

# COMPOSER L'ESPACE EN ENVIRONNEMENT CSOUND : LE CAS DE *SIDERAL*

*Isabel Pires*

ATER

Université de Marne-la-Vallée  
Cité Descartes – 5, bd. Descartes, Champs-sur-Marne  
77454 Marne-la-Vallée  
Université Paris VIII – CICM – MSH Paris Nord  
isabel.pires@univ-mlv.fr ; isabel\_pires@excite.com

## RÉSUMÉ

Dans cet article, nous aborderons certains aspects du fonctionnement de la perception auditive de l'espace en tant qu'éléments opératoires utilisables dans la création sonore et musicale.

Nous analyserons les techniques de mise en espace du son en environnement CSound utilisées dans *Sideral*<sup>1</sup>. Dans ce cadre, nous étudierons la programmation modulaire dans cet environnement en s'intéressant à des techniques de connexion d'unités fonctionnelles autonomes et d'acheminements multiples dans un même instrument. Nous observerons la mise en espace du son par paramétrage d'angle en utilisant l'opcode *locsig* et par paramétrage linéaire utilisant l'opcode *space*.

## 1. INTRODUCTION

Le son est aujourd'hui une nouvelle matière de l'espace. Au cours du XXe siècle, il a acquis une matérialité, une plasticité qui permet aux compositeurs de le travailler comme s'il s'agissait d'une matière palpable. Ce changement est dû tant au développement des technologies de manipulation sonore et qu'à celui des connaissances psycho-acoustiques du phénomène sonore.

Notre travail de composition de musique acousmatique nous a amené à étudier à la fois le fonctionnement de la perception auditive et les manipulations du son susceptibles de simuler des localisations ou mouvements sonores dans un espace physique spécifique. Cette étude et l'importance que nous attachons à la composition de l'espace à l'intérieur des œuvres musicales nous ont conduit à travailler sur l'environnement CSound en raison de sa flexibilité et de sa rigueur. Nous avons ainsi développé des procédés de programmation qui permettent un traitement très fin de la composante « espace » au sein des œuvres. Nous présenterons ici des exemples de ce travail de traitement spatial, en nous appuyant sur l'œuvre *Sideral*.

Nous évoquerons d'abord certains des éléments les plus pertinents du processus cognitif de perception auditive mis en œuvre dans la localisation des sons dans un espace physique. Nous analyserons ensuite quelques aspects concernant la composition de la spatialité de l'œuvre. Puis nous envisagerons les fonctions CSound que nous avons utilisées pour le traitement spatial des sons dans l'œuvre *Sideral* ainsi que les procédés de programmation modulaire. Nous finirons en exposant des exemples de cette programmation afin de clarifier son fonctionnement.

## 2. PROCESSUS DE PERCEPTION AUDITIVE DE LA LOCALISATION SONORE

La perception auditive est un des processus cognitifs qui permettent d'accéder à la connaissance de l'espace. La perception auditive des sons dans l'espace nécessite le recueil de différents types d'informations à propos du phénomène sonore perçu : la localisation de l'événement sonore dans un espace physique, et les « empreintes » liées aux caractéristiques de la source sonore et de l'environnement spatial dans lequel l'événement se produit.

Dans cet article, nous aborderons uniquement ce qui concerne la perception de la localisation spatiale des sons.

Les études psycho-acoustiques réalisées pour la compréhension de la perception auditive de la localisation sonore ont mis en évidence divers mécanismes cognitifs qui permettent une rapide estimation de la localisation d'une source sonore perçue[5]. Cette localisation auditive (estimation de la direction et de la distance) est basée sur un ensemble de processus cognitifs de traitement et d'analyse effectués par le cerveau sur les signaux captés par les deux oreilles (écoute binaurale)[3]. Cette estimation de la localisation spatiale d'un phénomène sonore perçu est effectuée sur trois dimensions principales : plan horizontal, plan vertical et distance.

<sup>1</sup> Œuvre acousmatique de Isabel Pires composée en 2006.

## 2.1. Perception auditive dans le plan horizontal

Les différences interaurales de temps, de niveau et de phase sont responsables de la perception de la localisation d'un événement sonore sur le plan horizontal (azimut). Ces différences sont estimées par le système auditif en évaluant le décalage en temps, niveau et phase[3] entre les sons qui arrivent aux deux oreilles. Cette analyse est fondée sur la comparaison des disparités perçues entre les caractéristiques des phénomènes sonores créés par le champ acoustique au niveau des deux oreilles.

## 2.2. Perception auditive dans le plan vertical

La perception du positionnement du son sur le plan vertical (élévation) dépend principalement du type de filtrage exercé par le pavillon de l'oreille mais aussi des asymétries spectrales créées par les réflexions sur les épaules et la tête. Ce filtrage provoque des zones de résonance spectrale qui aideront le système auditif à estimer la provenance du phénomène sonore. Il faut cependant dire que la perception de la localisation sonore sur le plan vertical ne correspond pas nécessairement au positionnement réel du son mais semble être produite en fonction des caractéristiques spectrales du son source.

## 2.3. Perception auditive de la distance

L'estimation de la distance de positionnement d'une source sonore, quant à elle, dépend de l'évolution des intensités, du rapport entre le son direct et le son réverbéré ainsi que de l'évolution de la densité spectrale du son. Par exemple, c'est le cas de phénomènes sonores dont l'intensité augmente ou diminue accompagné d'un changement proportionnel du rapport entre le son original et le son réverbéré, ainsi que des changements au niveau spectral. L'absorption inégale des fréquences aiguës et graves pendant la propagation du son permet également d'estimer un éloignement ou rapprochement de la source sonore réelle ou virtuelle.

La perception binaurale de la localisation sonore est un sujet d'étude très vaste dont ces brèves lignes ne font qu'effleurer l'importance. Leur unique objectif est d'énoncer des aspects perceptifs qui entrent en ligne de compte dans le processus de composition musicale de l'espace. En effet, la manipulation, à l'aide de logiciels et machines adéquats, de cet ensemble de facteurs permet la simulation artificielle de phénomènes perceptifs de localisation et de mouvement sonores.

## 3. COMPOSER DE L'ESPACE SONORE

Les aspects de la perception auditive du phénomène sonore qui concernent la localisation des sons dans l'espace, abordés précédemment, peuvent constituer des éléments opératoires dans la création sonore et musicale[13] [14].

Or, si la spatialisation est la pratique générale de mise en espace des sons, la simulation de la localisation implique des techniques spécifiques et outils précis qui permettent la manipulation des paramètres du phénomène sonore entrant en ligne de compte dans la perception auditive[15]. Bien que le rapport entre les représentations de la localisation sonore prétendue, le résultat des algorithmes de calcul de la répartition du son dans les divers haut-parleurs et la perception effective d'une localisation sonore donnée, ne soit pas direct, ce travail de composition de l'espace sonore à l'intérieur de l'œuvre reste essentiel car opératoire.

C'est la détermination, à l'intérieur d'une œuvre musicale, de schémas d'étalement des sons dans un espace physique imaginaire, basés sur des connaissances de la perception auditive, qui permet la création de réseaux opératoires au niveau de la composition de l'espace de l'œuvre, au même titre que ceux créés à d'autres niveaux (rythme, hauteur, texture, etc.). Ainsi, la détermination de la position du son dans un espace virtuel défini est une étape cruciale de la production d'une œuvre musicale car la « composition de l'espace » peut aussi être un élément structurant de l'œuvre comme n'importe quel autre. De ce fait, le traitement, la simulation et la manipulation virtuelle du déploiement spatial des sons dans une œuvre permettent de créer des ambiances sonores complexes et de structurer la musique elle-même.

Mais, l'utilisation du facteur « espace » dans la création musicale est une problématique complexe et double puisqu'il s'agit à la fois de percevoir et de construire.

Percevoir : dans l'acte d'écoute, il s'agit d'interpréter, de comprendre, d'estimer la direction et la distance à laquelle la source sonore se trouve, mais aussi de capter les spécificités de la source sonore, les caractéristiques acoustiques du lieu de production du son et celles du lieu de propagation de ce son.

Construire : cet acte implique des espaces imagés ou imaginaires, des idées abstraites mais opératoires qui permettent au compositeur de construire à la fois des simulations de situations perceptives possibles et des cohérences musicales. Dans cet acte de construire le compositeur jouera, dans l'espace interne de l'œuvre, avec des « images d'espaces » construites virtuellement par des procédés de synthèse ou captées lors d'enregistrements microphoniques, ces « images d'espaces » étant la trace d'une sorte d'ambiance induite par l'espace physique autour de la source sonore. Si la source sonore est réelle dans le cas d'un enregistrement, l'image d'espace correspondra à l'empreinte de l'acoustique du lieu sur le son dans le moment de sa captation. Si la source sonore est virtuelle, c'est-à-dire s'il s'agit d'un son de synthèse, l'image d'espace se superposera complètement au son produit ou alors elle sera engendrée artificiellement par le compositeur. L'engendrement d'images d'espace de façon artificielle donne au compositeur la liberté de créer des espaces imaginaires, potentiellement impossibles à produire dans le réel. Le compositeur peut par exemple imaginer un espace d'étalement du sonore qui soit à la fois rigide

et malléable, il peut créer l'image d'un espace dans lequel des murs solides s'étirent ou se contractent, deviennent sphériques ou anguleux calculant à l'aide d'outils informatiques adaptés, le comportement acoustique d'un son spécifique dans cet espace imaginaire, construisant ainsi, autour du son, l'image d'un espace virtuel.

#### 4. TRAITEMENT DU FACTEUR ESPACE À L'AIDE DU LOGICIEL CSOUND

Comme nous l'avons vu, la perception auditive de la localisation sonore dépend des structures de fonctionnement du système auditif, des informations sensorielles et cognitives disponibles ainsi que des conditions d'écoute.

Or ces éléments peuvent aujourd'hui être manipulés, et donc simulés virtuellement avec beaucoup de rigueur à l'aide de multiples logiciels dont CSound. Nous analyserons ici uniquement le traitement de la mise en espace des sons en environnement CSound par rapport au plan horizontal et la distance,

Dans le cas de l'utilisation de CSound pour la mise en espace des sons, le calcul de certaines de ces variations peut être réalisé de façon automatique en donnant tout simplement au logiciel des coordonnées pour la simulation virtuelle de la localisation ou du déplacement des sources sonores. Ainsi l'utilisation des paires d'opcodes<sup>2</sup> *locsig* / *locsend* et *space* / *spsend* présentés ici permettent aisément, à l'aide d'une programmation à la fois simple, flexible et claire, de créer des espaces sonores complexes et riches dans le domaine des transparences et de la superposition de couches sonores indépendantes mais solidaires.

Les opcodes *locsig* et *space* permettent, en environnement CSound, la détermination de localisations spatiales respectivement par paramétrage d'angle et par définition linéaire de points dans un plan cartésien dans espace virtuel représentatif. Ici nous prenons comme espace représentatif une salle imaginaire carrée dans laquelle un système de projection sonore quadriphonique est installé :

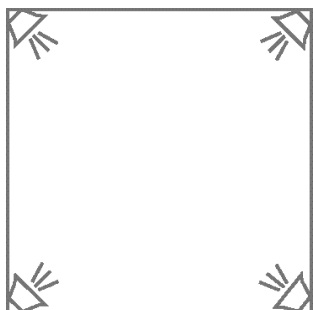


Figure 1. Schéma d'un système de projection sonore quadriphonique simple.

#### 4.1. Syntaxe des codes d'opération pour la définition de la localisation spatial de sons

##### 4.1.1. Code d'opération locsig

La fonction *locsig* permet le calcul automatique de la distribution du signal entre 2 ou 4 canaux en utilisant des valeurs en degrés. Ceci détermine le calcul automatique du niveau du son direct et la réverbération à envoyer à chaque haut-parleur individuellement à partir de la détermination d'un angle et d'une distance demandée de la localisation virtuelle des sons dans un espace acoustique. Ainsi, cette fonction permet de distribuer le son de façon circulaire et de calculer des distances par rapport à un certain point où la source sonore est virtuellement localisée.

Dans le paramétrage de l'opcode *locsig*, et par rapport à une « salle virtuelle » carrée, le degré 0° est placé devant à gauche, il correspond donc à l'emplacement habituel du canal 1 dans un système classique de projection sonore quadriphonique :

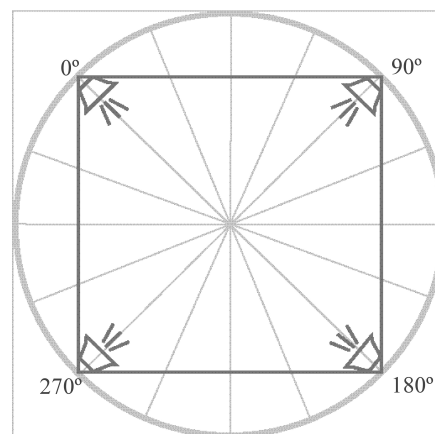


Figure 2. Représentation des positions en degrés paramétrables en utilisant l'opcode *locsig* par rapport à un espace virtuel de projection sonore quadriphonique.

Pour chaque localisation de la source sonore, la fonction *locsig* calcule automatiquement les niveaux du son parmi les quatre points d'écoute ainsi que la quantité de réverbération nécessaire.

Concernant le paramétrage d'une distance, l'opcode *locsend* calcule l'équilibre entre le niveau du son original et du son réverbéré pour créer la sensation perceptive de l'éloignement de la source sonore correspondant à la distance demandée.

L'opcode *locsig* peut utiliser la fonction coordonnée de l'opcode *locsend*. *Locsend* dépend de l'existence préalable de l'opcode *locsig* et, en fonction des coordonnées spatiales, de la distance déterminée et de la quantité de réverbération définie, *locsend* calcule la quantité de son à envoyer vers l'instrument de réverbération.

<sup>2</sup> Opcode : code d'opération.

La syntaxe de l'opcode *locsig* est la suivante :

```
a1, a2      locsig asig, kdegree, kdistance,
kreverbsend
a1, a2, a4, a3 locsig asig, kdegree, kdistance,
kreverbsend2
```

*kdegree* : correspond à une valeur en degrés compris entre 0° et 360°, pour une configuration de : a1=0°, a2=90°, a3=180°, a4=270°<sup>3</sup> (un *kdegree* = 45 impliquera un son balancé entre a1 et a2, donc placé devant au centre).

*kdistance* : correspond à la définition d'une certaine distance par rapport au centre de la salle virtuelle, une valeur supérieure à 1 impliquera un décroissement du niveau du signal et le calcul d'une certaine quantité de réverbération (en coordination avec l'opcode *locsend*) nécessaire à la simulation de la distance demandée.

*kreverbsend* : correspond à la quantité (en pourcentage) du signal direct qui, en rapport avec les facteurs de distance et de degré, à l'intérieur de l'opcode *locsend*, pourra être envoyé vers l'instrument de réverbération.

#### 4.1.2. Code d'opération *space*

La fonction *space* permet le calcul automatique de la distribution du signal en 4 canaux en utilisant la détermination des localisations et des mouvements sonores linéaires à travers la définition cartésienne d'un point de coordonnées *xy* dans un plan spatial virtuel.

L'opcode *space* peut aussi être coordonné avec l'opcode *spsend* qui (comme dans le cas de *locsend*) dépend de l'existence préalable de l'opcode *space*. Avec des coordonnées spatiales définies, la distance déterminée et la quantité de réverbération, *spsend* calcule la quantité de son à envoyer vers l'instrument de réverbération, de façon à bien simuler la localisation sonore demandée.

La syntaxe de l'opcode *locsig* est la suivante :

```
a1, a2, a3, a4 space asig, ifn, ktime,
kreverbsend, kx, ky
```

*ifn* : correspond à une table d'onde dans laquelle est définie une trajectoire. Ce type de fonction est créé en utilisant le générateur 28 (*Gen28*) qui contient des ensembles de 3 valeurs qui représentent le temps et les coordonnées cartésiennes *xy*.

Si nous choisissons de paramétrer *ifn* avec la valeur 0, l'opcode *space* prendra les valeurs correspondants aux paramètres *kx* et *ky*, respectivement les 5<sup>ème</sup> et 6<sup>ème</sup> entrées de l'opcode.

<sup>3</sup> Par rapport à une configuration quadraphonique standard, le positionnement de canaux 3 et 4 est inversé, il s'avère donc crucial de prendre en considération ce facteur au risque de voir les mouvements et / ou les localisations sonores complètement inversées en arrière-salle.

La configuration des coordonnées spatiales de l'opcode *space* est la suivante :

```
a1 = -1,1      a2 = 1,1
a3 = -1,-1     a4 = 1,-1
```

pour une valeur 0 qui se trouvera virtuellement au centre de la salle.

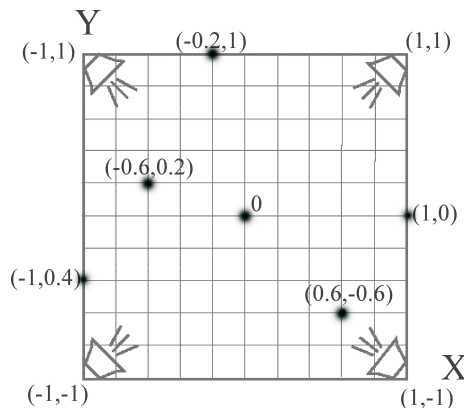


Figure 3. Schéma représentant la configuration des coordonnées spatiales de l'opcode *space*, ainsi que quelques positions dans un espace virtuel de projection quadraphonique.

*asig* : se réfère au signal d'entrée à spatialiser.

*ktime* : index temporel correspondant à l'évolution des coordonnées *xy*.

*kreverbsend* : comme dans l'opcode *locsig*, correspond à la quantité (en pourcentage) du signal direct qui pourra être envoyé vers l'instrument de réverbération à travers l'opcode *spsend*.

*kx*, *ky* seront les valeurs des coordonnées cartésiennes *xy* ou son évolution ; si cette valeur est 0,0, le son sera virtuellement localisé au centre de la salle. Ces paramètres sont pris en compte par *space* uniquement si *ifn* = 0.

## 5. LE TRAITEMENT SPATIAL DES SONS EN *SIDERAL*

### 5.1. Notes brèves sur l'œuvre

*Sideral* est une pièce quadraphonique, d'une durée de 10'17'', qui a été composée pendant le 1<sup>er</sup> trimestre 2006 et créée à Paris le 1<sup>er</sup> avril 2006 dans le cadre du festival *Itinéraires de Nuit*.

*Sideral* a été réalisée en utilisant des sons enregistrés dans des situations diverses et manipulés en utilisant des processus de traitement que nous avons développés essentiellement en environnement Max / Msp et CSound. Les fichiers son résultants de ces manipulations, que nous pouvons appeler pré-compositionnels, ont été à nouveau retravaillés à l'aide de divers procédés de synthèse (notamment la synthèse granulaire) en environnement CSound. CSound a

également été utilisé pour la génération de sons purement synthétiques, ainsi que pour le montage, le mixage et la composition de l'élément espace à l'intérieur de l'œuvre.

L'expérience de traitement de l'espace acquise précédemment, soit par la composition d'autres œuvres acousmatiques, soit par la pratique de projection sonore en direct, ont influencé la mise en place de certaines stratégies de composition de l'espace interne dans cette œuvre, notamment :

- la création de plans de transparence, couches sonores à la fois superposées et clairement indépendantes

- l'opposition entre trajectoire, localisation et ambiance sonore diffuse

- la création de mouvements sonores circulaires, oscillatoires et linéaires indépendants attribués à chaque son individuellement.

Les enjeux de la composition de l'espace sonore dans le cas de *Sideral*, nous a permis ainsi d'explorer la fiabilité et l'efficacité des multiples possibilités de traitement spatial en environnement CSound, créant des situations de remplissage de l'espace sonore à la fois complexes et rigoureuses.

## 5.2. Stratégies de remplissage de l'espace sonore en environnement CSound

### 5.2.1. Simulation de localisations et mouvements du son par rapport à un espace physique imaginaire – salle virtuelle

Dans *Sideral*, l'espace sonore est traité en considérant deux situations complémentaires : la localisation virtuelle des sons à l'intérieur d'un espace imaginaire donné et d'autres à l'extérieur. C'est-à-dire que dans la composition de cette œuvre nous avons pris comme stratégie la simulation de situations sonores dans lesquelles les sources sonores pouvaient être virtuellement localisées à l'intérieur de l'espace d'une « salle virtuelle » ou à une certaine distance à l'extérieur de cet espace.

La possibilité d'imposer à chaque son – à chaque « note » individuelle programmée en CSound – des paramètres différents permet de créer des sons indépendants. Ainsi, à chaque son individuel nous allons attribuer des spécificités spatiales de localisation et mouvement, cette spécificité individuelle étant maintenue indépendamment du nombre de sons présents simultanément. Cette possibilité que nous donne CSound de calculer des centaines de positions indépendantes pour chaque son évite la création d'amalgames sonores. Nous contournerons de cette manière la contrainte de traiter l'espace a posteriori comme un élément global.

En ce qui concerne la localisation de sons à l'intérieur de la « salle virtuelle », nous avons utilisé l'opcode *space* tant pour placer des sons à des points précis de l'espace que pour les déplacer dans cet espace, en utilisant des fonctions linéaires ou exponentielles qui

permettent de paramétrer des coordonnées *xy* ou leur évolution temporelle.

Concernant la simulation de placement des sons à l'extérieur de la « salle virtuelle », nous avons utilisé l'opcode *locsig* qui nous permet de définir un point quelconque dans une ligne courbe définie par un cercle imaginaire, dont le rayon déterminera la distance de placement virtuel de la source sonore. Des évolutions temporelles peuvent être définies en paramétrant des fonctions linéaires ou exponentielles qui contrôlent les mouvements virtuels de la source sonore dans le cercle ainsi que son éloignement ou rapprochement.

### 5.2.2. Création d'ambiances sonores, localisations spécifiques et trajectoires

Dans la composition de *Sideral*, nous avons construit un schéma de couches sonores – comme des sortes de rideaux de sons – qui en fonctionnant indépendamment créent des ambiances sonores complexes. De ce fait la définition de la superposition temporelle de sons paramétrés spatialement de façon disparate permet la création d'ambiances sonores complexes. Cette définition de couches sonores indépendantes est engendrée par la définition de types de mouvements divers mais simultanés.

Types de mouvement utilisés :

- mouvements linéaires (définis à l'intérieur ou à l'extérieur de l'espace virtuel)

- définition de distance, éloignement ou rapprochement de la source sonore

- déplacement de sons d'un point du cercle vers un autre point variable en termes de distance

- mouvements circulaires et / ou oscillants définis dans un arc de circonférence (définition variable de distance)

- mouvements circulaires et / ou oscillants doubles entre deux arcs d'angle coordonnés (définition variable de distance)

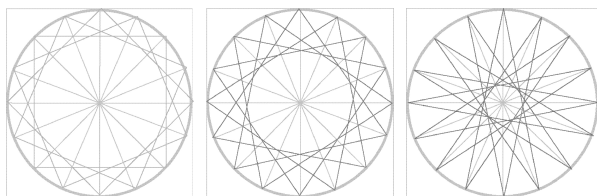
La programmation CSound de mouvements oscillants doubles permet l'utilisation coordonnée de mouvements sonores en sens direct ou contraire entre deux arcs d'angle symétriques. Cette possibilité autorise à coordonner la localisation et le mouvement spatial des canaux 1 et 2 de fichiers son stéréo<sup>4</sup>.

Cette stratégie est également applicable à des fichiers son mono puisque le même son peut être doublé et envoyé à différents points d'un cercle simultanément permettant ainsi de générer des mouvements oscillatoires directs ou contraires mais coordonnés entre eux et en rapport avec sa position initiale.

Nous avons créé des schémas de contrôle de ces mouvements qui permettent à la fois de coordonner les mouvements oscillatoires de ces « sons doubles »

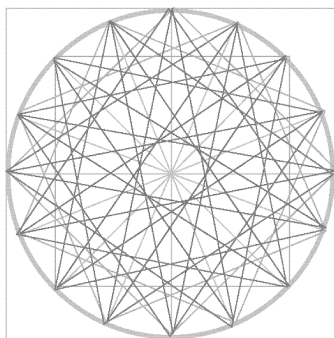
<sup>4</sup> Ce type de traitement spatial appliqué à des sons stéréo peut entraîner des singularités auditives si ces fichiers son ont déjà un composant spatial fort, principalement dans le cas de mouvements contraires coordonnés.

(soient-ils stéréo au départ ou mono dupliqués) et de visualiser en quelque sorte le remplissage virtuel de l'espace. Nous présentons dans les schémas suivants comment ont été imaginées puis programmés les localisations des sons doubles oscillatoires et leur coordination dans le processus de production des mouvements spatiaux.



**Figure 4.** Les trois schémas de coordination des mouvements oscillatoires dans l'œuvre *Sideral*.

La possibilité de faire coexister simultanément les trois schémas de mouvements coordonnés, représentés dans la figure précédente, permet d'obtenir le potentiel de remplissage de l'espace sonore, à travers des mouvements oscillants de sons doubles, représentable dans le schéma suivant :



**Figure 5.** Représentation des possibilités de remplissage de l'espace par combinaison des trois schémas précédents.

Dans ces schémas, nous avons négligé le facteur distance pour des raisons de clarté de représentation, il s'avère donc nécessaire d'évoquer le fait que chaque localisation, mouvement ou paire de mouvements représentés dans les schémas précédents inclue un facteur distance variable.

### 5.3. La programmation en environnement CSound des paramètres déterminants de l'espace dans *Sideral*

#### 5.3.1. Programmation modulaire en environnement CSound

La programmation en environnement CSound permet, à l'aide de fonctions de contrôle de l'ordre d'exécution des opérations, de construire des instruments qui fonctionnent de façon modulaire, c'est-à-dire qu'à l'intérieur d'un même instrument existent des unités fonctionnelles autonomes, que nous appelons ici modules, et qui peuvent être associées selon des

configurations diverses, donnant ainsi à l'instrument une plus grande souplesse d'utilisation.

De cette façon, l'utilisation de ces fonctions de contrôle permet la construction de modules de traitement ou de synthèse plus ou moins élémentaires, ce qui engendre à la fois la flexibilisation de l'instrument et la simplification de l'orchestre (fichier *.orc*). Imaginons par exemple que nous ayons besoin d'utiliser des fichiers son avec des traitements distincts, au lieu de construire plusieurs instruments comprenant chacun un traitement spécifique. Nous pourrions alors utiliser des conditions et des fonctions de contrôle en incluant ainsi dans le même instrument tous les traitements souhaités. Certes, cela implique une complexification de l'instrument, mais également une simplification de l'orchestre puisque le nombre d'instruments se voit réduit et, dans la pratique, la gestion d'un orchestre avec moins d'instruments devient plus simple.

Dans le cas de *Sideral*, nous avons besoin d'une certaine pluralité de procédés de traitement spatial de fichiers son. Nous avons donc décidé d'utiliser la définition de conditions et la fonction de contrôle *kgoto* pour contrôler le fonctionnement interne d'un petit nombre d'instruments comprenant chacun l'ensemble des procédés de traitement spatial.

Nous avons donc construit des instruments qui fonctionnent de façon modulaire. Analysons un d'entre eux.

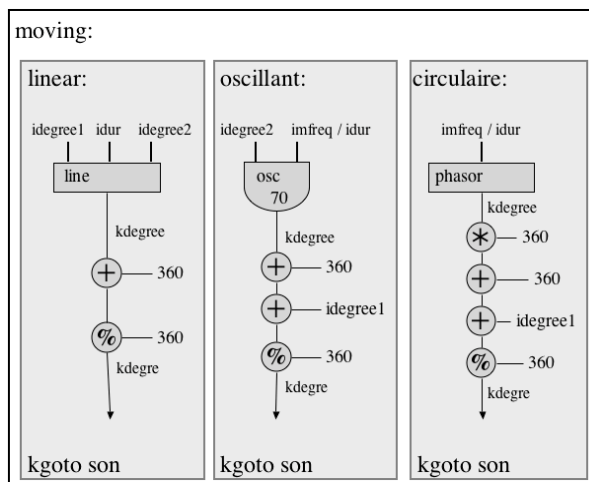
La construction d'un instrument CSound qui fonctionne de façon modulaire, implique la définition de conditions initiales permettant de choisir un type de traitement spatial pour chaque ligne de code de la « partition » (chaque « note » du fichier *.sco*). La fonction *kgoto*, avec le label qui lui est associé, permet de diriger ou rediriger l'enchaînement des opérations à l'intérieur de l'instrument. Les instruments ainsi construits sont en quelque sorte modulaires, puisque le résultat de chaque opération peut être envoyé vers l'un ou l'autre des blocs de fonctions qui exécutera l'opération suivante et ainsi de suite.

#### 5.3.2. Programmation des mouvements circulaires et oscillants en utilisant l'opcode *locsig*

Comme nous venons de voir, l'opcode *locsig* reçoit, en plus de l'entrée son, qui est à spatialiser, des valeurs de degré, distance et réverbération. Dans la programmation de *Sideral*, nous avons défini des sortes de modules qui permettent une plus grande flexibilité dans le traitement de l'espace à l'intérieur d'un même instrument. Nous avons élaboré trois méthodes différentes pour définir les mouvements ou localisations choisies en utilisant l'opcode *locsig*. Nous avons construit trois modules que nous appelons *moving*, *son* et *localiser*, à l'intérieur desquels se trouvent des unités fonctionnelles autonomes, un genre de sous-modules, qui définissent spécifiquement chacune des opérations à exécuter nécessaires à la définition de chaque procédé de traitement spatial.

Dans la pratique ces éléments sont programmés de la façon suivante :

Un ensemble de conditions, définies dans l'initialisation de l'instrument, permet le choix entre des modules utilisant l'opcode *locsig* ou *space*. Un premier module – *moving* – visant à l'utilisation de l'opcode *locsig*, permet de choisir, à l'aide de conditions qui y sont déterminées, entre un traitement spatial linéaire, circulaire ou oscillant.



moving:

```
idist1 = p10
idist2 = p11
imfreq = p18
```

```
imeth = p19; meth=1 degré final
; meth=2 déviation
; m=3 rotation
```

```
idegree1 = p12
idegree2 = (imeth == 3 ? 1 / p13 : p13)
```

```
if (imeth == 1 ) kgoto linear
if (imeth == 2 ) kgoto oscillant
if (imeth == 3 ) kgoto circulaire
```

linear:

```
kdegree line idegree1, idur, idegree2
kdegre = (360+ kdegree)%360
kgoto son
```

oscillant:

```
kdegree oscil idegree2, imfreq/idur, 70
kdegre = (360+ kdegree+ idegree1)%360
kgoto son
```

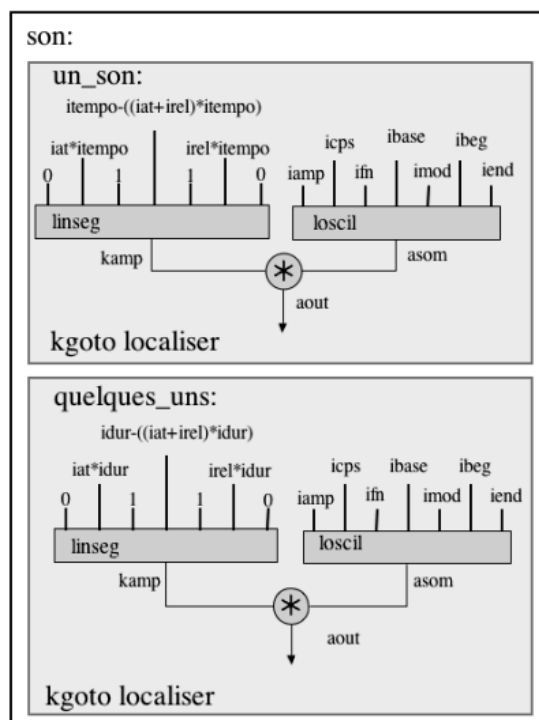
circulaire:

```
kline line idegree2, idur, idegree1
kdegree phasor imfreq/idur
kdegre = (360+ (kdegree*360)+ idegree1)%360
kgoto son
```

Figure 6. Schéma et programmation-texte du module *moving* avec ses unités fonctionnelles (sous-modules) qui permettent de basculer entre un traitement linéaire, oscillatoire ou circulaire des sons dans l'espace virtuel.

Ainsi, dans le premier cas (*linear*), une ligne définira le déplacement en degrés du son d'un point à un autre d'un cercle ; dans le deuxième cas (*oscillant*), l'oscillateur, qui utilise une table d'onde *hanning*, permettra de paramétrer des mouvements oscillatoires, à une vitesse variable, entre deux points ; dans le troisième cas (*circulaire*), le *phasor* définira des mouvements circulaires simples ou multiples.

Les mouvements ainsi définis sont envoyés vers le module *son*. Ce module permet de déterminer si pendant la durée de la « note » (définie dans le fichier « partition ») le mouvement défini est appliqué à un seul son (une lecture unique du fichier son) ou si au contraire il est appliqué à une lecture multiple du fichier son (le mouvement est appliqué par exemple à un enchaînement des plusieurs lectures du même fichier).



son:

```
if (imod == 0) kgoto un_son
if (imod != 0) kgoto quelques_uns
```

un\_son:

```
kamp linseg 0, iat*itempo, 1, itempo-
((iat+irel)*itempo), 1, irel*itempo, 0
asom loscil iamp, icps, ifn, ibase, imod, ibeg, iend
aout = asom*kamp
kgoto localiser
```

quelques\_uns:

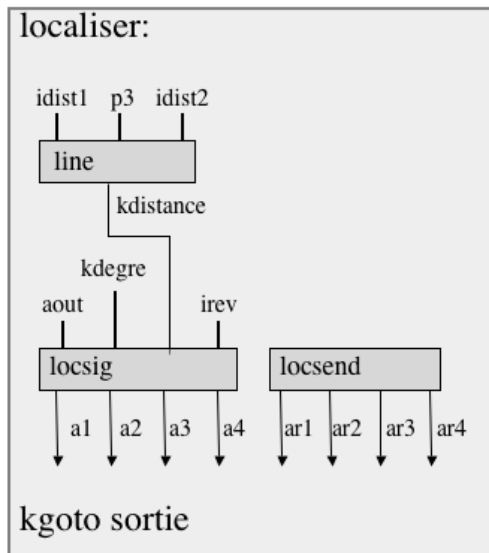
```
kamp linseg 0, iat*idur, 1, idur-
((iat+irel)*idur), 1, irel*idur, 0
asom loscil iamp, icps, ifn, ibase, imod, ibeg, iend
aout = asom*kamp
kgoto localiser
```

Figure 7. Schéma et programmation-texte du module *son* ainsi que de ses unités fonctionnelles autonomes (sous-modules).



Après la définition de ces éléments, le son est envoyé vers le module *localiser*, qui permet à son tour la définition d'une distance, la variabilité de cette distance étant programmée à l'aide d'une ligne.

Dans ce module *localiser*, nous observons également la présence de l'opcode *locsend* permettant de calculer la quantité de son à envoyer vers l'instrument de réverbération. Ce calcul dépend des coordonnées de localisation (angle et distance) définis dans les modules *moving* et *son*.



**localiser:**

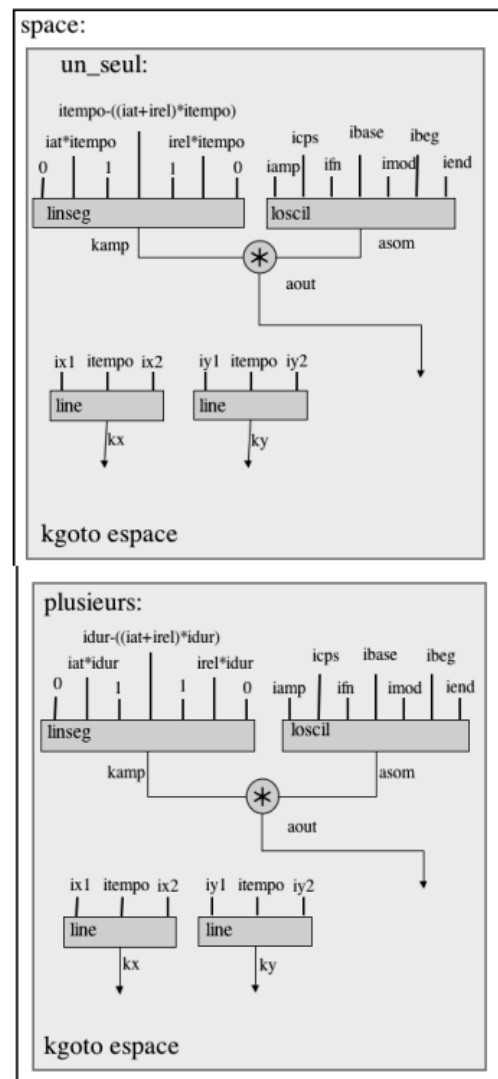
```
kdistance line idist1,p3,idist2
a1,a2,a3,a4 locsig aout,kdegre,kdistance,irev
ar1,ar2,ar3,ar4 locsend
kgoto sortie
```

**Figure 8.** Schéma et programmation-texte du module *localiser*.

### 5.3.3. Programmation de localisations et mouvements linéaires en utilisant l'opcode *space*

D'un fonctionnement très semblable à celui du module *moving*, le module *space* comporte des conditions rendant possible la définition du nombre de fois qu'un fichier son sera lu pendant la durée de la « note » définie dans le fichier « partition ».

Ce module permet également de déterminer l'évolution des coordonnées spatiales du son, admettant de ce fait tantôt des localisations fixes des sons par la détermination de coordonnées *xy* non-évolutives, tantôt des mouvements linéaires entre des mêmes coordonnées.



**space:**

```
if (imod == 0) kgoto un_seul
if (imod != 0) kgoto plusieurs
```

**un\_seul:**

```
kamp linseg 0,iat*idur,1,idur-
((iat+irel)*idur),1,irel*idur,0
;kamp linseg 0,iat*itempo,1,itempo-
((iat+irel)*itempo),1,irel*itempo,0
asom loscil iamp,icps,ifn,ibase,imod,ibeg,iend
aout = asom*kamp
kx line ix1,itempo,ix2
ky line iy1,itempo,iy2
kgoto espace
```

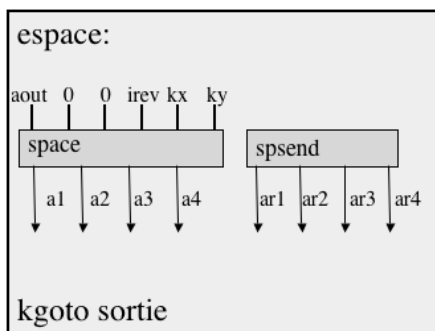
**plusieurs:**

```
kamp linseg 0,iat*idur,1,idur-
((iat+irel)*idur),1,irel*idur,0
;kamp linseg 0,iat*itempo,1,itempo-
((iat+irel)*itempo),1,irel*itempo,0
asom loscil iamp,icps,ifn,ibase,imod,ibeg,iend
aout = asom*kamp
kx line ix1,idur,ix2
ky line iy1,idur,iy2
kgoto espace
```

**Figure 9.** Schéma et programmation-texte du module *space* ainsi que des unités fonctionnelles (sous-modules) autonomes.



Le module *espace* qui suit, calcule à l'aide des coordonnées *xy*, dont l'évolution a été déterminée dans le module antérieur (*space*), la position exacte ou le mouvement précis à insuffler au son ; c'est-à-dire qu'il calculera à la fois la quantité de son à envoyer à chaque haut-parleur ainsi que la quantité devant être envoyée à l'instrument de réverbération, de façon à simuler la situation spatiale spécifique de ce son-là.

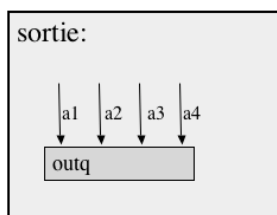


**espace:**

```
a1,a2,a3,a4   space   aout, 0,0,irev, kx, ky
ar1,ar2,ar3,ar4 spsend
kgoto sortie
```

**Figure 10.** Schéma et programmation-texte du module *espace*.

À la fin de l'instrument se présente la sortie quadriphonique (module *sortie*), vers laquelle sont acheminés les sons traités et porteurs de spécificités spatiales, arrivant du module *localiser* ou du module *espace*, donc d'un processus de traitement spatial par l'usage de l'opcode *locsig* ou par l'usage de l'opcode *space*.



**sortie:**

```
outq a1,a2,a3,a4
```

**Figure 11.** Schéma et programmation-texte du module *sortie*.

## 6. CONCLUSION

Nous avons présenté ici, à la lumière de l'œuvre *Sideral*, à la fois des aspects pertinents de la perception auditive de l'espace et des méthodes de programmation en environnement CSound qui permettent une grande souplesse et une rigueur dans la programmation de la

simulation de localisations ou mouvements sonores dans un espace physique défini par un ensemble de haut-parleurs.

Cette approche permet de conclure que, malgré sa complexité, le logiciel CSound offre la possibilité d'effectuer un travail compositionnel important sur l'élément espace, ce qui est difficilement réalisable par d'autres moyens.

## 7. RÉFÉRENCES

- [1] BISSELL, Roger E. « Music and Perceptual Cognition VI: The Analogy to Spatial Location and Motion » *The Journal of Ayn Rand Studies*, 1999.
- [2] BONNET, Claude. GHIGLIONE, Rodolphe. RICHARD, Jean-François. (dir.). *Traité de psychologie cognitive : Perception, action, langage* ; ed. Dunod, Paris : 2003. (1<sup>er</sup> édition : Paris, Bordas, 1989).
- [3] BOTTE, Marie-Claire. CANÉVET, Georges. DEMANY, Laurent. SORIN, Christel. *Psychoacoustique et perception auditive*. Paris : Éditions INSERN, 1988.
- [4] BREGMAN, Albert S. *Auditory Scene Analysis : The Perceptual organization of sound*. Cambridge : MIT Press, 1991.
- [5] CANÉVET, Georges. « La localisation auditive des sons dans l'espace » In : GENEVOIS, Hugues. ORLAREY, Yann. *Le son et l'espace*. Lyon : GRAME / Aléas Éditeur, 1998. Collection Musique et Sciences. Sous la direction de Yann ORLAREY. P. 15 – 32.
- [6] DODGE, Charler. JERCE, Thomas. *Computer music, Synthesis, Composition, and Performance*. Schirmer Books, New York, 1985
- [7] DUFOURT, Hugues. « L'espace sonore, "paradigme" de la musique de la seconde moitié du XXème siècle ». In : *L'Espace : Musique / Philosophie*. Textes réunis et présentés par Jean-Marc CHOUVEL, Makis SOLOMOS. Paris : L'Harmattan, 1998. Collection Musique et Philosophie. P. 178-186.
- [8] JULLIEN, Jean-Pascal. WARUSFEL, Olivier. « Technologies et perception auditive de l'espace ». *Les Cahiers de l'IRCAM, n° 5*, 1<sup>er</sup> trimestre de 1994 – *Espaces*. Éditions Ircam – Centre Georges-Pompidou. 1994. Collection Musique et Recherche sous la direction de Laurent BAYLE. P. 65 – 94

- [9] ROADS, Curtis. « Composition et diffusion : quelques observations ». *Académie Bourges, Actes III 1997 – Composition / Diffusion en Musique Électroacoustique*. Bourges : Éditions Mnémosyne. 1998, p. 158 – 160.
- [10] ROADS, Curtis. *The Computer Music Tutorial*. MIT Press. Cambridge. 1996.
- [11] ROADS, Curtis. *Audionumérique*. Dunod. Paris : 2001. (Version française de Jean de Reydellet)
- [12] ROADS, Curtis. *Microsound*. MIT Press. Cambridge. 2001.
- [13] TRUAX, Barry. « Composition et diffusion : espace du son dans l'espace ». *Académie Bourges, Actes III 1997 – Composition / Diffusion en Musique Électroacoustique*. Bourges : Éditions Mnémosyne. 1998. P. 177 – 181,
- [14] VAGGIONE, Horacio. « L'espace composable. Sur quelques catégories opératoires dans la musique électroacoustique ». In : *L'Espace : Musique / Philosophie*. Textes réunis et présentés par Jean-Marc CHOUVEL, Makis SOLOMOS. Paris : L'Harmattan, 1998. Collection Musique et Philosophie. P. 154 – 166.
- [15] VERFAILLE, Vincent. « Utilisation d'espaces perceptifs pour la synthèse et la transformation sonore ». In : *Espaces Sonores – actes de recherche*. Sous la direction d'Anne SEDES. CICM / Éditions Musicales Transatlantiques. P. 39 – 54.

## ANNEXE:

```
instr 2

idur = p3
iamp = p4
icps = cpspch(p5)
ifn = p6
ibase = cpspch(8.00)
imod = p7      ;;loop mode: 0=no loop
                  ; 1=normal
                  ; 2=dans les deux sens

ibeg = p8      ;; point de commencement
iend = p9      ;; point d'arrêt

ix1=p10
iy1=p11
ix2=p12
iy2=p13

irev = p14
iat = p15
irel= p16
iref = p17
ivel = icps/ibase
itempo = iref/ivel

if ((ix1 > 1) || (ix2 > 1) || (iy1 > 1) ||
    (iy2 > 1)) kgoto moving
if ((ix1 <= 1) || (ix2 <= 1) || (iy1 <=1) ||
    (iy2 <=1)) kgoto space

moving:

idist1 = p10
idist2 = p11
imfreq = p18

imeth = p19;meth=1 degré final
          ; meth=2 déviation
          ; m=3 rotation

idegree1 = p12
idegree2 = (imeth == 3 ? 1 / p13 : p13)

if (imeth == 1 ) kgoto linear
if (imeth == 2 ) kgoto oscillant
if (imeth == 3 ) kgoto circulaire

linear:

kdegree line idegree1, idur,idegree2
kdegree = (360+ kdegree)%360
kgoto son

oscillant:

kdegree oscil idegree2, imfreq/idur,70
kdegree = (360+ kdegree+ idegree1)%360
kgoto son

circulaire:

kline line idegree2,idur, idegree1
kdegree phasor imfreq/idur
kdegree = (360+ (kdegree*360)+ idegree1)%360
kgoto son
```

```

son:

if (imod == 0) kgoto un_son
if (imod != 0) kgoto quelques_uns

un_son:

kamp linseg 0,iat*itempo,1,itempo-((iat+irel)*itempo),1,irel*itempo,0
asom loscil iamp,icps,ifn,ibase,imod,ibeg,iend
aout = asom*kamp
      kgoto localiser

quelques_uns:

kamp linseg 0,iat*idur,1,idur-((iat+irel)*idur),1,irel*idur,0
asom loscil iamp,icps,ifn,ibase,imod,ibeg,iend
aout = asom*kamp
      kgoto localiser

localiser:

kdistance line idist1,p3,idist2
a1,a2,a3,a4 locsig aout,kdegre,kdistance,irev

ar1,ar2,ar3,ar4 locsend
      kgoto sortie

space:

if (imod == 0) kgoto un_seul
if (imod != 0) kgoto plusieurs

un_seul:

kamp linseg 0,iat*idur,1,idur-((iat+irel)*idur),1,irel*idur,0
asom loscil iamp,icps,ifn,ibase,imod,ibeg,iend
aout = asom*kamp
kx   line  ix1,itempo,ix2
ky   line  iy1,itempo,iy2
      kgoto espace

plusieurs:

kamp linseg 0,iat*idur,1,idur-((iat+irel)*idur),1,irel*idur,0
asom loscil iamp,icps,ifn,ibase,imod,ibeg,iend
aout = asom*kamp
kx   line  ix1,idur,ix2
ky   line  iy1,idur,iy2
      kgoto espace

espace:

a1,a2,a3,a4 space aout, 0,0,irev, kx, ky
ar1,ar2,ar3,ar4 spsend
      kgoto sortie

sortie:

outq a1,a2,a3,a4

garvb5 = garvb5 + ar1
garvb6 = garvb6 + ar2
garvb7 = garvb7 + ar3
garvb8 = garvb8 + ar4

endin

```