~ vous permet d'utiliser des délais de rétroaction très courts pour **filtrer** un signal **en peigne**. Un filtre en peigne crée des augmentations et des diminutions d'amplitude en fonction de la fréquence dans le signal qui le traverse, à des intervalles de fréquence régulièrement espacés (c'est-à-dire liés aux harmoniques). L'intervalle de fréquence est déterminé par l'inverse du temps de retard. Le filtre en peigne est particulièrement efficace lorsque le temps de retard (et donc l'intervalle de fréquence) varie dans le temps, ce qui accentue différentes régions de fréquence dans le signal filtré.

L'objet d'interface utilisateur *gain* ~ est utile pour mettre à l'échelle l'amplitude d'un signal selon une courbe logarithmique ou exponentielle spécifique. Les changements d'amplitude causés par *gain* ~ s'effectuent progressivement sur une certaine durée (20 ms par défaut), de sorte qu'il n'y a pas de discontinuité soudaine indésirable dans le signal de sortie.