

音響和声法

立体の思考

前章で定義した複数の音程と音色を内包する音響和音という(音)面の時間的な連鎖としての音響和声は、垂直と奥行からなる面にさらにもうひとつの次元(水平)が加わったひとつの「(音)立体」と捉えることができる。それは時間軸上で変化する音響和音を、ある量塊を持ったヴォリュームとして立体的に捉えることである。同時にそれは、音響を構成するひとつひとつの音をパラレルに聴き、それをコントロールするために必要な感覚的なバランスを習得するための新たなメソッドの提案でもある。立体的な音響テクスチャーとしての全体の響きと、マイクロな部分(内部)としての線の動きの双方を、同時に聴きとる「立体の思考」による音響和音の連鎖のデザイン-それが音響和声法である。

音響和声(音響和音の連鎖)を記述するために、周波数マトリックスと振幅マトリックスの初期値と終値を設定しその間を線形補間する関数を定義する。

```
// 8x8の周波数、振幅、パンマトリックスと基音の初期値と終値を設定し線形補間する関数
(
~out = {arg dur = 10, base1 = 441, base2 = 441, level = 0.8;
  var farray1, farray2, aarray1, aarray2, parray1, parray2;
  farray1 = Control.names(\farray1).kr(Array.fill(64, 0));
  farray2 = Control.names(\farray2).kr(Array.fill(64, 0));
  aarray1 = Control.names(\aarray1).kr(Array.fill(64, 0));
  aarray2 = Control.names(\aarray2).kr(Array.fill(64, 0));
  parray1 = Control.names(\parray1).kr(Array.fill(64, 0));
  parray2 = Control.names(\parray2).kr(Array.fill(64, 0));
  Mix.new(
    Pan2.ar(
      SinOsc.ar(
        Line.kr(farray1 * base1, farray2 * base2, dur),
        0.0,
        Line.kr(aarray1, aarray2, dur),
        0.0),
      Line.kr(parray1, parray2, dur), level));
)
```

```
// フェイドアウト時間を0秒に設定(フェイドアウトしない)
p.fadeTime = 0;
```

音響と調性

調や調性とは、ある主音と他の音との関係を構造化するための枠組みである。古典的な旋律やリズムは相対的(音高よりも音程が重要)であるが、調性は絶対的である。(主)音の絶対的な高さを変化させることで、移調と呼ばれる調の移動が起こる。

調性にとって最も重要なのは主音の音高(周波数)である。主音を核として、他の音をその主音に自然と引き寄せられるように関係づけることが調性である。逆にいえば、聴き手がある音を聴くことで、そこに中心(主音)となる最も重要な音高(周波数)があると感じれば、そこに調性が生まれる。調性が「音の重力」と呼ばれる所以である。

音響和声法において、調性は主音の周波数の絶対値で表現される。

古典的ドミナント・モーション

古典的な和声法において最も重要な和音進行が、ドミナント(V7)→トニック(I)の動きである。このドミナント・モーションをコードで記述する。

```
// ゆっくりと遷移する古典的なドミナント・モーション(主音=441[Hz]、ランダムパン)
```

```

(
~out.setn(
  \dur, 10,
  \base1, 441,
  \base2, 441,
  \farray1, (Array.series(16, 1, 1) * .x [15/16, 9/8, 4/3, 3/2]).postln,
  \farray2, (Array.series(16, 1, 1) * .x [15/16, 1, 5/4, 3/2]).postln,
  \aarray1, (Array.series(16, 1, 1).reciprocal * .x Array.fill(4, 1)).normalizeSum.postln,
  \aarray2, (Array.series(16, 1, 1).reciprocal * .x Array.fill(4, 1)).normalizeSum.postln,
  \parray1, Array.fill(64, {1.0.rand2}).postln,
  \parray2, Array.fill(64, {1.0.rand2}).postln).rebuild;
)

```

// ドミナント(V7)の代理コード(II ♭7)を用いる(主音=441[Hz]、ランダムパン)

```

(
~out.setn(
  \dur, 10,
  \base1, 441,
  \base2, 441,
  \farray1, (Array.series(16, 1, 1) * .x [15/16, 16/15, 4/3, 8/5]).postln,
  \farray2, (Array.series(16, 1, 1) * .x [15/16, 1, 5/4, 3/2]).postln,
  \aarray1, (Array.series(16, 1, 1).reciprocal * .x Array.fill(4, 1)).normalizeSum.postln,
  \aarray2, (Array.series(16, 1, 1).reciprocal * .x Array.fill(4, 1)).normalizeSum.postln,
  \parray1, Array.fill(64, {1.0.rand2}).postln,
  \parray2, Array.fill(64, {1.0.rand2}).postln).rebuild;
)

```

音響ドミナント・モーション

ドミナント・モーションにおいて、導音(リーディング・ノート)から主音へと短2度上行(解決)する音の動き、および同様に短2度下行する音の動き(アプローチ・ノート/クロマティック・アプローチ)が重要である。

この2つの可逆な音の動きにより音響和音を連結する(音響ドミナント・モーション)。

// 5度構成の音響和音に対するランダムなドミナント・モーション

```

(
var tonic = Array.series(8, 1, 1) * .x Array.geom(8, 1, 3/2);
var dominant = tonic * Array.fill(64, {[15/16, 16/15].choose});
tonic.postln;
dominant.postln;
~out.setn(
  \dur, 10,
  \base1, 126,
  \base2, 126,
  \farray1, dominant,
  \farray2, tonic,
  \aarray1, (Array.series(8, 1, 1).reciprocal * .x Array.fill(8, 1)).normalizeSum.postln,
  \aarray2, (Array.series(8, 1, 1).reciprocal * .x Array.fill(8, 1)).normalizeSum.postln,
  \parray1, Array.fill(64, {1.0.rand2}).postln,
  \parray2, Array.fill(64, {1.0.rand2}).postln).rebuild;
)

```

// ランダム構成の音響和音に対するドミナント・モーション(和声の動きの抽象)

```

(
var tonic = Array.exprand(64, 1.0, 100.0);
var dominant = tonic * Array.fill(64, {[15/16, 16/15].choose});
tonic.postln;
dominant.postln;
~out.setn(
  \dur, 3,
  \base1, 100,
  \base2, 100,

```

```

    \farray1, dominant,
    \farray2, tonic,
    \aarray1, Array.rand(64, 0.0, 1.0).normalizeSum.postln,
    \aarray2, Array.rand(64, 0.0, 1.0).normalizeSum.postln,
    \parray1, Array.fill(64, {1.0.rand2}).postln,
    \parray2, Array.fill(64, {1.0.rand2}).postln).rebuild;
)

// より密集した音響和音を用いる
(
var tonic = Array.exprand(64, 1.0, 8.0);
var dominant = tonic * Array.fill(64, {[15/16, 16/15].choose});
tonic.postln;
dominant.postln;
~out.setn(
  \dur, 3,
  \base1, 441,
  \base2, 441,
  \farray1, dominant,
  \farray2, tonic,
  \aarray1, Array.rand(64, 0.0, 1.0).normalizeSum.postln,
  \aarray2, Array.rand(64, 0.0, 1.0).normalizeSum.postln,
  \parray1, Array.fill(64, {1.0.rand2}).postln,
  \parray2, Array.fill(64, {1.0.rand2}).postln).rebuild;
)

```

▼エクササイズ：音響ドミナント・モーションの連鎖による音響和声

```

// TaskProxyの定義
x = TaskProxy.basicNew;
x.play;

// 音響ドミナント・モーションによる音響和声の連鎖
(
var dur = 3;
var dt = 10;
var base1, base2 = 441;
var farray1, farray2 = Array.exprand(64, 1.0, 8.0);
var aarray1, aarray2 = Array.rand(64, 0.0, 1.0);
var parray1, parray2 = Array.fill(64, {1.0.rand2});
var ratio = [15/16, 16/15];
base1 = base2;
x.source = {
  loop{
    farray1 = farray2;
    aarray1 = aarray2;
    parray1 = parray2;
    farray2 = farray1 * Array.fill(64, {ratio.choose});
    aarray2 = Array.rand(64, 0.0, 1.0);
    parray2 = Array.fill(64, {1.0.rand2});
    ~out.setn(
      \dur, dur,
      \base1, base1,
      \base2, base2,
      \farray1, farray1.postln,
      \farray2, farray2.postln,
      \aarray1, aarray1.normalizeSum,
      \aarray2, aarray2.normalizeSum,
      \parray1, parray1,
      \parray2, parray2 ).rebuild;
    dt.wait;
  }
}
)

```

```
// 和声の進行を止める
x.stop;
```

異なる初期値からの和声の進行を聴きくらべてみよ。何か共通の感覚が得られるか?

- ・より疎な和音 : `Array.exprand(64, 1.0, 100.0)`
 - ・1本の線 : `Array.fill(64, 1.0)`
 - ・5度構成の和音 : `Array.series(8, 1, 1) * .x Array.geom(8, 1, 3/2)`
- などのさまざまな初期値を試してみよ。

さまざまな和声

さまざまな音程列と倍音列の組み合わせによる和音の連鎖を聴いてみる。遷移時間と基音と遷移時間の変化にも着目する。

```
// 振幅反比例の完全5度と短2度の音程列から、長3度と短3度の音程列へ(基音200[Hz]、ランダムパン)
(
~out.setn(
  \dur, rrand(0.1, 1.0).postln, // 和音の遷移時間は1~2秒
  \base1, 200,
  \base2, 200,
  \farray1, (Array.geom(8, 1, 1.5) * .x Array.geom(8, 1, 1.067)).postln,
  \farray2, (Array.geom(8, 1, 1.25) * .x Array.geom(8, 1, 1.2)).postln,
  \aarray1, (Array.series(8, 1, 1) * .x Array.series(8, 1, 1)).reciprocal.normalizeSum.postln,
  \aarray2, (Array.series(8, 1, 1) * .x Array.series(8, 1, 1)).reciprocal.normalizeSum.postln,
  \parray1, Array.fill(64, {1.0.rand2}).postln,
  \parray2, Array.fill(64, {1.0.rand2}).postln).rebuild;
)
~out.play
```

```
// 振幅反比例の完全5度と短2度の音程列から、長3度と短3度の音程列へ(基音200[Hz]、ランダムパン)
(
~out.setn(
  \dur, rrand(0.1, 1.0).postln, // 和音の遷移時間は1~2秒
  \base1, 200,
  \base2, 200,
  \farray1, (Array.geom(8, 1, 1.5) * .x Array.geom(8, 1, 1.067)).postln,
  \farray2, (Array.geom(8, 1, 1.25) * .x Array.geom(8, 1, 1.2)).postln,
  \aarray1, (Array.series(8, 1, 1) * .x Array.series(8, 1, 1)).reciprocal.normalizeSum.postln,
  \aarray2, (Array.series(8, 1, 1) * .x Array.series(8, 1, 1)).reciprocal.normalizeSum.postln,
  \parray1, Array.fill(64, {1.0.rand2}).postln,
  \parray2, Array.fill(64, {1.0.rand2}).postln).rebuild;
)
~out.play
```

```
// ランダム振幅の倍音列と増4度の音程列から振幅反比例の倍音列と振幅一定の完全5度の音程列へ(基音50[Hz]、ランダムパン)
(
~out.setn(
  \dur, exprand(0.1, 10.0).postln, // 和音の遷移時間は0~10秒
  \base1, 50,
  \base2, 50,
  \farray1, (Array.series(8, 1, 1) * .x Array.geom(8, 1, 2.sqrt)).postln,
  \farray2, (Array.series(8, 1, 1) * .x Array.geom(8, 1, 1.5)).postln,
  \aarray1, (Array.rand(8, 0.0, 1.0) * .x Array.rand(8, 0.0, 1.0)).normalizeSum.postln,
  \aarray2, (Array.series(8, 1, 1) * .x Array.series(8, 1, 1)).reciprocal.normalizeSum.postln,
  \parray1, Array.fill(36, {1.0.rand2}).postln,
  \parray2, Array.fill(36, {1.0.rand2}).postln).rebuild;
)

```

```

// ランダム振幅の倍音列と長3度から完全7度へと変化する音程列(基音100[Hz]→50[Hz]、ランダムパン)
(
~out.setn(
  \dur, 10,
  \base1, 100,
  \base2, 50,
  \farray1, (Array.series(8, 1, 1) * .x Array.geom(8, 1, 1.25)).postln,
  \farray2, (Array.series(8, 1, 1) * .x Array.geom(8, 1, 1.75)).postln,
  \aarray1, Array.rand(64, 0.0, 1.0).normalizeSum.postln,
  \aarray2, Array.rand(64, 0.0, 1.0).normalizeSum.postln,
  \parray1, Array.fill(64, {1.0.rand2}).postln,
  \parray2, Array.fill(64, {1.0.rand2}).postln).rebuild;
)

// ランダム振幅の倍音列と増4度から完全5度へと変化する音程列(基音100[Hz]→80[Hz]、ランダムパン)
(
~out.setn(
  \dur, 10,
  \base1, 150,
  \base2, 100,
  \farray1, (Array.series(8, 1, 1) * .x Array.geom(8, 1, 4/3)).postln,
  \farray2, (Array.series(8, 1, 1) * .x Array.geom(8, 1, 4/3)).postln,
  \aarray1, (Array.rand(64, 0.0, 1.0)).normalizeSum.postln,
  \aarray2, (Array.rand(64, 0.0, 1.0)).normalizeSum.postln,
  \parray1, Array.fill(64, {1.0.rand2}).postln,
  \parray2, Array.fill(64, {1.0.rand2}).postln).rebuild;
)

// ランダム振幅の倍音列と増4度の音程列から振幅反比例の倍音列と振幅一定の完全5度の音程列へ(基音441[Hz]→400[Hz]、ランダムパン)
(
~out.setn(
  \dur, 10.0.rand.postln, // 和音の遷移時間は0~10秒
  \base1, 441,
  \base2, 400,
  \farray1, (Array.geom(8, 1, 2**0.2) * .x Array.geom(8, 1, 2**0.25)).postln,
  \farray2, (Array.geom(8, 1, 2**0.2) * .x Array.geom(8, 1, 2**0.55)).postln,
  \aarray1, (Array.series(8, 1, 1) * .x Array.series(8, 1, 1)).reciprocal.normalizeSum.postln,
  \aarray2, (Array.series(8, 1, 1) * .x Array.series(8, 1, 1)).reciprocal.normalizeSum.postln,
  \parray1, Array.fill(36, {1.0.rand2}).postln,
  \parray2, Array.fill(36, {1.0.rand2}).postln).rebuild;
)

```

緊張と弛緩(解決)の連鎖

ランダムな音程列(緊張)と協和音程列(弛緩)のくりかえしによる音響和声を聴く

```

// TaskProxyの定義
x = TaskProxy.basicNew;
x.play;

p.fadeTime = 10;

//
(
var dur = 10;
var dt = 10;
var base1, base2 = 50;
var farray1, farray2 = Array.series(8, 1, 1) * .x Array.geom(8, 1, 1.5);
var aarray1, aarray2 = Array.series(8, 1, 1).reciprocal * .x Array.fill(8, 1);
var parray1, parray2 = Array.fill(64, {1.0.rand2});
var ratio = [6/5, 5/4, 4/3, 3/2];
base1 = base2;
x.source = {
  loop{
    farray1 = farray2;
    aarray1 = aarray2;

```

```

parray1 = parray2;
farray2 = farray2 * Array.fill(64, {rrand(4/5, 5/4)});
aarray2 = Array.rand(64, 0.0, 1.0);
parray2 = Array.fill(64, {1.0.rand2});
~out.setn(
  \dur, dur,
  \base1, base1,
  \base2, base2,
  \farray1, farray1.postln,
  \farray2, farray2.postln,
  \aarray1, aarray1.normalizeSum,
  \aarray2, aarray2.normalizeSum,
  \parray1, parray1,
  \parray2, parray2 ).rebuild;
dt.wait;
farray1 = farray2;
aarray1 = aarray2;
parray1 = parray2;
farray2 = Array.series(8, 1, 1) * .x Array.geom(8, 1, 1.5);
aarray2 = Array.series(8, 1, 1).reciprocal * .x Array.fill(8, 1);
parray2 = Array.fill(64, {1.0.rand2});
~out.setn(
  \dur, dur,
  \base1, base1,
  \base2, base2,
  \farray1, farray1.postln,
  \farray2, farray2.postln,
  \aarray1, aarray1.normalizeSum,
  \aarray2, aarray2.normalizeSum,
  \parray1, parray1,
  \parray2, parray2 ).rebuild;
dt.wait;
}
}
)

// 和声の進行を止める
x.stop;

```

和声の視覚化

音響和声の全体構造を把握するために、和音を構成する64本の線の遷移を視覚的に表現する。周波数が音の高さ、振幅が線の太さに対応している。

```

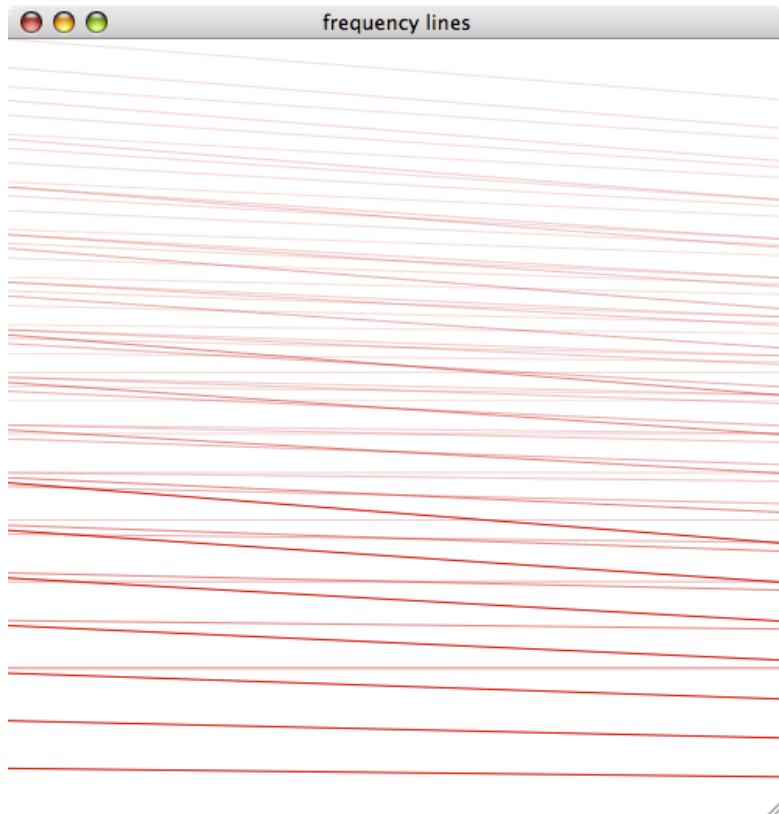
// 振幅反比例の倍音列と振幅一定の長3度の音程列から振幅一定の短3度の音程列へ(基音200[Hz]、ランダムパン)
(
var wmax = 500;
var fgw = SCWindow("frequency lines", Rect(50, 300, wmax, wmax));
var dur = 10;
var base1 = 200;
var base2 = 200;
var farray1 = Array.series(8, 1, 1) * .x Array.geom(8, 1, 5/4);
var farray2 = Array.series(8, 1, 1) * .x Array.geom(8, 1, 6/5);
var aarray1 = Array.series(8, 1, 1).reciprocal * .x Array.fill(8, 1);
var aarray2 = Array.series(8, 1, 1).reciprocal * .x Array.fill(8, 1);
var parray1 = Array.fill(64, {1.0.rand2});
var parray2 = Array.fill(64, {1.0.rand2});
var farray = farray1 ++ farray2;
var fmax = farray.sort.last.log10;
fgw.view.background = Color.gray(1.0);
fgw.alpha = 1.0;
fgw.front;
fgw.drawHook = {
  Pen.use {

```

```

    Color.red.set;
    64.do { arg i;
      Pen.width = 0.5 * (aarray1.at(i) + aarray2.at(i));
      Pen.beginPath;
      Pen.moveTo(Point(0, wmax * (1 - (farray1.at(i).log10/fmax))));
      Pen.lineTo(Point(wmax, wmax * (1 - (farray2.at(i).log10/fmax))));
      Pen.stroke;
    };
  };
};
fgw.refresh;
~out.setn(
  \dur, dur,
  \base1, base1,
  \base2, base2,
  \farray1, farray1,
  \farray2, farray2,
  \aarray1, aarray1.normalizeSum,
  \aarray2, aarray2.normalizeSum,
  \parray1, parray1,
  \parray2, parray2 ).rebuild;
)

```



```

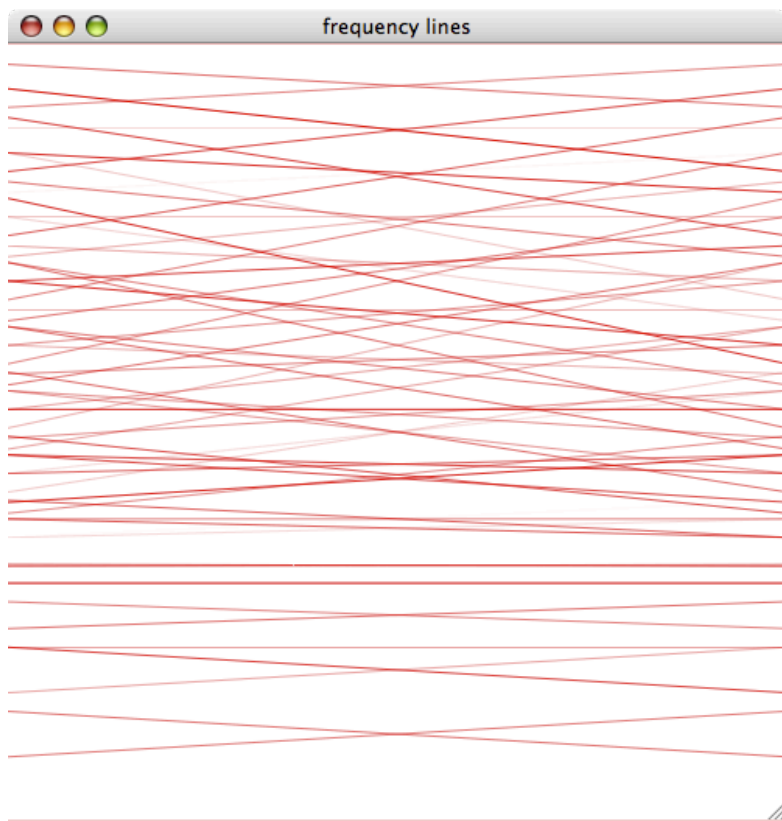
// ランダム振幅の倍音列と完全5度の音程列の行と列の入れかえ(基音=50[Hz]、ランダムパン)
(
var wmax = 500;
var fgw = SCWindow("frequency lines", Rect(50, 300, wmax, wmax));
var dur = 10;
var base1 = 50;
var base2 = 50;
var farray1 = Array.series(8, 1, 1) * .x Array.geom(8, 1, 1.5);
var farray2 = Array.geom(8, 1, 1.5) * .x Array.series(8, 1, 1);

```

```

var aarray1 = Array.fill(64, {1.0.rand});
var aarray2 = Array.fill(64, {1.0.rand});
var parray1 = Array.fill(64, {1.0.rand2});
var parray2 = Array.fill(64, {1.0.rand2});
var farray = farray1 ++ farray2;
var fmax = farray.sort.last.log10;
fgw.view.background = Color.gray(1.0);
fgw.alpha = 1.0;
fgw.front;
fgw.drawHook = {
  Pen.use {
    Color.red.set;
    64.do { arg i;
      Pen.width = 0.5 * (aarray1.at(i) + aarray2.at(i));
      Pen.beginPath;
      Pen.moveTo(Point(0, wmax * (1 - (farray1.at(i).log10/fmax))));
      Pen.lineTo(Point(wmax, wmax * (1 - (farray2.at(i).log10/fmax))));
      Pen.stroke;
    };
  };
};
fgw.refresh;
~out.setn(
  \dur, dur,
  \base1, base1,
  \base2, base2,
  \farray1, farray1,
  \farray2, farray2,
  \aarray1, aarray1.normalizeSum,
  \aarray2, aarray2.normalizeSum,
  \parray1, parray1,
  \parray2, parray2 ).rebuild;
)

```



遷移の中間点で、一瞬和音が澄んで聴こえるところがある(線の交点)

保持音

和声の進行とは無関係に持続し続ける音のことを、保持音(オルゲンブнкт/オルガンポイント)と呼ぶ。保持音には線以外にも、断続する線、点、面など様々な音を用いることができる。

```
// 8820[Hz]の保持音を鳴らす
~pedal = { SinOsc.ar(8820.dup, 0, 0.1, 0) };
~pedal.play

x = TaskProxy.basicNew;
x.play;

// この保持音にもう1本別の線(441[Hz])を初期値とする音響ドミナント・モーションを加える
(
var dur = 3;
var dt = 10;
var base1, base2 = 441;
var farray1, farray2 = Array.fill(64, 1.0);
var aarray1, aarray2 = Array.rand(64, 0.0, 1.0);
var parray1, parray2 = Array.fill(64, {1.0.rand2});
var ratio = [15/16, 16/15];
base1 = base2;
x.source = {
  loop{
    farray1 = farray2;
    aarray1 = aarray2;
    parray1 = parray2;
    farray2 = farray1 * Array.fill(64, {ratio.choose});
    aarray2 = Array.rand(64, 0.0, 1.0);
    parray2 = Array.fill(64, {1.0.rand2});
    ~out.setn(
      \level, 0.4,
      \dur, dur,
      \base1, base1,
      \base2, base2,
      \farray1, farray1.postln,
      \farray2, farray2.postln,
      \aarray1, aarray1.normalizeSum,
      \aarray2, aarray2.normalizeSum,
      \parray1, parray1,
      \parray2, parray2 ).rebuild;
    dt.wait;
  }
}
)

// 保持音の高さを88.2[Hz]にする
~pedal = { SinOsc.ar(88.2.dup, 0, 0.5, 0) };

// 保持音の高さを441[Hz](和声の基音)にする
~pedal = { SinOsc.ar(441.dup, 0, 0.1, 0) };

// 保持音を70[Hz]の線の断続にする
~pedal = { SinOsc.ar(70.dup, 0, LFPulse.kr(10, 0, 0.5, 0.3), 0) };

// 保持音を2[Hz]の点列にする
~pedal = { Impulse.ar(2.dup, 0, 0.5) };
```

```
// 保持音を面にする
~pedal = { WhiteNoise.ar(0.01.dup) };
```

```
// 保持音と和声の進行を止める
~pedal.stop;
x.stop;
```

輪郭と骨組

```
x = TaskProxy.basicNew;
x.play;
```

```
p.fadeTime = 10;
```

```
// 和音の最高音と最低音を保持音とする(和音の輪郭を既定する)
(
var dur = 10;
var dt = 10;
var base1, base2 = 441;
var flow = 1.0, fhigh = 4.0;
var farray1, farray2, aarray1, aarray2, parray1, parray2;
var ratio = [15/16, 1, 16/15];
base1 = base2;
farray2 = [ flow, fhigh ] ++ Array.rand(62, flow, fhigh);
aarray2 = [ 1.0, 1.0 ] ++ Array.rand(62, 0.0, 1.0);
parray2 = Array.fill(64, {1.0.rand2});
x.source = {
  loop{
    flow = flow * ratio.choose; // 輪郭(保持音)も半音進行させる
    fhigh = fhigh * ratio.choose;
    farray1 = farray2;
    aarray1 = aarray2;
    parray1 = parray2;
    farray2 = [ flow, fhigh ] ++ Array.rand(62, flow, fhigh);
    aarray2 = [ 1.0, 1.0 ] ++ Array.rand(62, 0.0, 1.0);
    parray2 = Array.fill(64, {1.0.rand2});
    ~out.setn(
      \level, 0.4,
      \dur, dur,
      \base1, base1,
      \base2, base2,
      \farray1, farray1.postln,
      \farray2, farray2.postln,
      \aarray1, aarray1.normalizeSum,
      \aarray2, aarray2.normalizeSum,
      \parray1, parray1,
      \parray2, parray2 ).rebuild;
    dt.wait;
  }
}
)
```

```
// 初期値としての3和音(長3和音)を保持音とする(和音の骨組を既定する)
(
var dur = 10;
var dt = 10;
var base1, base2 = 441;
var ftriad = [1, 5/4, 3/2];
var fadd = Array.fill(61, {ftriad.choose});
var farray1, farray2, aarray1, aarray2, parray1, parray2;
var ratio = [15/16, 1, 16/15];
base1 = base2;
farray2 = ftriad ++ fadd;
```

```

aarray2 = [ 1.0, 1.0, 1.0 ] ++ Array.rand(61, 0.0, 1.0);
parray2 = Array.fill(64, {1.0.rand2});
x.source = {
  loop{
    fadd = fadd * Array.fill(61, {ratio.choose});
    farray1 = farray2;
    aarray1 = aarray2;
    parray1 = parray2;
    farray2 = ftriad ++ fadd;
    aarray2 = [ 1.0, 1.0, 1.0 ] ++ Array.rand(61, 0.0, 1.0);
    parray2 = Array.fill(64, {1.0.rand2});
    ~out.setn(
      \level, 0.4,
      \dur, dur,
      \base1, base1,
      \base2, base2,
      \farray1, farray1.postln,
      \farray2, farray2.postln,
      \aarray1, aarray1.normalizeSum,
      \aarray2, aarray2.normalizeSum,
      \parray1, parray1,
      \parray2, parray2 ).rebuild;
    dt.wait;
  }
}
)
x.stop;

```

持続と変化

保持音と和声の組み合わせのように、和声法において、持続する(変化しない)部分(ペダル、ドローン)と変化する部分(メロディー)の組み合わせが重要である。和音の連鎖は、経時的なポリ・コードとも捉えられる(逆に、ポリ・コードは同時的な和声とも捉えられる)。

```

// 64本(8x8のマトリクス)の線を2組に分けて、初期値と終値を設定し線形補間する関数を定義する
(
~out = {arg dur = 10, base11 = 441, base12 = 441, base21 = 441, base22 = 441, level = 0.4;
  var farray1, farray2, aarray1, aarray2, parray1, parray2;
  var farray11, farray12, aarray11, aarray12, parray11, parray12;
  var farray21, farray22, aarray21, aarray22, parray21, parray22;
  farray11 = Control.names(\farray11).kr(Array.fill(32, 0));
  farray12 = Control.names(\farray12).kr(Array.fill(32, 0));
  farray21 = Control.names(\farray21).kr(Array.fill(32, 0));
  farray22 = Control.names(\farray22).kr(Array.fill(32, 0));
  aarray11 = Control.names(\aarray11).kr(Array.fill(32, 0));
  aarray12 = Control.names(\aarray12).kr(Array.fill(32, 0));
  aarray21 = Control.names(\aarray21).kr(Array.fill(32, 0));
  aarray22 = Control.names(\aarray22).kr(Array.fill(32, 0));
  parray11 = Control.names(\parray11).kr(Array.fill(32, 0));
  parray12 = Control.names(\parray12).kr(Array.fill(32, 0));
  parray21 = Control.names(\parray21).kr(Array.fill(32, 0));
  parray22 = Control.names(\parray22).kr(Array.fill(32, 0));
  farray1 = (farray11 * base11) ++ (farray21 * base21);
  farray2 = (farray12 * base12) ++ (farray22 * base22);
  aarray1 = aarray11 ++ aarray21;
  aarray2 = aarray12 ++ aarray22;
  parray1 = parray11 ++ parray21;
  parray2 = parray12 ++ parray22;
  Mix.new(
    Pan2.ar(
      SinOsc.ar(

```

```

        Line.kr(farray1, farray2, dur),
        0.0,
        Line.kr(aarray1, aarray2, dur),
        0.0),
    Line.kr(parray1, parray2, dur, level));
)

```

```
p.fadeTime = 3;
```

```
// 持続する和音(保持和音)と変化する和音(音響ドミナントモーション)の組み合わせ
```

```

(
var dur = 3;
var dt = 10;
var base11, base12 = 441;
var base21, base22 = 441;
var farray11, farray12 = Array.exprand(32, 1.0, 8.0);
var aarray11, aarray12 = Array.rand(32, 0.0, 1.0);
var parray11, parray12 = Array.fill(32, {1.0.rand2});
var farray21, farray22 = Array.exprand(32, 1.0, 8.0);
var aarray21, aarray22 = Array.rand(32, 0.0, 1.0);
var parray21, parray22 = Array.fill(32, {1.0.rand2});
var ratio = [15/16, 16/15];
base11 = base12;
base21 = base22;
farray21 = farray22;
aarray21 = aarray22;
parray21 = parray22;
x.source = {
  loop{
    farray11 = farray12;
    aarray11 = aarray12;
    parray11 = parray12;
    farray12 = farray11 * Array.fill(32, {ratio.choose});
    aarray12 = Array.rand(32, 0.0, 1.0);
    parray12 = Array.fill(32, {1.0.rand2});
    ~out.setn(
      \dur, dur,
      \base11, base11,
      \base12, base12,
      \base21, base21,
      \base22, base22,
      \farray11, farray11.postln,
      \farray12, farray12.postln,
      \aarray11, aarray11.normalizeSum,
      \aarray12, aarray12.normalizeSum,
      \parray11, parray11,
      \parray12, parray12,
      \farray21, farray21,
      \farray22, farray22,
      \aarray21, aarray21.normalizeSum,
      \aarray22, aarray22.normalizeSum,
      \parray21, parray21,
      \parray22, parray22 ).rebuild;
    dt.wait;
  }
}
)

```

▼エクササイズ：和声のデザイン

変奏や和音の場合と同様に、数式や数列を使って、特定の志向性を持たないランダム(確率的)な和音から「音響和声のコンポジション」へと発展させていく。

```
// TaskProxyの定義
x = TaskProxy.basicNew;
x.play;

// ランダムな初期値からクロマティックに変化していく2つの和声
(
var dur = 3;
var dt = 3;
var base11, base12 = 441;
var base21, base22 = 441;
var farray11, farray12 = Array.exprand(32, 1.0, 4.0);
var aarray11, aarray12 = Array.rand(32, 0.0, 1.0);
var parray11, parray12 = Array.fill(32, {1.0.rand2});
var farray21, farray22 = Array.exprand(32, 4.0, 8.0);
var aarray21, aarray22 = Array.rand(32, 0.0, 1.0);
var parray21, parray22 = Array.fill(32, {1.0.rand2});
var ratio = [15/16, 1, 16/15];
base11 = base12;
base21 = base22;
farray21 = farray22;
aarray21 = aarray22;
parray21 = parray22;
x.source = {
  loop{
    farray11 = farray12;
    aarray11 = aarray12;
    parray11 = parray12;
    farray12 = farray11 * Array.fill(32, {ratio.choose});
    aarray12 = Array.rand(32, 0.0, 1.0);
    parray12 = Array.fill(32, {1.0.rand2});
    farray21 = farray22;
    aarray21 = aarray22;
    parray21 = parray22;
    farray22 = farray21 * Array.fill(32, {ratio.choose});
    aarray22 = Array.rand(32, 0.0, 1.0);
    parray22 = Array.fill(32, {1.0.rand2});
    ~out.setn(
      \dur, dur,
      \base11, base11,
      \base12, base12,
      \base21, base21,
      \base22, base22,
      \farray11, farray11.postln,
      \farray12, farray12.postln,
      \aarray11, aarray11.normalizeSum,
      \aarray12, aarray12.normalizeSum,
      \parray11, parray11,
      \parray12, parray12,
      \farray21, farray21.postln,
      \farray22, farray22.postln,
      \aarray21, aarray21.normalizeSum,
      \aarray22, aarray22.normalizeSum,
      \parray21, parray21,
      \parray22, parray22 ).rebuild;
    dt.wait;
  }
}
)
```

```
// 上行していく和声と下行していく和声の組み合わせ
(
```

```

var dur = 3;
var dt = 3;
var base11, base12 = 441;
var base21, base22 = 441;
var farray11, farray12 = Array.exprand(32, 1.0, 4.0);
var aarray11, aarray12 = Array.rand(32, 0.0, 1.0);
var parray11, parray12 = Array.fill(32, {1.0.rand2});
var farray21, farray22 = Array.exprand(32, 1.0, 4.0);
var aarray21, aarray22 = Array.rand(32, 0.0, 1.0);
var parray21, parray22 = Array.fill(32, {1.0.rand2});
var ratio1 = [15/16, 1];
var ratio2 = [1, 16/15];
base11 = base12;
base21 = base22;
farray21 = farray22;
aarray21 = aarray22;
parray21 = parray22;
x.source = {
  loop{
    farray11 = farray12;
    aarray11 = aarray12;
    parray11 = parray12;
    farray12 = farray11 * Array.fill(32, {ratio1.choose});
    aarray12 = Array.rand(32, 0.0, 1.0);
    parray12 = Array.fill(32, {1.0.rand2});
    farray21 = farray22;
    aarray21 = aarray22;
    parray21 = parray22;
    farray22 = farray21 * Array.fill(32, {ratio2.choose});
    aarray22 = Array.rand(32, 0.0, 1.0);
    parray22 = Array.fill(32, {1.0.rand2});
    ~out.setn(
      \dur, dur,
      \base11, base11,
      \base12, base12,
      \base21, base21,
      \base22, base22,
      \farray11, farray11.postln,
      \farray12, farray12.postln,
      \aarray11, aarray11.normalizeSum,
      \aarray12, aarray12.normalizeSum,
      \parray11, parray11,
      \parray12, parray12,
      \farray21, farray21.postln,
      \farray22, farray22.postln,
      \aarray21, aarray21.normalizeSum,
      \aarray22, aarray22.normalizeSum,
      \parray21, parray21,
      \parray22, parray22 ).rebuild;
    dt.wait;
  }
}
)

```

// 上行/下行していきながらしだいに消えていく(疎になっていく)和声

```

(
var dur = 1;
var dt = 5;
var base11, base12 = 441;
var base21, base22 = 441;
var farray11, farray12 = Array.exprand(32, 1.0, 4.0);
var aarray11, aarray12 = Array.fill(32, 1.0);
var parray11, parray12 = Array.fill(32, {1.0.rand2});
var farray21, farray22 = Array.exprand(32, 1.0, 4.0);
var aarray21, aarray22 = Array.fill(32, 1.0);
var parray21, parray22 = Array.fill(32, {1.0.rand2});
var ratio1 = [15/16, 1];

```

```

var ratio2 = [1, 16/15];
base11 = base12;
base21 = base22;
farray21 = farray22;
aarray21 = aarray22;
parray21 = parray22;
x.source = {
  loop{
    farray11 = farray12;
    aarray11 = aarray12;
    parray11 = parray12;
    farray12 = farray11 * Array.fill(32, {ratio1.choose});
    aarray12 = aarray11 * Array.fill(32, {[0.5, 1.0].choose});
    parray12 = Array.fill(32, {1.0.rand2});
    farray21 = farray22;
    aarray21 = aarray22;
    parray21 = parray22;
    farray22 = farray21 * Array.fill(32, {ratio2.choose});
    aarray22 = aarray21 * Array.fill(32, {[0.5, 1.0].choose});
    parray22 = Array.fill(32, {1.0.rand2});
    ~out.setn(
      \dur, dur,
      \base11, base11,
      \base12, base12,
      \base21, base21,
      \base22, base22,
      \farray11, farray11.postln,
      \farray12, farray12.postln,
      \aarray11, aarray11.normalizeSum,
      \aarray12, aarray12.normalizeSum,
      \parray11, parray11,
      \parray12, parray12,
      \farray21, farray21.postln,
      \farray22, farray22.postln,
      \aarray21, aarray21.normalizeSum,
      \aarray22, aarray22.normalizeSum,
      \parray21, parray21,
      \parray22, parray22 ).rebuild;
    dt.wait;
  }
}
)

```

// 和音全体は上行/下行し、基音は逆に下行/上行しながらしだいに消えていく和声

```

(
var dur = 1;
var dt = 5;
var base11, base12 = 441;
var base21, base22 = 441;
var farray11, farray12 = Array.exprand(32, 1.0, 4.0);
var aarray11, aarray12 = Array.fill(32, 1.0);
var parray11, parray12 = Array.fill(32, {1.0.rand2});
var farray21, farray22 = Array.exprand(32, 1.0, 4.0);
var aarray21, aarray22 = Array.fill(32, 1.0);
var parray21, parray22 = Array.fill(32, {1.0.rand2});
var ratio1 = [15/16, 1];
var ratio2 = [1, 16/15];
base11 = base12;
base21 = base22;
farray21 = farray22;
aarray21 = aarray22;
parray21 = parray22;
x.source = {
  loop{
    base11 = base12.postln;
    farray11 = farray12;
    aarray11 = aarray12;
    parray11 = parray12;
    base12 = base11 * ratio2.choose;

```

```

farray12 = farray11 * Array.fill(32, {ratio1.choose});
aarray12 = aarray11 * Array.fill(32, {[0.5, 1.0].choose});
parray12 = Array.fill(32, {1.0.rand2});
base21 = base22.postln;
farray21 = farray22;
aarray21 = aarray22;
parray21 = parray22;
base22 = base21 * ratio1.choose;
farray22 = farray21 * Array.fill(32, {ratio2.choose});
aarray22 = aarray21 * Array.fill(32, {[0.5, 1.0].choose});
parray22 = Array.fill(32, {1.0.rand2});
~out.setn(
  \dur, dur,
  \base11, base11,
  \base12, base12,
  \base21, base21,
  \base22, base22,
  \farray11, farray11.postln,
  \farray12, farray12.postln,
  \aarray11, aarray11.normalizeSum,
  \aarray12, aarray12.normalizeSum,
  \parray11, parray11,
  \parray12, parray12,
  \farray21, farray21.postln,
  \farray22, farray22.postln,
  \aarray21, aarray21.normalizeSum,
  \aarray22, aarray22.normalizeSum,
  \parray21, parray21,
  \parray22, parray22 ).rebuild;
dt.wait;
}
}
)

// マトリックスの部分集合としての疎な和音
(
var dur = 10;
var dt = 10;
var base11, base12 = 441;
var base21, base22 = 441;
var farray11, farray12 = Array.exprand(32, 1.0, 4.0);
var aarray11, aarray12 = Array.fill(32, 1.0);
var parray11, parray12 = Array.fill(32, {1.0.rand2});
var farray21, farray22 = Array.exprand(32, 4.0, 8.0);
var aarray21, aarray22 = Array.fill(32, 1.0);
var parray21, parray22 = Array.fill(32, {1.0.rand2});
var ratio = [15/16, 1, 16/15];
base11 = base12;
base21 = base22;
farray21 = farray22;
aarray21 = aarray22;
parray21 = parray22;
x.source = {
  loop{
    farray11 = farray12;
    aarray11 = aarray12;
    parray11 = parray12;
    farray12 = farray11 * Array.fill(32, {ratio.choose});
    aarray12 = Array.fill(32, {[0.001, 1.0].wchoose([0.9, 0.1]}));
    parray12 = Array.fill(32, {1.0.rand2});
    farray21 = farray22;
    aarray21 = aarray22;
    parray21 = parray22;
    farray22 = farray21 * Array.fill(32, {ratio.choose});
    aarray22 = Array.fill(32, {[0.001, 1.0].wchoose([0.9, 0.1]}));
    parray22 = Array.fill(32, {1.0.rand2});
    ~out.setn(
      \dur, dur,
      \base11, base11,

```



```
\base12, base12,  
\base21, base21,  
\base22, base22,  
\farray11, farray11.postln,  
\farray12, farray12.postln,  
\aarray11, aarray11.normalizeSum.postln,  
\aarray12, aarray12.normalizeSum.postln,  
\parray11, parray11,  
\parray12, parray12,  
\farray21, farray21.postln,  
\farray22, farray22.postln,  
\aarray21, aarray21.normalizeSum.postln,  
\aarray22, aarray22.normalizeSum.postln,  
\parray21, parray21,  
\parray22, parray22 ).rebuild;  
dt.wait;  
}  
}  
)  
  
x.stop;
```