TP4 traitement et synthèse modulaire du signal audio
Annexe B : exemples de patch

notions : toutes les notions du cours

Ce document présente quelques patchs pour le TP4 “synthèse et traitement modulaire du signal audio”. Il ne s’agit que d’exemples pour commencer à tester votre code ! vous pouvez bien sûr réaliser vos propres patchs. **Pensez à amener un casque audio doté d’une prise minijack en séance !**

Table des matières

Première partie

Annexe : exemples de patchs

1 Légendes des symboles utilisés dans ce document

La Figure 1 montre les symboles utilisés dans les schémas des pages suivantes.

Lorsque c'est utile, les schémas des patchs des pages suivantes font apparaitre l'ordre d'exécution des modules par un numéro en bleu.

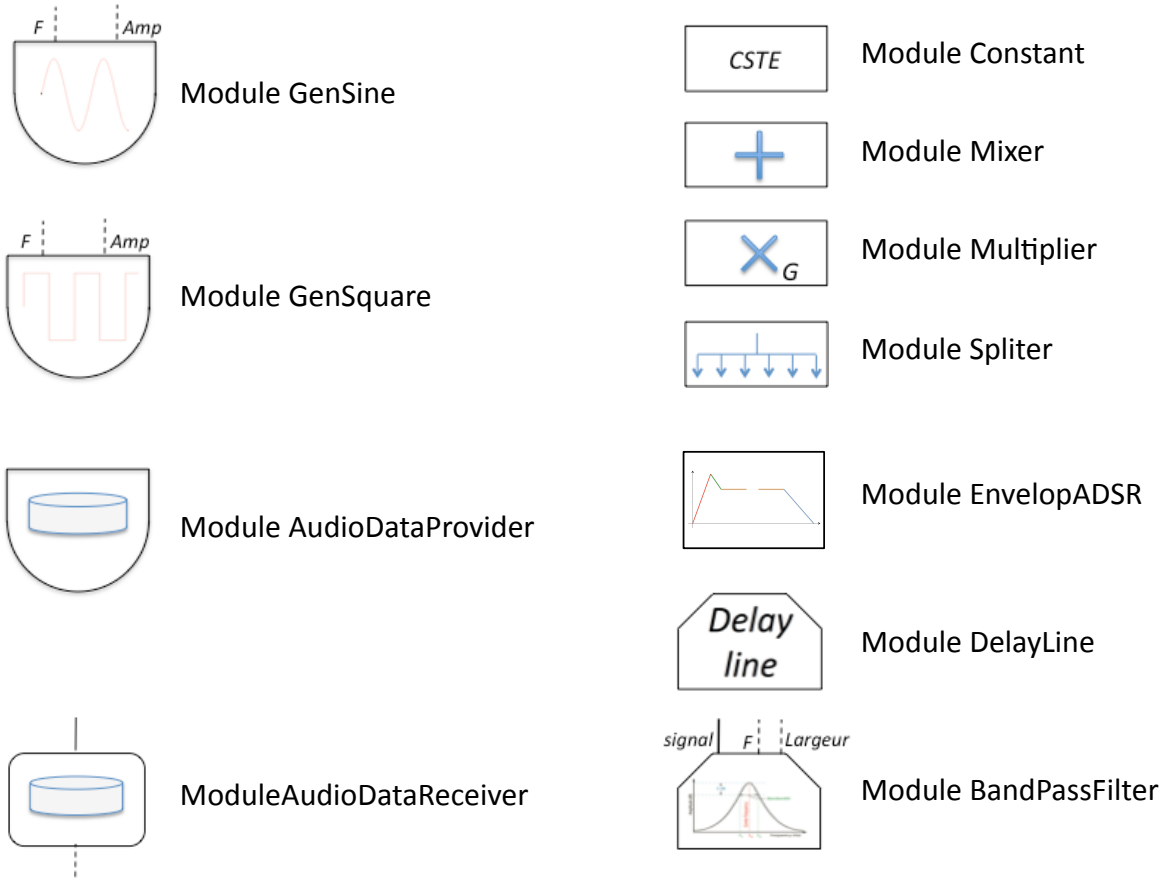

FIGURE 1 – Symboles utilisés dans ce document pour les modules

2 Un accord basique

Modules utilisés par ce patch : GenSine (ou GenSquare), Mixer, AudioDataReceiver.

Le patch : voir schéma le de la Figure 2.

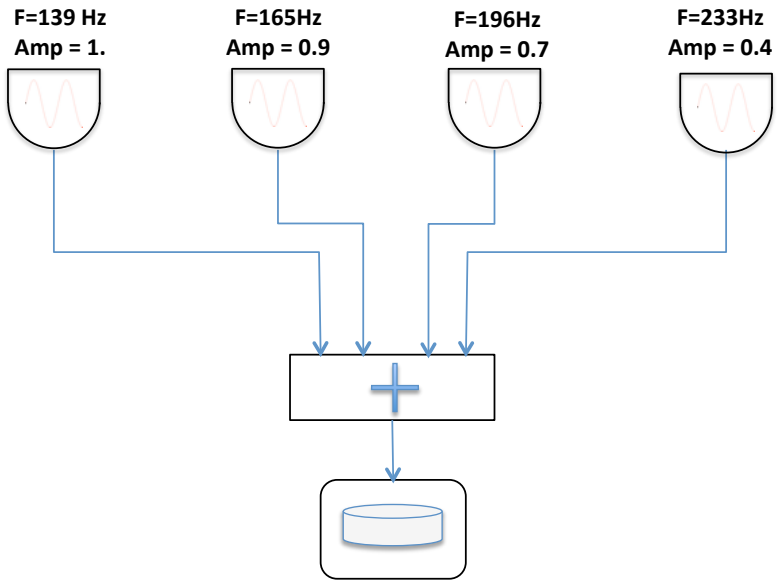

FIGURE 2 – Patch pour un accord basique

Piste d'évolutions : Pour jouer un autre accord, il suffit bien sûr de modifier les fréquences des sinus. On peut ajouter d'autres générateurs pour complexifier l'accord.

Si on utilise deux oscillateurs seulement, de fréquences très proches (par exemple 440 Hz et 445 Hz), on entendra des battements.

Autres pistes : changer les GenSine en GenSquare ; faire varier les amplitude au cours du temps, par exemple au moyen de GenSine basse fréquence (eg : $0.5Hz$) ; Ajouter un module EnvelopADSR et un Multiplier pour contrôler l'enveloppe du son ; etc.

3 Echo

Modules utilisés par ce patch : AudioDataProvider, DelayBasic, Mixer, Spliter, Multiplier, AudioDataReceiver.

Le patch : voir schéma le de la Figure 3.

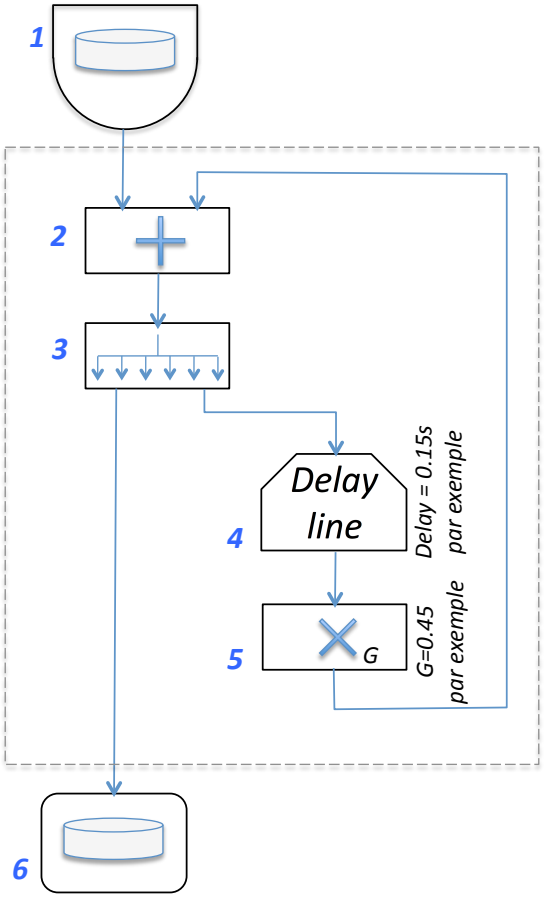

FIGURE 3 – Diagramme UML des premières classes

Piste d'évolutions : Expérimenter d'autres durées de délai ou d'autres valeurs du gain.

Sur le schéma, la partie encadrée de ce patch se prête bien à la réalisation d'un macro-module d'écho (cf. partie 7 du chapitre 2 du sujet). Pour cela, une solution propre pourrait être d'écrire une nouvelle classe `Echo` sous classe de la classe `MacroModule`, dont le constructeur crée le patch contenu.

Il est possible de réaliser une "réverb" (effet de salle) élémentaire en montant en parallèle plusieurs "échos" (une bonne dizaine par exemple) dont les délais et gains sont différents. Un tel patch peut être vu comme un modèle (naïf, certes !) des réverbérations qui apparaissent entre les différents murs d'une salle hypothétique.

4 Modulation de la fréquence d'un sinus et Synthèse FM

Le patch présenté dans cette section se propose de moduler la fréquence d'un générateur sinusoïdal par un autre sinus.

Ce patch peut avoir deux effets bien distincts, en fonction de la valeur des paramètres utilisés : soit il fait entendre une note dont on modifie la fréquence fondamentale au cours du temps, soit il fait entendre une seule note avec un timbre (spectre) complexe, généré par "modulation de fréquence".

Modules utilisés par ce patch : GenSine, Constant, Mixer, AudioDataReceiver.

Le patch : voir schéma le de la Figure 4.

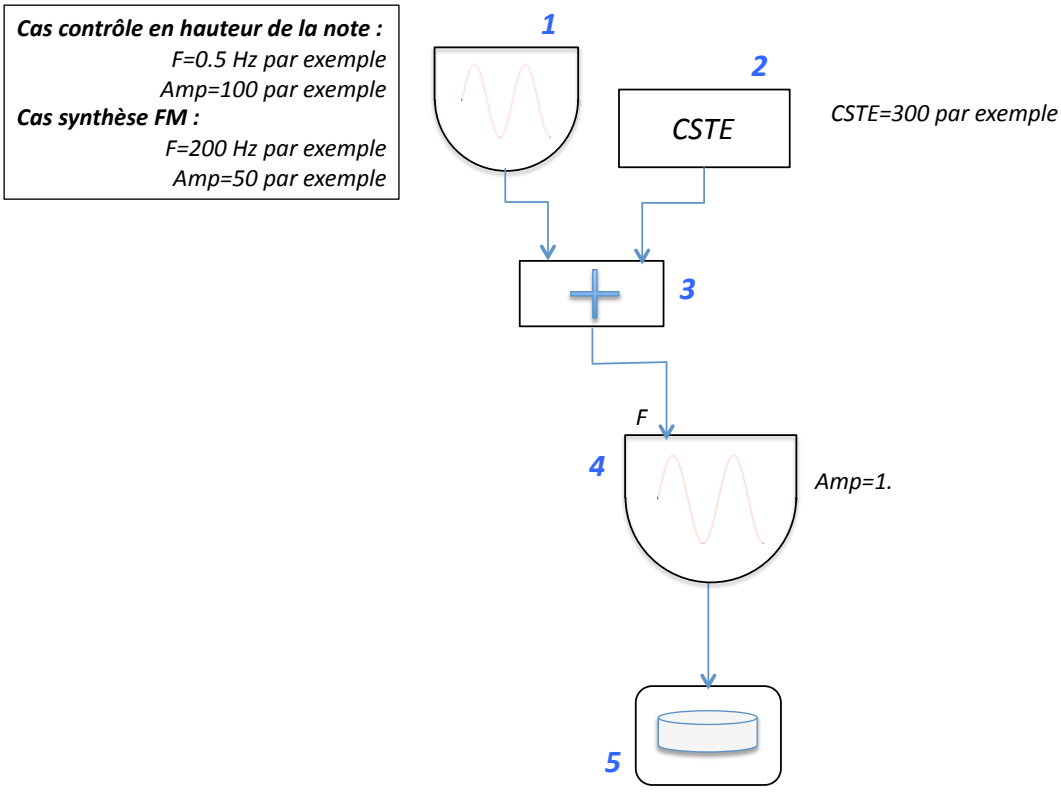

FIGURE 4 – Patch de modulation de fréquence

Remarques : Si la fréquence fondamentale du premier GenSine de contrôle est basse (par exemple moins de 5 Hz, c'est à dire "à l'échelle des évènements musicaux"), ce patch fait entendre une note dont la hauteur varie au cours du temps.

Si par contre la fréquence fondamentale du premier GenSine de contrôle est haute (c'est à dire : dans le domaine audio, par exemple 200 Hz), ce patch réalise une 'synthèse par modulation de fréquence" (synthèse FM) [1].

La synthèse FM est une méthode de synthèse numérique du signal qui a connu un très très grand succès dans les années 80. Elle fut au coeur de nombreux synthétiseurs *hardware* de cette époque et a

1. Référence : CHOWNING (J.) - The synthesis of complex audio spectra by means of frequency modulation, in Fundation of computer music, Roads (C.) et Strawn (J.) Eds., Cambridge, Massaschussets : MIT Press, 1985.

ensuite beaucoup marqué les timbres de la musique pop. L'une des raisons tient au fait qu'avec très, très peu de moyens (seulement quelques oscillateurs !), elle permet de générer des timbres déjà riches et complexes. On peut noter, par contre, que le contrôle de la synthèse FM (ou : "comment générer des sons intéressants par synthèse FM") est particulièrement délicat. Il y a là un véritable savoir-faire !

Piste d'évolutions : En mode "contrôle basse fréquence", remplacer le GenSine par un GenSquare - ou par un patch de synthèse déjà plus complexe, par exemple de 'synthèse additive".

En mode "synthèse FM" :
— ajouter une enveloppe ADSR sur l'entrée d'amplitude du second sinus, pour obtenir quelque chose qui ressemble déjà plus à une "note" plutôt qu'un son de durée infinie.
— essayez avec d'autres valeurs pour les fréquences des deux GenSine et la valeur de la constante.

Autres pistes d'expérimentation : A votre avis, que se passe-t-il si, au lieu de moduler la fréquence d'un sinus, on contrôle son *l'amplitude*par exemple en fonction d'un signal préexistant (par exemple : une musique, ou un texte enregistré) ?

5 Filtrage passe-bande modulé dans le temps

Modules utilisés par ce patch : BandPassFilter, AudioDataProvider ou GenWhiteNoise (générateur de bruit blanc), GenSine, Mixer, AudioDataReceiver.

Pour commencer à tester le module BandPassFilter, on pourra d'abord utiliser une fréquence de résonance du filtre fixe, par exemple fixée à $500Hz$.

Le patch : voir schéma le de la Figure 5.

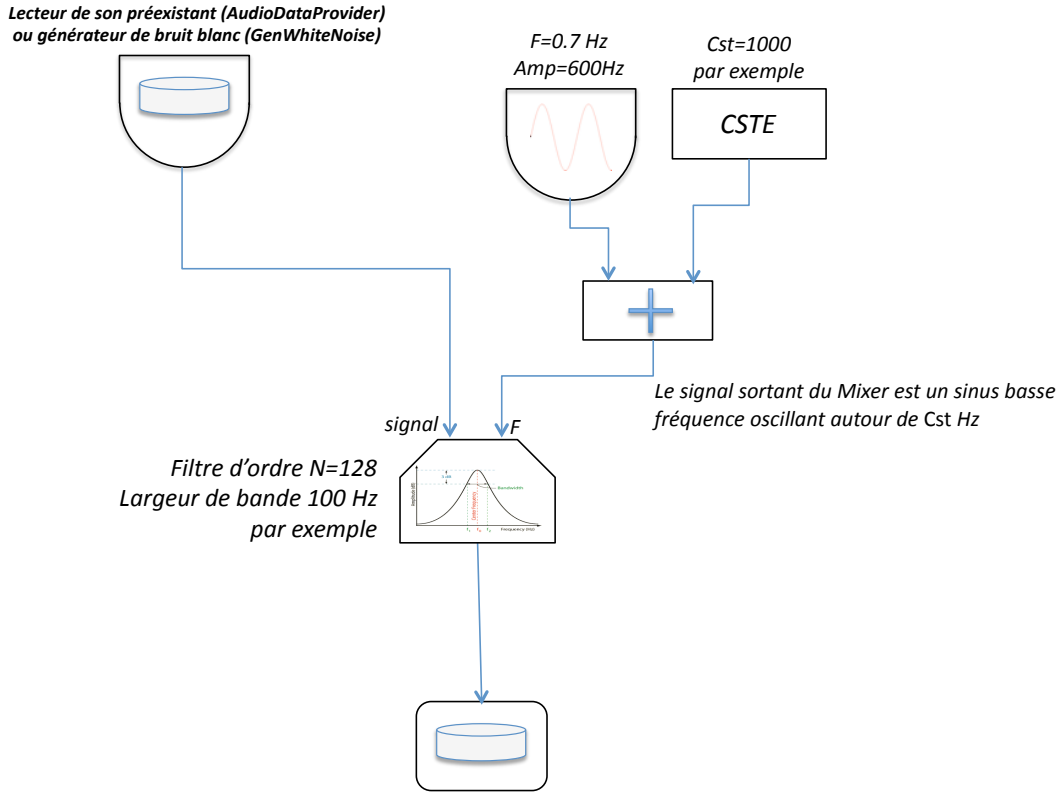

FIGURE 5 – Contrôle de la fréquence de résonance d'un filtre passe-bande au cours du temps

Piste d'évolutions : Bien évidemment, la façon dont la fréquence de résonance est modifiée au cours du temps (et/ou la largeur de bande) impacte beaucoup le résultat sonore. Une variation suivant un simple sinus devient assez rapidement ennuyeuse à l'oreille... D'autres variations sont intéressantes à tester : linéaire, exponentielle, somme de sinus, etc.

Pour tester cela, on pourra :

— Approche basique : se débrouiller avec un logiciel d'édition audio quelconque, par exemple **Audacity**, pour génrer des fichiers audio .wav qui ne contiennent pas un son, mais la forme de la variation qu'on veut pour la fréquence de résonance du filtre ; puis lire cette "forme" au moyen d'un module `AudioDataProvider` connecté au port "fréquence de résonance" du filtre.

— Ou mieux : introduire de nouveaux types de modules : générateurs de rampe, générateur qui interpole entre une série de points de contrôle par exemple lus dans un fichier texte ; etc.

Il peut être aussi intéressant de faire varier également la largeur de bande du filtre au cours du temps.

6 Synthèse additive d'un son harmonique

Modules utilisés par ce patch : Constant, GenSine, Multipler, Mixer, AudioDataReceiver ; amélioration : EnvelopADSR.

Le patch : voir schéma le de la Figure 6.

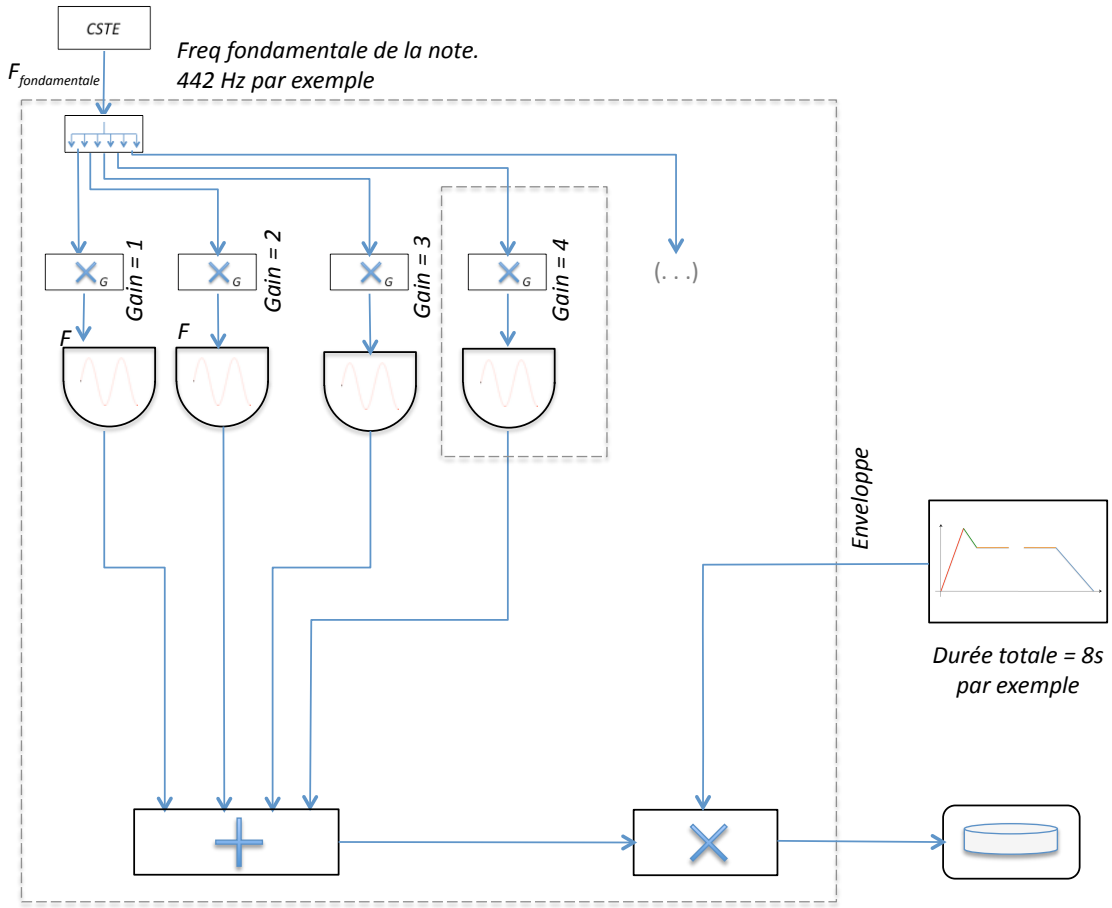

FIGURE 6 – Synthèse additive basique d'un signal harmonique

Remarques : La méthode de synthèse additive du signal sonore consiste à générer le signal final en additionnant des contributions sinusoïdales dont on contrôle les fréquences et amplitudes. La théorie de Joseph Fourrier montre que cela permet de synthétiser tous les sons périodiques.

L'approche a ceci d'intéressant qu'elle permet de maîtriser finement, et relativement simplement, l'amplitude de chaque composante sinusoïdale du son (appelée partiel dans la méthode), propriétés auxquelles l'oreille est très sensible, et ainsi d'une certaine manière de "sculpter" le spectre du son.

Pour aller vers des sons plus intéressants par synthèse additive, il devient nécessaire de pouvoir faire varier dans le temps, à basse fréquence, l'amplitude - et parfois la fréquence - de chaque partiel. Une difficulté de l'approche tient alors au fait qu'elle nécessite de contrôler un grand nombre de paramètres : si on utilise par exemple plus de 100 oscillateurs (100 partiels), il faut contrôler dans le temps les règles d'évolution des 100 amplitudes et des 100 fréquences. Une solution est alors de travailler, en amont du patch de synthèse, des lois (ou "règles") contrôlant tout cela à partir d'un ensemble de paramètres de commande plus réduit.

Le “catalogue de sons synthétisés” de Jean Claude Risset de 1969, considéré comme fondateur dans le domaine de la synthèse de signal sonore, démontre qu'un type de son (un timbre sonore) est lié à des règles qui lui sont spécifiques. Ce catalogue a introduit plusieurs patchs de ce type [2].

Dans ce catalogue, on peut s'intéresser par exemple à la synthèse additive d'un son évoquant la trompette. Risset a montré que, pour évoquer la famille des cuivres (trompette, saxophone...), il suffit que les amplitudes des partiels croissent de façon non linéaire en fonction de la puissance du son : plus le son est puissant, plus les partiels de fréquence élevée sont présents dans le spectre. Cette étude de Risset a ainsi démontré *par synthèse* que cette propriété est une signature essentielle de ce type de son (*analyse*). On parle de méthode “d'analyse par synthèse”.

Un autre exemple remarquable est celui du son paradoxal de glissando infini dit de “Shepard-Risset”.

Piste d'évolutions : Dans le patch de la figure 6, seule la fréquence des partiels est contrôlée, et elle est de plus fixée à un multiple de la fondamentale : le patch est donc limité à la synthèse d'un son purement harmonique et encore très simple.

Un patch de synthèse additive plus complet devrait permettre de choisir le nombre de partiels, les fréquences de chaque partiel et les amplitudes de chaque partiel. Il devrait permettre également de faire varier les amplitudes, voire les fréquences, au cours du temps. Ainsi, il deviendrait possible de travailler plus finement le spectre du son de synthèse additive.

L'approche se prête bien à la mise en oeuvre de macro-modules (voir les deux rectangles en pointillés sur le schéma de la figure 6). Une mise en oeuvre possible consiste à écrire une sous-classe de la classe `MacroModule`, dont le constructeur prend par exemple en paramètre un tableau de fréquences et un tableau d'amplitudes et crée le patch interne en conséquence. Un tel macro-module peut aussi avoir des ports d'entrée permettant de faire varier fréquences et amplitudes dans le temps.

2. Risset Catalogue de sons synthétisés. 1969 ; réédition Wergo 1995. Quelques exemples audio : https ://www.youtube.com/watch ?v=zXDaBA3aggI catalogue risset

7 Jouons quelques notes...

Modules utilisés par ce patch : Constant, GenSquare (ou GenSine), EnvelopADSR, Mixer, DelayBasic, AudioDataReceiver.

Le patch : voir schéma le de la Figure 7.

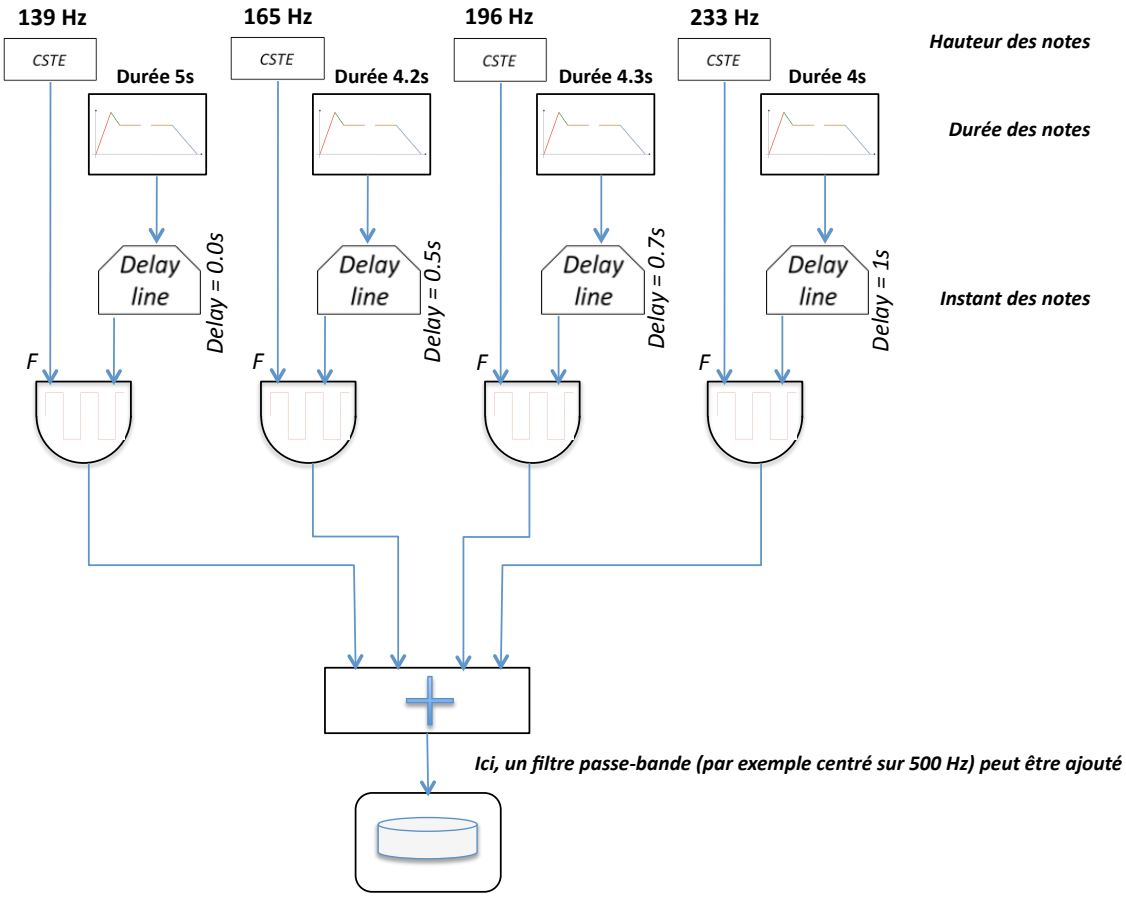

FIGURE 7 – Synthèse additive basique d'un signal harmonique

Remarques et piste d'évolutions : L'approche réellement adaptée pour "faire jouer un patch à un instant donné" consiste bien sûr à mettre en oeuvre un ordonanceur (*Scheduler*), comme proposé dans la section 8 du chapitre 2.

Supposant que cet ordonnanceur n'est pas encore implanté, ce patch propose d'utiliser des délais pour choisir à quelle instant la "note" doit être jouée (c'est à dire des délais qui contrôlent à quel instant l'enveloppe ADSR sera délivrée sur le port d'amplitude de l'oscillateur). Cette approche est certes fort peu pratique, et fort peu économique, mais elle vous permettra de commencer à jouer quelques "partitions musicales" de votre composition même sans ordonnanceur.

8 Références

CHOWNING (J.) - The synthesis of complex audio spectra by means of frequency modulation, in Fundation of computer music, Roads (C.) et Strawn (J.) Eds., Cambridge, Massaschussets : MIT Press, 1985

C. Roads, S.T. Pope, A. Piccialli, G. De Poli, Musical Signal Processing, Lisse, Swets et Zeitlinger, pp. xii477, 1997.

G. De Poli, A. Piccialli, C. Roads, editors, Representations of music signals, MIT Press, Cambridge, MA, 494 pp, 1991.

Anne Veitl : LE LOGICIEL MUSIC V, TECHNOLOGIE D'ECRITURE MUSICALE : RAPPELS HISTORIQUES ET ELEMENTS D'ANALYSE. Journées d'Informatique Musicale JIM 2009, Grenoble. http://jim.afim-asso.org/jim09/downloads/actes/JIM09_Veitl.pdf

Risset Catalogue de sons synthétisés. 1969 ; réédition Wergo 1995. Quelques exemples audio : https://www.youtube.com/watch?v=zXDaBA3aggI