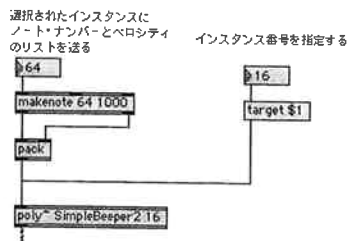


じノート・ナンバーとベロシティのリストを渡していたのではポリフォニー処理は実現されない。逆に言えば、各サブ・パッチに個別にリストを渡してやらなければならない。

MSPではpolyオブジェクトによって内部的にコピーされた個々のサブ・パッチをインスタンスという名前と呼んでいる。個々のインスタンスには内部的にインスタンス番号が付けられ、ポリフォニー数が16であれば、1~16のインスタンス番号でこれらは区別される。特定のインスタンスを選んでメッセージやシグナルを送るためには、まずpolyオブジェクトに“target インスタンス番号”というメッセージを与えてやる。5-15-11のパッチの場合、16番のインスタンスを選択し、そのインスタンスに対してのみノート・ナンバーとベロシティのリストを送っている。ちなみにtarget -1をpolyオブジェクトに渡すとすべてのインスタンスに対して同じメッセージやシグナルが送られる。

#### ■5-15-11 targetメッセージによるインスタンスの選択



もう1度5-15-9のパッチに戻ってtargetメッセージがどのように使用されているのを見よう。インスタンス番号として利用されているのはpolyオブジェクトが出力するボイス番号である。target \$1, \$2 \$3によって、まず“target polyが出力するボイス番号”のリストがpolyオブジェクトに渡されインスタンスを選択し、続けて“ノート・ナンバーベロシティ”のリストがそのインスタンスに対してのみ送られる。こうして16音ポリフォニーが実現することになる。

polyオブジェクトは“1つのオブジェクトで1つの機能を果たす”という、Maxが長年堅持してきた考え方を1歩踏み越えるものだ。その分インスタンスの概念やそのルーティングなどでとまどう面もあるかもしれないが、ポリフォニー処理を効率的に実現するためにはぜひとも習得したいオブジェクトである。

## 5-16 空間処理

音は時間の中で知覚される現象であると同時に、空間の中で聞こえる現象でもある。われわれは音色やピッチだけではなく、その位置によってもある音と別の音を区別する。あるいは目をつぶっていても音を聞いただけで、自分がいる空間の大きさを推測することができる。音と空間は不可分に結びついている。メロディや和声、リズムといった音楽そのものの要素以外に、音の空間処理は古くから作曲家の大きな関心事だった。オーケストラの楽器の配置は物理的な空間処理の一例だし、今日CDなどで聞く音楽の多くは、制作過程で綿密な空間処理とその効果を計算して作られている。ここではMSPプログラミングにおける空間処理について紹介したい。

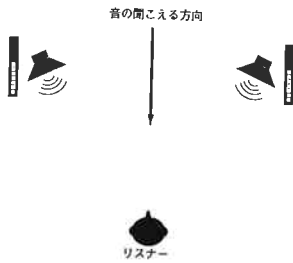
### ● パニングの意味

通常、音は空間内の特性の位置から聞こえるように知覚され、その位置を定位と呼ぶ。音の定位を操作する最も効果的な方法は、実際に音源を空間的に配置することだ。オーケストラの楽器配置、多数のマルチチャンネル・スピーカーの配置、最近ではドルビー・サラウンド5.1に見られるように6つのスピーカーでリスナーを囲む方法などもある。

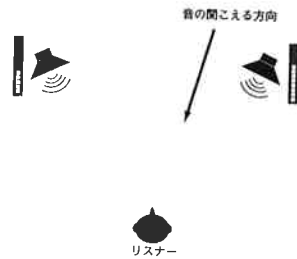
しかし、こうした音源の空間配置は、現時点ではコンサート会場や映画館などの特定の場所でしか実現できないものであり、広く一般のリスナーにとってなじみ深いのは、やはり2個のスピーカーあるいはヘッドフォンからなる2チャンネル・ステレオ再生になるだろう。ここでは2チャンネル・ステレオ再生を前提にどのような空間処理が可能なのかを紹介していくことにする。

音の定位を操作することをパン(pan)あるいはパニング(panning)と言う。ステレオ・パニングの最も基本的な方法は、水平方向のパニングだ。これは、左右のスピーカーから同じ音を出し、それぞれの音量を調整することで音の定位を操作する。もし左右のスピーカーの音量が同じであれば、5-16-1のようにわれわれはスピーカーの真ん中から音が聞こえてくるように知覚する。

## ■5-16-1 音のセンター定位



## ■5-16-2 音量バランスの調整による定位の変更



ここで、右チャンネルのスピーカーから出る音の音量を大きくし、左チャンネルのスピーカーから出る音の音量を小さくすると、音の聞こえる方向は5-16-2のように右方向に変わるだろう。

## ● リニア・パンニング

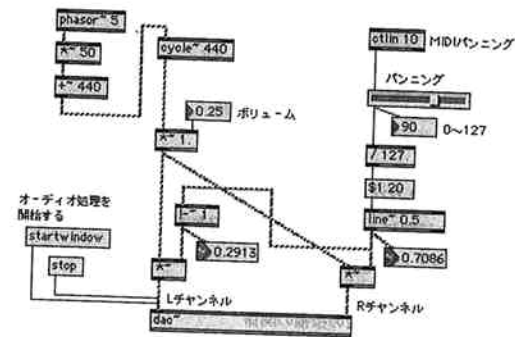
パンニングのプログラミングでは、一方のスピーカーから出る音の音量をどれくらい大きくし、それに連動して他方の音量をどれくらい小さくするかという左右の音量バランスにおいて、いくつかの方法がある。

最も単純な方法は、左右の振幅の和が常に一定になるように調整することで、5-16-3のパッチようになる。テスト用の音源となるオシレーターはcycle`オブジェクトであるが、変化のない一定のサイン波サウンドは音の定位を感覚しにくいいため、ここではその周波数をphasor`オブジェクトにより周期的に変化させている。cycle`オブジェクトの出力信号は\*`オブジェクトでボリューム調整された後、左右のオーディオ・チャンネルから出力されるが、dac`オブジェクトの左右のインレットの直前でパンニングされている。

パンニングは、MIDIコントロール・チェンジ・メッセージあるいはhslider《Horizontal Slider》オブジェクトのマウス操作によりコントロールしている。MIDIではコントロール・

チェンジ・ナンバーの10がパンニングに割り当てられている。hsliderオブジェクトの0~127の出力は0~1の信号に変換されて右チャンネルの振幅をコントロールする。一方、\*`オブジェクトは右インレットに受け取る信号値あるいはアークメントに書かれた数値から、左インレットに受け取る信号値を減算したものを信号として出力する。ちょうど\*`の左右のインレットが逆の順序になっていると考えればよい。ここでは1から入力信号の値を減算する。つまり、このパンニングでは左右の振幅の和が常に1になるように調整されているわけだ。

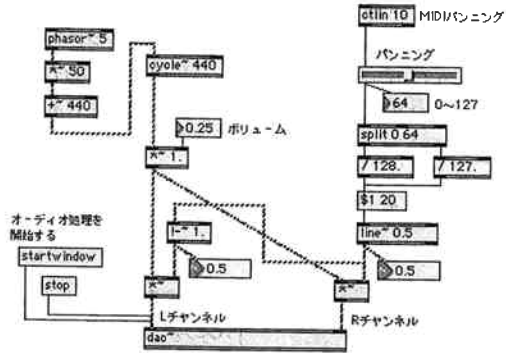
## ■5-16-3 リニア・パンニング・パッチ



ただし、5-16-3のパッチに問題はないかと言うとそうとも言えない。通常、MIDIコントロール・チェンジ・メッセージでは中央値を64とする場合が多い。このときパンニングにおける左右の振幅はどちらも0.5になっている必要がある。しかし5-16-3のパッチではhsliderオブジェクトが64を出力すると右チャンネルの振幅は0.5039……、右チャンネルの振幅は0.4960……となり厳密な意味での中央値にはなっていないのである。

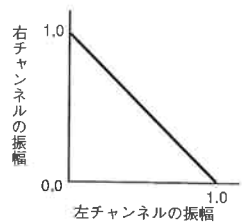
そこで次の5-16-4のパッチではhsliderオブジェクトが64を出力すると左右の振幅がどちらも0.5になるように多少の工夫を加えてみた。hsliderオブジェクトの出力はsplitオブジェクトにより0~64と65~127に分岐され、0~64は128で除算されて64のときに0.5になるよう調整されているのが分かるだろう。

■5-16-4 正確に中央に定位させる工夫



さて、このようなパンニングの方法はリニア（線形）パンニングと呼ぶことができるだろう。つまり左右の振幅の和が常に一定であるため、それらの関係を2次元のグラフで表現した場合に直線で描けるからだ。

■5-16-5 リニア・パンニングにおける左右の振幅の関係



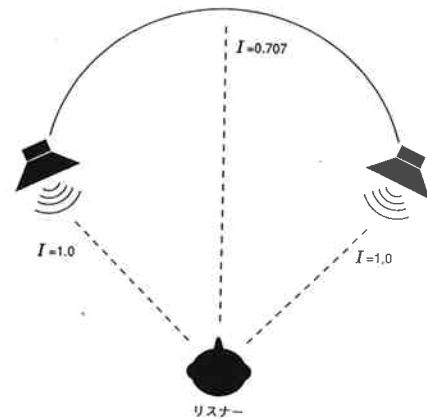
リニア・パンニングでは、聴感上の音の定位は、5-16-6のように中央になるほどリスナーから遠ざかって聞こえる。これは知覚される音の大きさが、音の強度 (intensity) に比例することから生じる。音の強度と左右のチャンネルの振幅の関係は5-16-6の式で

table 行外

画像解不

イナロフ

■5-16-6 リニア・パンニングでの音の定位と強度の関係



$$I = \sqrt{L\text{Amp}^2 + R\text{Amp}^2}$$

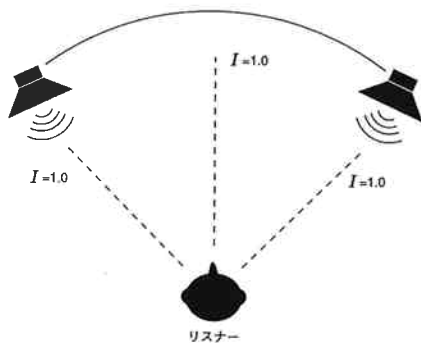
I = 音の強度  
Lamp = 左チャンネルの振幅  
RAmp = 右チャンネルの振幅

表現される。ちょうど定位が中央になったとき、左右のチャンネルの振幅はどちらも0.5なので、その2乗の和の平方根は0.707……となって、強度は定位が左端あるいは右端にあるときよりも小さくなる。このため中央になるほど遠ざかって聞こえることになる。

● 等距離パンニング

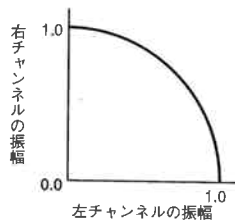
リニア・パンニングに対し、定位が移動してもリスナーから等距離に聞こえるパンニングの方法を等距離パンニングと呼ぶことができる。一般にミキサー機器などに付属しているパンニングつまみ(パンポット)はこのパンニングの方式を利用していることが多い。等距離パンニングでの音の定位と強度の関係は5-16-7のようなイメージになるだろう。

■5-16-7 等距離パンニングでの音の定位と強度の関係



等距離パンニングでの左右チャンネルの幅の関係を2次元のグラフで表現すると5-16-87ようになる。リニア・パンニングと違って左右の振幅の和が一定ではない。

■5-16-8 等距離パンニングにおける左右の振幅の関係

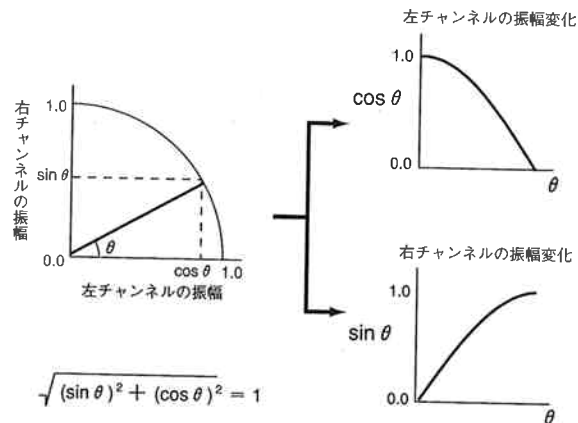


このグラフを見ると、それが半径1の円弧になっていることが分かる。これは三角関数で言う単位円と同じものだと考えることができ、円周上の特定の点(つまり特定のパンニング位置)での左右の振幅の値は、 $\sin$ や $\cos$ の三角関数を使って表現することができる。5-16-9のように円周グラフ上の点を角度 $\theta$ で表すと、左チャンネルの振幅は $\cos \theta$ 、右チャンネルの振幅は $\sin \theta$ となる。 $\theta$ は0~360度の角度、あるいは0~ $2\pi$ の

ラジアンで表現されることが一般的だ。あるいは後に紹介するように任意の範囲の数値(例えば0~1)に置き換えることもできる。ここで、 $\theta$ が変化する(パンニングの定位が移動すると)、左右チャンネルの振幅がどのように変化するかは、それぞれ $\cos \theta$ 、 $\sin \theta$ のグラフで示すことができる。リニア・パンニングの場合は直線となるが、ここでは非線形のカーブを描く。

さらに、音の強度は、左右の振幅の2乗を足したものの平方根であるが、 $\cos \theta$ 、 $\sin \theta$ の2乗を足したものの平方根は、三角関数の定理により必ず1になる。これによっても、定位が移動しても音の強度は一定に保たれ、その結果、リスナーから等距離に聞こえることが示されている。

■5-16-9 等距離パンニングにおける三角関数の考え方



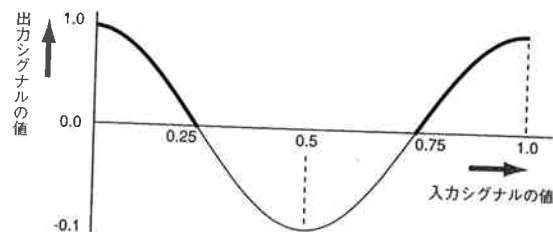
等距離パンニングのプログラミングは、ユーザーが決定する定位をもとに、左右の振幅の値を計算することで行う。つまり $\cos$ 、 $\sin$ の計算を行えばよい。MSPではこのためにいくつかのオブジェクトが用意されているが、ここでは $\cos$ オブジェクトを使ってみることにする。

$\cos$  は、 $\cos$ 関数のシグナル計算を行うオブジェクトで、1周期が0~1の範囲にマップされている。入力シグナルと出力シグナルの関係は5-16-10ようになる。

ここで、前の5-16-9の左右チャンネルの振幅変化のグラフを見比べると、左チャンネルの振幅変化は入力信号が0~0.25の部分、右チャンネルの振幅変化は0.75~1.0の部分にそれぞれ対応していることが分かる。これらの部分をプログラミングに利用すればよいだろう。

先ほど、右チャンネルの振幅は $\sin \theta$ で得られると説明したが、 $\sin \theta$ と $\cos \theta$ は位相が異なっているだけなので、 $\cos$ オブジェクトだけでプログラミングを行うことができる。

■5-16-10  $\cos$ オブジェクトの入力信号と出力信号の関係

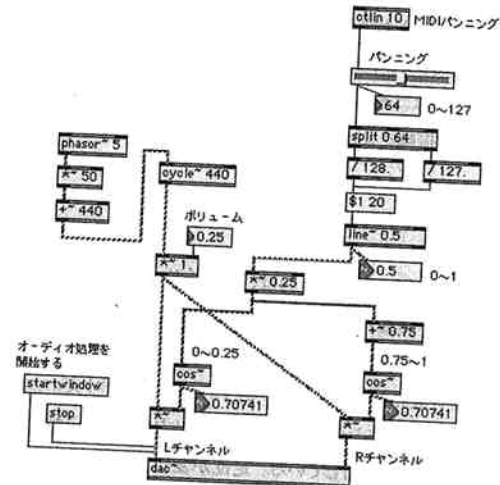


5-16-11のパッチは、 $\cos$ オブジェクトを使って等距離バンニングを実現している例だ。MIDI入力あるいはhsliderオブジェクトの操作によりユーザーが決めたバンニング値は、lineオブジェクトから0~1の範囲のシグナルとして出力される。これを0.25倍して0~0.25とし、 $\cos$ オブジェクトで $\cos$ 関数の計算をさせて左チャンネルの振幅を決定。さらにこれに0.75を足して0.75~1とし同様に右チャンネルの振幅を決定している。

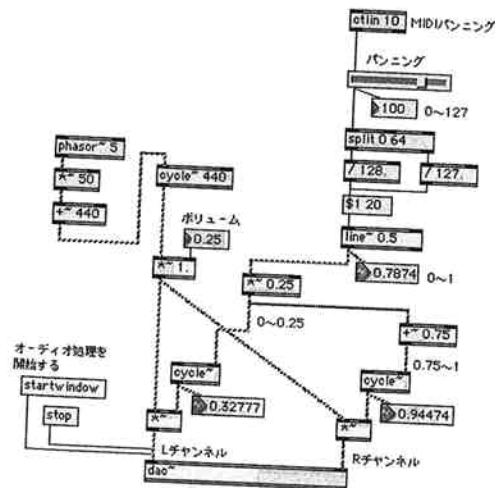
また5-16-11のパッチは、 $\cos$ オブジェクトの代わりにcycleオブジェクトを用いて5-16-12のように書き換えることもできる。cycleオブジェクトはこれまでサイン波オシレーター、あるいは波形テーブル参照オシレーターとして紹介してきた。オシレーターとは外部から何の力ももらわないでも、それ自身で動作を行う発振器という意味だった。通常、cycleオブジェクトは左インレットに数値やシグナルを受け取り、それを周波数として解釈して繰り返しサイン波波形のシグナルを出力する。

しかし、cycleオブジェクトは右インレットにシグナルを受け取ると、それ自身で動作を行うことを止めて、外部の力で駆動されて波形テーブルを読み出すオブジェクトにな

■5-16-11  $\cos$ オブジェクトを利用した等距離バンニングのパッチ例



■5-16-12 cycleオブジェクトを利用した等距離バンニングのパッチ例



る。cycle`オブジェクトは初期状態で内部にコサイン波形1周期分の波形テーブルを持っているため、これをline`オブジェクトからの出力信号で読み出すことになる。cycle`オブジェクトが右インレットから受け取る数値や信号は、位相の値として解釈され、位相は0~1の範囲で、ちょうど5-16-10のように波形の1周期に対応している。

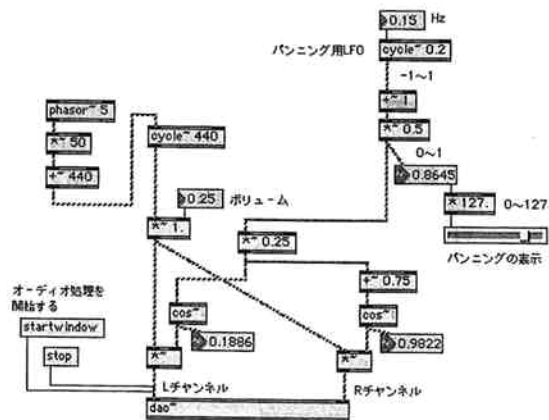
こうしてcycle`オブジェクトはコサイン波形テーブルの一部を読み出し、その出力信号で左右のチャンネルの振幅をコントロールすることができる。

### ● オート・パンニング

これまで紹介してきたパンニングのパッチ例は、音の定位をユーザーが決めるものだった。今度は、5-16-12の等距離パンニングのパッチを改良して、LFOを使って音の定位を自動的に移動させてみよう。

LFOとして使うのはサイン波オシレーターであるcycle`オブジェクトだ。cycle`オブジェクトは-1~1の範囲のサイン波信号を周期的に出力するが、これを0~1の範囲に変換してオート・パンニングに利用している。定位が変化する状況をグラフィカルに確認できるよう、フロート・ナンバー・ボックス(flonum《Float Number Box》)の右アウトレット(信号の値を実数で出力する)をhsliderオブジェクトにつないでいる。

■5-16-13 オート・パンニングのパッチ例



オーディオ処理を開始するとサウンドが左右に繰り返し移動する。等距離パンニングのパッチなので移動しても聴感上の音の大きさは変わらない。LFOとして使うオシレーターはcycle`オブジェクトだけでなく、phasor`やrand`オブジェクトなどいろいろと試してみると面白い効果が得られるだろう。