INTERVALLES MUSICAUX &
SYNTHÈSE PAR MODULATION D'AMPLITUDE OU DE FRÉQUENCE
ELEMENTS DE CONTRÔLE NUMÉRIQUE DU SIGNAL

Grégoire Bonnet

(2020)

Introduction

On propose de détailler l'écriture d'intervalles musicaux se rapportant aux rapports des partiels à la porteuse dans le cas des modulations d'amplitude ou de fréquence.
Ces rapports reflètent des lois constituant la particularité de ces timbres et on suppose la maîtrise de leur écriture justifiée dans le cadre d'une écriture compositionnelle se référant à la synthèse et au médium, électronique ou numérique, les rendant possibles.
On détaillera ces rapports et on s'intéressera notamment à produire des inversions spectrales.
En effet on considère ce procédé comme directement lié aux principes de ces synthèses tout en restituant à la fois des principes d'inversions de rapport, ou symétrie d'intervalle, justifiés dans le cadre d'une volonté d'étendre le caractère de ces synthèses à des lois d'écritures, qui tout en leur étant liées par leur constitution, n'en demeurent pas moins libres et détachées de la matière originelle qui leur a donné naissance.
On proposera, à cet égard, deux modèles de synthèses librement inspirés de ces principes.
On propose également un modèle d'écriture des durées, en rapport avec les hauteurs, et cohérent avec ce qui aura été décrit précédemment.
Enfin, on proposera une implémentation objet de l'écriture de séquences (se rapportant, ou non, aux principes mentionnés plus haut) et une autre destinée à la manipulation des données de séquences entre elles.

1. HAUTEURS

1.1 Rapports et intervalles musicaux

Un *intervalle* de deux sons est représenté par un rapport de fréquences.

Soit deux fréquences f_1 et f_2 ($f_1 < f_2$), k le *ratio* :

$$k = \frac{f_2}{f_1}$$

Un *intervalle do-mi,* de tierce, correspond à un rapport de 5/4.

- En tempérament égal *do=261.6255* et *mi=329.6275* ont un rapport *k=1.2599*. Ou encore: une tierce correspondant à 4 demi-tons (chacun étant une racine douzième de 2): $2^{\frac{4}{12}} = 2^{\frac{1}{3}}$.
- En intonation juste: on considère *mi* comme la 5ème harmonique d'une fréquence *do* fondamentale *f0* (soit cette fréquence fondamentale *f0* multipliée par 5), *do* représentant l'harmonique puissance de deux (à l'octave) la plus proche soit la 4ème harmonique de cette fréquence fondamentale *f0*:
  k = 5.f0 / 4.f0 = 5 / 4 = 1.25

1.2 Principes de composition des fréquences

Deux fréquences f_1 et f_2 font apparaitre leur somme et leur différence:

$$cos(f_1)cos(f_2) = \frac{1}{2}[cos(f_1 + f_2) + cos(f_1 - f_2)]$$

Dans une modulation de signal (AM, FM), la fréquence *porteuse* (*carrier*) f_c est modulée par une fréquence *modulante* f_m produisant de part et d'autre de la *porteuse* deux partiels de fréquences f_{p0} et f_{p1} telles que[1]:

$$\begin{cases} f_{p0} = f_c - f_m \\ f_{p1} = f_c + f_m \end{cases}$$

On remarque que le rapport de chaque partiel à la fréquence porteuse est distinct:

$$\frac{f_c}{f_{p0}} \neq \frac{f_{p1}}{f_c}$$

Alors qu'il y a une équidistance, à l'échelle des fréquences, de ces partiels par rapport à la porteuse.

$$f_c - f_{p0} = f_{p1} - f_c = f_m$$

En effet f_{p1} est le symétrique de f_{p0} par rapport à la porteuse f_c :

$$f_{p1} = 2.f_c - f_{p0}$$

1.3 Rapports

Les rapports de ces partiels de part et d'autre des deux fréquences f_c et f_m peuvent être inversés ; produisant un miroir, l'un de l'autre, par rapport à l'une ou l'autre de ces fréquences.

Soit une fréquence f, le symétrique f' de cette fréquence f par rapport à une autre fréquence f_c :

$$\frac{f'}{f_c} = \frac{f_c}{f} \qquad \text{soit} \qquad f' = \frac{f_c^2}{f}$$

Dans une modulation avec une porteuse f_c et une modulante f_m, faisant apparaitre des partiels f_{p0} *et* f_{p1} tels que décrits précédemment, on obtient comme les partiels f'_{p0} *et* f'_{p1}, symétrique de f_{p0} *et* f_{p1} par rapport à f_c:

[1] On simplifie le modèle aux premiers partiels apparaissant de part et d'autre de la porteuse. On peut prédire plus généralement l'amplitude des partiels dans ces timbres: M. Puckette, *The theory and technique of electronic music*, Chapitre 5.

$$\begin{cases} f'_{p0} = \frac{f_c^2}{f_{p0}} = \frac{f_c^2}{f_c - f_m} \\ f'_{p1} = \frac{f_c^2}{f_{p1}} = \frac{f_c^2}{f_c + f_m} \end{cases}$$

1.4 Cas particuliers

- <u>*Valeur pour laquelle le symétrique de la somme de deux fréquences est égal à leur différence*</u>

Soit *x* le rapport de ces deux fréquences, pour que le symétrique de la somme de ces deux fréquences soit égal à leur différence il faut que

$$\frac{1}{x+1} = x - 1 \qquad \text{soit} \qquad x^2 - 2 = 0$$

On a deux racines

$$\begin{cases} a = \sqrt{2} \\ b = -\sqrt{2} \end{cases}$$

- <u>*Cas du Nombre d'Or*</u>

Soit ω_0 *et* ω_1 racines de l'équation $x^2 = x + 1$:

$$\begin{cases} \omega_0 = \frac{\sqrt{5} - 1}{2} \\ \omega_1 = \frac{\sqrt{5} + 1}{2} \end{cases}$$

avec

$$\omega_0 = \omega_1^{-1} \qquad \text{et} \qquad \omega_0 = \omega_1 - 1$$

Soit ω_1 le *nombre d'or* ($\omega_1 = 1.618$, soit un intervalle de sixte), *s*i la porteuse f_c et la modulante f_m sont telles que:

$$f_c = \omega_1 . f_m$$

alors

$$\begin{cases} f_{p0} = f_m . (\omega_1 - 1) = \omega_0 . f_m \\ f_{p1} = f_m . (\omega_1 + 1) = \omega_1^2 . f_m = \omega_1 . f_c \end{cases}$$

Les partiels symétriques sont :

$$\begin{cases} f'_{p0} = \frac{f_c^2}{f_{p0}} = \frac{f_c^2}{\omega_0 \cdot f_m} = \omega_1^2 . f_c \\ f'_{p1} = \frac{f_c^2}{f_{p1}} = \frac{f_c^2}{\omega_1 \cdot f_c} = \omega_0 . f_c \end{cases}$$

On obtient une répartition à intervalles égaux.

2. DURÉES

2.1 Dynamique, rapport

Une fréquence est l'inverse d'une durée

$$[hertz] = [sec.]^{-1}.$$

Si deux fréquences f_0 et f_1 ont un rapport k tel que

$$f_1 = k \, . f_0$$

alors leurs longueurs d'ondes respectives sont en rapport inverse

$$L_1 = k^{-1} \, . \, L_0.$$

On peut intégrer ce rapport.

$$\int \frac{1}{x} \, dx = log \, |\, x \,| + C \text{ avec } x \neq 0$$

avec $k > 0$

$$log(f_1) + C = log(k \, . f_0) + C = log(k) + log(f_0) + C = log(k \, . \, e^C) + log(f_0)$$

et

$$|\, log(k) \,| = |\, log(k^{-1}) \,| \, .$$

Un ensemble de hauteurs peut se rapporter dynamiquement à un ensemble de durées en ramenant celles-ci proportionnellement à leurs rapports, ou à leurs variations de rapports. Cela induit une relation directe entre un ensemble de hauteurs et leurs durées respectives. Une hauteur se trouve ainsi comme amplifiée, ou appuyée, par sa durée ; et ce de façon relative à d'autres. Par extension l'espace de perception de ces hauteurs devient un espace de perception de géométrie différentielle.

2.2 Interprétation

Intégrer le rapport d'une harmonique à sa fondamentale revient à prendre le logarithme de ce rapport.
Une interprétation géométrique de l'intégration correspond à l'aire d'une surface déterminée par la variation de ce rapport.

Attribuer à une durée l'intégrale de la variation du rapport d'une harmonique à sa fondamentale (ou plus généralement d'une fréquence à une autre fréquence de référence) revient à faire entendre, dans le temps, la valeur, ou l'intensité, d'une fréquence par rapport à une autre.

On peut relier le rapport d'une fréquence à une autre fréquence à sa durée. Ce faisant, on induit une forme d'écoute du temps plus complète du fait de la perception simultanée d'un temps court, la fréquence, et d'un temps long, correspondant au rapport de ce temps court avec un autre temps court, la fréquence de référence ou fondamentale.
On produit simultanément un rapport de résonance mis en relation avec la durée de perception de ce rapport.

scalaires / durées / rapports / fréquences

Cette durée relative au rapport peut être ce rapport lui-même ou encore la valeur de l'intégration de ses variations.

Par extension nous percevons des perceptions d'espace, en durée, relativement à la variation du rapport d'une fréquence, d'une note, à sa fondamentale , ou note de référence.

Ou autrement dit, nous percevons des variations d'espace, en valeur de durée, relativement à la résonance d'une note par rapport à une autre.

Il s'agit de montrer qu'en prenant pour valeur de durée le logarithme du rapport d'une fréquence à une autre, une durée étant toujours positive, le logarithme d'un rapport et celui d'un rapport inversé sont égaux en valeur absolue.
En suggérant un tel choix, nous induisons l'idée que le rapport d'une note à une autre (d'une harmonique à sa fondamentale), est équivalent [et du moins égal dans la valeur absolue de sa valeur intégrée] , à son rapport inversé et donc au rapport d'une note à une autre note selon un rapport inversé. Une harmonique devenant ainsi une subharmonique de même éloignement.

Cela étant avancé, nous en tirons l'équivalence du point de vue de la perception des durées, d'une résonance à une résonance inversée. Soit d'une harmonique à une subharmonique.

A cet égard, le phénomène physique de résonance n'a pas d'équivalent dans la production sonore électronique. Celle-ci n'est qu'une reproduction, se référant théoriquement au modèle de décomposition de Fourier.
De la même manière, le modèle numérique, au delà de son attachement à reproduire des sons liés aux phénomènes de *résonance*, paradoxalement, restitue de façon plus complète, plus parfaite, une fréquence dont la fréquence d'échantillonnage est un multiple entier.
Ainsi l'expression d'une fréquence dans le cadre du modèle numérique correspond à une résonance plus ou moins harmonique de la fréquence d'échantillonnage selon un rapport inversé.

Nous justifions ici l'emploi d'un tel système d'écriture de hauteurs et de leurs durées (comme décrits précédemment) dans le cadre d'un système numérique à même de rendre compte, à la fois de phénomènes résonants en rapport à une fondamentale, liés à la physique et au modèle de la transformée de Fourier, *et* des phénomènes propres liés au statut d'une fréquence échantillonnée, tel que décrit essentiellement par le théorème de Shannon-Nyquist.

Cela produit une forme d'homogénéisation des lois sonores dans le cadre de leur production numérique.

2.3 Perception d'une durée, relativement à une hauteur, et réciproquement, comme géométrie différentielle

En unité

$$[freq] = [dur]^{-1}$$

Une *fréquence f* est de la forme

$$f = k_f \,.\, f_0 \qquad \text{avec} \qquad k_f > 0$$

Si $k_f > 1$, *rapport harmonique* (si k_f entier naturel f *harmonique*)
Si $k_f = 1$, f est la fondamentale f_0
Si $k_f < 1$, *rapport subharmonique* (si k_f de la forme $\frac{1}{q}$ avec q entier naturel, f *subharmonique*)

Une durée *d* est de la forme

$$d = k_d \,.\, d_0 \qquad \text{avec} \qquad k_d > 0$$

On peut associer, par une fonction t, une durée d à une fréquence f, fonction du rapport k_f de cette dernière à une autre fréquence ou fondamentale f_0

$$d = t(f) = t(f, f_0) = t(k_f)$$

Réciproquement, par une fonction t, on peut associer une fréquence f à une durée d, fonction du rapport k_d de cette dernière à une autre durée de référence d_0

$$f = t(d) = t(d, d_0) = t(k_d)$$

On remarque que

$$ln(k_f) + C = \int_{f_0}^{f} \frac{1}{x}\, dx$$

et

$$ln(k_d) + C = \int_{d_0}^{d} \frac{1}{x}\, dx$$

Et aussi que

$$ln(k) = -\, ln(k^{-1}), k > 0$$

donc

$$ln(k_f) + C = -\, ln(k_f^{-1}) + C = \int_{f_0}^{f} \frac{1}{x}\, dx = -\int_{f}^{f_0} \frac{1}{x}\, dx$$

Une durée étant toujours positive,

si

$$d = t(k_f) = ln(k_f) + C = \int_{f_0}^{f} \frac{1}{x}\, dx$$

alors

$$d = t(k_f) = t(k_f^{-1}) = |\, ln(k_f) + C \,| = |\int_{f_0}^{f} \frac{1}{x}\, dx\,| == |\int_{f}^{f_0} \frac{1}{x}\, dx\,|$$

Dérivation

$t(k) = k^2$ ou $t(k) = k^{-2}$

$t(k) = k$ ou $t(k) = k^{-1}$

$t(k) = |\, ln(k) \,|$ ou $t(k) = |\, ln(k^{-1}) \,|$

$t(k) = |\, k * ln(k) - k \,|$ ou $t(k) = |\, k^{-1} * ln(k^{-1}) - k^{-1} \,|$

Intégration

Et comme $t(k) = k$, par récursivité, et à un scalaire et une constante près, nous en déduisons l'espace des fonctions reliant de façon homogène les fréquences aux durées.

3. ANNEXES

3.1 Synthèse par modulation d’amplitude hybride

On propose une synthèse s’appuyant sur le modèle de la modulation d’amplitude dans laquelle on produira des partiels symétriques par rapport à la porteuse (modélisée, comme par la suite, dans le langage sclang de SuperCollider).

```
SynthDef( ring1, { arg out, freq, amp, freq2, amp2, env;
    var myEnv, sigCar, sigMod, ringSig, mySig;
    myEnv = EnvGen.kr(env, doneAction: 2);
    sigCar = SinOsc.ar(freq,0,amp);
    sigMod = SinOsc.ar(freq2,0,amp2);
    ringSig = sigCar * sigMod;
    mySig = Mix.new( [ sigCar, ringSig ] );
    mySig = mySig * myEnv;
    Out.ar(out, mySig);
});

SynthDef( ring1_inverted, { arg out, freq, freq2, amp, amp2, env;
    var myEnv, sigCar, sigMod, ringSig, mySig;
    var partial0, partial1, invertedPartial0, invertedPartial1;
    myEnv = EnvGen.kr(env, doneAction: 2);
    //inversion
    partial0 = ( freq2 - freq ).abs;
    partial1 =  freq + freq2 ;
    invertedPartial0 = freq * freq / partial0;
    invertedPartial1 = freq * freq / partial1;
    //signals
    sigCar = SinOsc.ar(freq, 0, amp);
     ringInvSig = SinOsc.ar( (invertedPartial0 + invertedPartial1) /
2 , 0, amp) * SinOsc.ar( ( invertedPartial1 - invertedPartial0 ).abs
/ 2, 0, amp2);
    //mixing
    mySig = Mix( [ sigCar, ringInvSig ] );
    mySig = mySig * myEnv;
    Out.ar(out, mySig)
});
```

3.2 Traitement FFT

On procède à l’inversion d’un spectre FFT, par rapport au *bin* d’une fréquence centrale, en redistribuant les *bins.* On note la dépendance directe des résultats à la fréquence d’échantillonnage. Détail pour un block de calcul:

```
var size;
var mag,magOut;
var kC,kT;

//FFT size
size = 16 << 1;

//Array In of amplitudes for each bin
mag = Array.fill(size,{1.0.rand});

//Array Out
magOut = Array.newClear(size);
```

```
//FOND = SAMPLERATE / WINDOWSIZE

//***Point of Inversion***
kC = 6;

//***Point of Transposition***
kT = 6;

size.do { arg bin;

   var binInv;
   var k = bin + 1, ki;

   /***Transfert***/
   ki = kC * kT / k;
   ki = ki.abs;
   ki = ( size / ki ).round.max(1);
   ki = ( size / ki ).round;

   binInv = ki - 1;
   magOut[bin] = mag[binInv]

};
```

post:

```
numFrames : 32

In :

[ 0.011272072792053, 0.099066019058228, 0.74963998794556, 0.83413374423981,
0.24126005172729, 0.48250138759613, 0.86760604381561, 0.56131553649902,
0.02237868309021, 0.69347631931305, 0.46185421943665, 0.66997182369232,
0.98952543735504, 0.5582799911499, 0.98287582397461, 0.27776002883911,
0.39898645877838, 0.79964792728424, 0.5860538482666, 0.74896252155304,
0.73582911491394, 0.17250633239746, 0.088589191436768, 0.67151844501495,
0.038662075996399, 0.36029529571533, 0.93727147579193, 0.66694629192352,
0.1666202545166, 0.76466071605682, 0.0082577466964722, 0.33068788051605 ]

         Point d'Inversion : 6 - Point de Transposition : 6

détail:

                  (0 -> 31)               (16 -> 1)
                  (1 -> 15)               (17 -> 1)
                  (2 -> 10)               (18 -> 1)
                  (3 -> 7)                (19 -> 1)
                  (4 -> 7)                (20 -> 1)
                  (5 -> 5)                (21 -> 1)
                  (6 -> 4)                (22 -> 1)
                  (7 -> 4)                (23 -> 1)
                  (8 -> 3)                (24 -> 0)
                  (9 -> 3)                (25 -> 0)
                  (10 -> 2)               (26 -> 0)
                  (11 -> 2)               (27 -> 0)
                  (12 -> 2)               (28 -> 0)
                  (13 -> 2)               (29 -> 0)
                  (14 -> 1)               (30 -> 0)
                  (15 -> 1)               (31 -> 0)
```

```
Out :

[ 0.33068788051605, 0.27776002883911, 0.46185421943665, 0.56131553649902,
0.56131553649902, 0.48250138759613, 0.24126005172729, 0.24126005172729,
0.83413374423981, 0.83413374423981, 0.74963998794556, 0.74963998794556,
0.74963998794556, 0.74963998794556, 0.099066019058228, 0.099066019058228,
0.099066019058228, 0.099066019058228, 0.099066019058228, 0.099066019058228,
0.099066019058228, 0.099066019058228, 0.099066019058228, 0.099066019058228,
0.011272072792053, 0.011272072792053, 0.011272072792053, 0.011272072792053,
0.011272072792053, 0.011272072792053, 0.011272072792053, 0.011272072792053 ]
```

3.3 Écriture d'évènements, structuration

Dans le cadre d'une programmation orientée objet, on propose une série de classes, avec leurs méthodes, prenant en charge l'écriture de séquences (pouvant inclure, ou non, des principes énoncés plus haut).

```
MyPieces {
      *new { }
      normalizeData { }
      getInstrument { }
      getKeysForInstruments { }
      generic { }
      instrumentAndScore { }
}

PieceOfMusic : MyPieces {
      var initValues;
      *new { arg seq,durseq,cut,instrument; }
      init {
            /*===================*/
            /*      data        */
            /*===================*/
            initValues = (seq: seq);
      }
      gen { arg seq,durseq,cut,instrument;
            var outdata,transferVoixFunc;
            var normseq;
            seq = seq ? initValues[\seq];
            transferVoixFunc = { arg seq;
                  /*================*/
                  /* data transfert */
                  /*================*/

            };
            outdata = transferVoixFunc.(seq);
      }
      synthDefs { ^anArrayOfSynthDefs }
}
```

3.3 Interface de programmation pour la manipulation de données

Dans le cadre d'une programmation orientée objet, également, on propose une série de classes, avec leurs méthodes, prenant en charge la manipulation de blocs de données de séquences. *Toute structure est un tableau de séquences parallèles.*

```
SequenceOfData {
   *new { arg pairs; }
   normalize {
         this.normalizeInstrument;
         this.everyValueIsArray;
         this.wrapExtendArray;
         this.normalizeDuration;
         this.timeLine;
   }
   normalizeInstrument { }
   everyValueIsArray { }
   numOfEvents { }
   wrapExtendArray { }
   normalizeDuration { }
   wrap_dur_extend { arg durseq ; }
   timeLine { }
   dur_cut_correction { }
   padding { }
   getPairs { }
   at { arg key; }
   getInstrument { }
   replaceInstrument { }
   makePattern { }
}

SequenceParalleleOfData {
   *new { arg array ; }
   normalize { arg durseq, cut ; }
   getPairs { }
   data { }
   durseq { }
   getInstrument { }
   numOfVoices { }
   replaceInstrument { }
   makePattern { }
}
```

Exemples :

```
Synthétiseur "instr0" (freq,amp,freq1,freqC,amp2,xc,youpi) { /*...code...*/ };
Synthétiseur "instr2" (freq1,freqC,amp,youpi) { /*...code...*/ };
Synthétiseur "instr3" (freq1,freqC,amp2,xc) { /*...code...*/ };
Synthétiseur "instr4" (freq1,freq2,amp2,xc) { /*...code...*/ };

//section composée de sequences d'évenements
[
   [instrument:"instr0", freq: [67,890], amp:[0.2], youpi:6.7, dur:6],
   [instrument:"instr2", freq1: [67,890], freqC:[90,889], amp:[0.2], youpi:
6.7, dur:6],
   [instrument:"instr3", freq1: 890, freqC:[90,889], amp2:[0.2], xc:0.9 ,
delta: [6,9] ],
   [instrument:"instr0", freq1: 890, freqC:90, amp2:0.2, xc:0.9 , delta: 6 ],
```

```
   [instrument:"instr4", freq1: 890, freq2:[90,889], amp2:[0.2], xc:0.9 ,
delta: [9,8,2] ]
];

//dictionnaire de la section
=> [
   'instrument', [ 'instr0', 'instr2', 'instr3', 'instr3', 'instr0', 'instr4',
'instr4', 'instr4' ],
   'freq', [ 67, nil, nil, nil, nil, nil, nil, nil ],
   'freqC', [ nil, 90, 90, 889, 90, nil, nil, nil ],
   'freq1', [ nil, 67, 890, 890, 890, 890, 890, 890 ],
   'freq2', [ nil, nil, nil, nil, nil, 90, 889, 90 ],
   'xc', [ nil, nil, 0.9, 0.9, 0.9, 0.9, 0.9, 0.9 ],
   'amp2', [ nil, nil, 0.2, 0.2, 0.2, 0.2, 0.2, 0.2 ],
   'amp', [ 0.2, 0.2, nil, nil, nil, nil, nil, nil ],
   'youpi', [ 6.7, 6.7, nil, nil, nil, nil, nil, nil ],
   'dur', [ 6, 6, 6, 9, 6, 9, 8, 2 ],
   'delta', [ 6, 6, 6, 9, 6, 9, 8, 2 ]
];

//dictionnaire corrigé pour certaines valeurs
=> [
   'instrument', [ 'instr0', 'instr2', 'instr3', 'instr3', 'instr0', 'instr4',
'instr4', 'instr4' ],
   'freq', [ 67, 0, 0, 0, 0, 0, 0, 0 ],
   'freqC', [ 0, 90, 90, 889, 90, 0, 0, 0 ],
   'freq1', [ 0, 67, 890, 890, 890, 890, 890, 890 ],
   'freq2', [ 0, 0, 0, 0, 0, 90, 889, 90 ],
   'xc', [ nil, nil, 0.9, 0.9, 0.9, 0.9, 0.9, 0.9 ],
   'amp2', [ 0, 0, 0.2, 0.2, 0.2, 0.2, 0.2, 0.2 ],
   'amp', [ 0.2, 0.2, 0, 0, 0, 0, 0, 0 ],
   'youpi', [ 6.7, 6.7, nil, nil, nil, nil, nil, nil ],
   'dur', [ 6, 6, 6, 9, 6, 9, 8, 2 ],
   'delta', [ 6, 6, 6, 9, 6, 9, 8, 2 ]
];

//extrait de la séquence, de l'évènement : à la 6ème seconde, pendant une
durée de 7 secondes.
extrait (at=6, dur=7) { /*...*/ };
=> [ 'youpi', [ 6.7, 6.7, nil ], 'freq2', [ 0, 0, 0 ], 'instrument',
[ 'instr0', 'instr2', 'instr3' ], 'xc', [ nil, nil, 0.9 ], 'delta', [ 6, 6,
6 ], 'freqC', [ 0, 90, 90 ], 'index', [ 0, 1, 2 ], 'dur', [ 6, 6, 6 ], 'freq',
[ 67, 0, 0 ], 'amp2', [ 0, 0, 0.2 ], 'amp', [ 0.2, 0.2, 0 ], 'freq1', [ 0, 67,
890 ] ];

//modification
[ \xc: [7,89,90] ];
=> [ 'youpi', [ 6.7, 6.7, nil ], 'freq2', [ 0, 0, 0 ], 'instrument',
[ 'instr0', 'instr2', 'instr3' ], 'xc', [ 7, 89, 90 ], 'delta', [ 6, 6, 6 ],
'freqC', [ 0, 90, 90 ], 'index', [ 0, 1, 2 ], 'dur', [ 6, 6, 6 ], 'freq',
[ 67, 0, 0 ], 'amp2', [ 0, 0, 0.2 ], 'amp', [ 0.2, 0.2, 0 ], 'freq1', [ 0, 67,
890 ] ]

//reinsertion
=> [ 'xc', [ 7, 89, 90, 0.9, 0.9, 0.9, 0.9, 0.9 ], 'instrument', [ 'instr0',
'instr2', 'instr3', 'instr3', 'instr0', 'instr4', 'instr4', 'instr4' ], 'dur',
[ 6, 6, 6, 9, 6, 9, 8, 2 ], 'freqC', [ 0, 90, 90, 889, 90, 0, 0, 0 ], 'freq',
[ 67, 0, 0, 0, 0, 0, 0, 0 ], 'freq1', [ 0, 67, 890, 890, 890, 890, 890, 890 ],
'delta', [ 6, 6, 6, 9, 6, 9, 8, 2 ], 'amp2', [ 0, 0, 0.2, 0.2, 0.2, 0.2, 0.2,
0.2 ], 'freq2', [ 0, 0, 0, 0, 0, 90, 889, 90 ], 'amp', [ 0.2, 0.2, 0, 0, 0, 0,
0, 0 ], 'youpi', [ 6.7, 6.7, nil, nil, nil, nil, nil, nil ] ]
```

Conclusion

On a décrit un ensemble d'éléments à même de rendre compte des possibilités de contrôle et d'écriture d'un signal, dans la continuité et le souci d'un certain nombre de lois sonores et musicales, dans le but d'envisager son écriture, son hybridation, sa synthèse. On peut, à cet égard, aller entendre les travaux de l'auteur de cet article.

Références

- digitaltonal.com / digitaltonal.com/albums
- « Stria » (1977) de John Chowning à https://waaw.csound.com/stria.html — (synthèse FM et utilisation du nombre d'or ; analyse de Laura Zattra à http://brahms.ircam.fr/analyses/Stria/)
- Plugin de traitement FFT, comme décrit dans la section 7, pour SuperCollider, à https://digitaltonal.com/download-pv_inverse-plugin-for-supercollider. Code source en C++: http://digitaltonal.com/dev/inverse/pv_inverse-cpp.txt. Utilisé dans le morceau *Idiom64*: http://digitaltonal.com/album/digitaltonal-album/1_idiom64.mp3

Bibliographie

- P. Boulez, *Penser la musique aujourd'hui*, 1963
- H. Helmoltz, *Théorie physiologique de la musique, basée sur l'étude des sensations auditives*, 1868
- M. Puckette, *The theory and technique of electronic music*, 2007