

# An Analysis of the Compositional Techniques in John Chowning's *Stria*

This article describes an analytic study of the process used by John Chowning for the composition of *Stria*. This article is intended to complete the description of the compositional process given in a previous work (Meneghini 2003), largely restated by Bossis in a subsequent paper (Bossis 2005). *Stria* was composed in 1977 and was fully generated by means of computer algorithms (Chowning 1977c) and the corresponding input files (Chowning 1977a, b): all the parameters of the sounds generated in *Stria* are determined and calculated by these algorithms on the basis of specific mathematical rules and of numerical parameters chosen by Mr. Chowning as input for the programs. For this reason, an accurate analysis of these algorithms is fundamental for a complete comprehension of the compositional process.

This analysis has three degrees of interest: at first, it enables a study of the composer's intentions regarding the piece's characteristics. Secondly, it complements the comprehension of *Stria* obtainable from listening with analytic data on the properties of the sounds. Finally, it provides information on the characteristics of the piece that can be used for the re-writing of the algorithms and for the analysis and re-synthesis of the piece (Baudouin 2007; Dahan 2007).

The work described in this article began in 2002, and it started with the study of the programming language (SAIL) used for the generation of the algorithms (Smith 1976; see also [www.xidak.com/mainsail/documentation\\_set\\_1630\\_html/docset.html](http://www.xidak.com/mainsail/documentation_set_1630_html/docset.html)), which was necessary for the understanding of the programs. The algorithms (Chowning 1977c) have been then carefully analyzed, line by line, to achieve a complete understanding on how the parameters of each sound generated in *Stria* are determined. Personal communication with the composer was indispensable for the interpretation of some of the procedures used, and for the understanding of Mr. Chowning's intentions related to programming choices.

## General Properties

This section explains the general properties that characterize *Stria*. The piece was fully composed by means of a computer program: The properties of each sound were generated by specific subroutines, starting from the input data chosen by the composer, and taking into account a set of mathematical relationships which will be explained shortly. Most of the information described here was extracted from the source code of the programs used by Mr. Chowning himself.

Starting from some basic definitions about the Golden Mean, this section gives a description of the pitch space and spectrum division, of the characteristics of the sounds, and of the temporal structure of the piece.

## The Golden Mean

*Stria* is based on the mathematical properties of the Golden Mean: The basic properties of this important ratio are summarized in this section. Consider a segment of length one, and look for a subset of length  $x$  such as the ratio between the segment's overall length and  $x$  is equal to the ratio between  $x$  and the remaining part of the segment itself. To do this, we consider the equality

$$\frac{1}{x} = \frac{x}{1-x} \quad (1)$$

Solving this equation for  $x$ , we find that

$$x = \frac{1}{2} \left( -1 + \sqrt{5} \right) \cong 0.618 \quad (2)$$

We can then extend this result to the continuous proportion

$$\frac{1-x}{x} = \frac{x}{1} = \frac{1}{1/x} = \dots \quad (3)$$

which numerically corresponds to

$$\frac{0.382}{0.618} \cong \frac{0.618}{1} \cong \frac{1}{1.618} \quad (4)$$

In ancient and current times, the important ratio we have obtained this way (and its reciprocal  $\Phi$ , usually referred to as “Golden Mean” or “Golden Section”) has been considered a rule of physical perfection. Many authors have indicated that it is easily recognizable in many human works (e.g., in architecture) and in nature as well (e.g., Runion 1990; Markowsky 1992).

In music, the Golden Section represents (to a good approximation) the interval of a minor sixth in Western notation. Traditionally, the minor sixth was expressed as the ratio  $8/5 = 1.6$ , which is close to the Golden Mean. In twelve-tone equal temperament, the minor sixth is eight semitones or

$$2^{8/12} \cong 1.59$$

Another important property of the Golden Mean is related to the Fibonacci series: Each of the terms of this series, starting with  $\{0, 1, 2\}$ , is the sum of the two immediately preceding terms. It can be proved that the ratio between two consecutive terms of this succession quickly approaches to the Golden Section. From this, we can easily derive that the powers of  $\Phi \cong 1.618$  are ordered in accordance with the Fibonacci series:

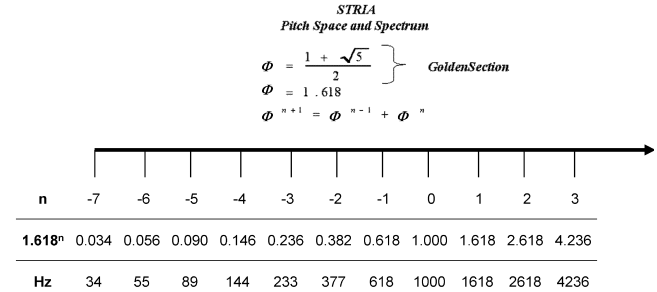
$$\Phi^n = \Phi^{n-1} + \Phi^{n-2} \quad (5)$$

These properties are very important in reading *Stria* and must be taken into account in the following analysis.

## Pitch Space and Spectrum

After a long series of experiments on frequency modulation (FM) synthesis, Mr. Chowning tried to discover an inharmonic ratio to re-define the concept of the octave. After many tests (executed before programming) and after becoming fascinated by the sound of FM carrier-to-modulator ratios  $c:m = \Phi^n : \Phi^m$  (where  $n$  and  $m$  are integer powers; Chowning 2003), he re-defined the concept of octave (usually based on the ratio 1:2), using the ratio  $1:\Phi \cong 1:1.618$ . The use of such FM ratios allowed him to obtain overtones, some of which were exact powers of that ratio. Each pseudo-octave generated was then equally divided into nine tones by the factor

Figure 1. Pitch space used in *Stria*. The fundamentals of the pseudo-octaves are listed in the diagram. The basic properties of the Golden Mean are also summarized.



**Table 1. Basic Powers of  $\Phi$  and Their Relation with Linear Combinations of  $\Phi$**

Power of $\Phi$	Linear combination $a + b\Phi$
$0.056 = \Phi^{-6}$	$13 - 8\Phi$
$0.090 = \Phi^{-5}$	$5\Phi - 8$
$0.146 = \Phi^{-4}$	$5 - 3\Phi$
$0.236 = \Phi^{-3}$	$2\Phi - 3$
$0.382 = \Phi^{-2}$	$2 - \Phi$
$0.618 = \Phi^{-1}$	$\Phi - 1$
$1 = \Phi^0$	1
$1.618 = \Phi^1$	$\Phi$
$2.618 = \Phi^2$	$1 + \Phi$
$4.236 = \Phi^3$	$1 + 2\Phi$

$$\Phi^{\frac{k}{9}}, k = 0 \dots 9 \quad (6)$$

An eighteen-tone division was also available, generating a series of “semitones.” The pseudo-octaves used in *Stria* are generated around the central frequency  $f = 1,000$  Hz, and the fundamentals of each octave are expressed by  $\dots\Phi^{-3}f, \Phi^{-2}f, \Phi^{-1}f, f, \Phi f, \Phi^2f, \dots$

A representation of the pitch space used in *Stria* is given in Figure 1. As seen in this figure, the fundamentals of each octave are related to powers of  $\Phi$ . However, the advantage of using the Golden Mean for the pseudo-octave definition is that the frequencies are related not only by exponents but also by a linear combination. In fact, because  $\Phi$  is the limit of the ratio between two successive terms of a Fibonacci series, the sum of two following powers of  $\Phi$  is a power of  $\Phi$ . This is summarized in Table 1.

The spectral components obtained as sums of powers of  $\Phi$  can be expressed by linear combinations  $a + b\Phi$ . Mr. Chowning decided to use this property defining a carrier-to-modulator ratio for

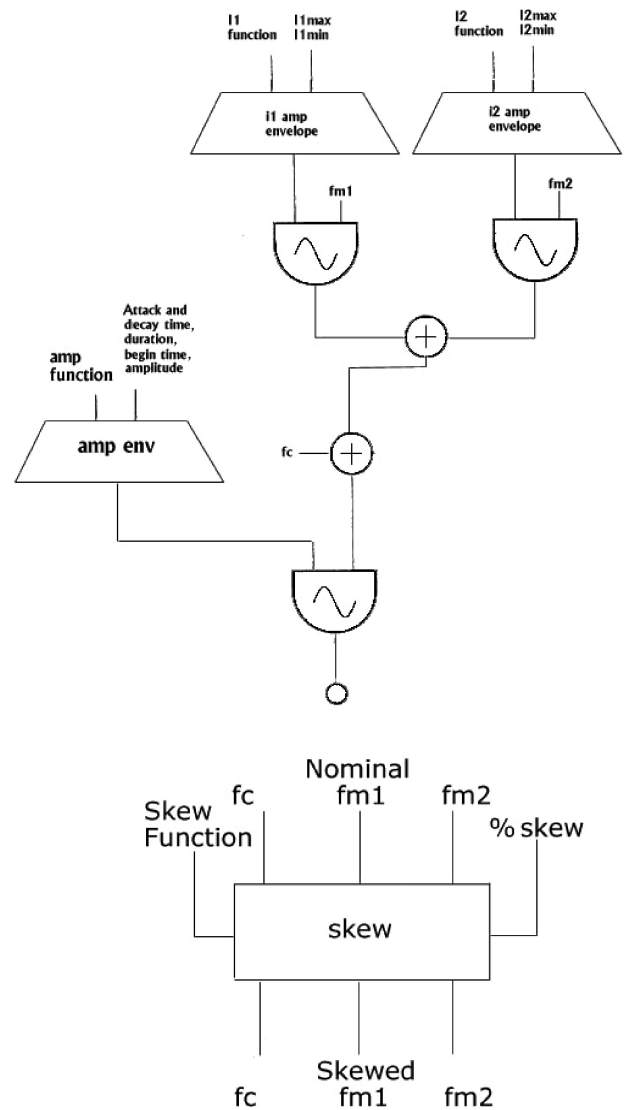
Figure 2. Instrument used in *Stria*.

the FM synthesis equal to powers of  $\Phi$ . In this way, the FM components generated were sums or differences of powers of  $\Phi$ , which were also in a linear relationship  $a + b\Phi$  with  $\Phi$ . With this efficient mechanism, the whole pitch space was ordered in a way that there was no component in discordance with the Golden Ratio. The composer used eight pseudo-octaves (see Figure 1): three above and five below the central frequency ( $f = 1,000$  Hz); all these pseudo-octaves were used in the composition.

### The Sounds

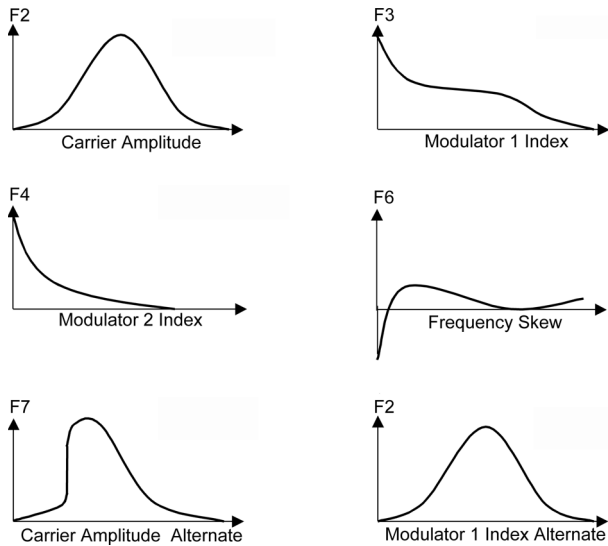
The properties of each of the sounds played in *Stria* were calculated by the source algorithms, starting from a set of global input data. In particular, each sound was defined by 30 parameters (begin time, attack duration, carrier frequency, etc.). The basic FM scheme was a single carrier with two parallel modulators: All the oscillators were sine functions. Furthermore, each sound was shaped in time: Mr. Chowning applied independent envelope generators to all the oscillators to have an accurate control of the spectral content of the sounds. A light deviation (skew) was superimposed to the frequencies of both the carrier and the modulators in a proportional way to obtain a “liveliness.” The two modulators allowed the composer to increase the spectral density without using large indices. Large modulation indexes would have reduced the contribution of the carrier in the modulated sound by reducing the zero-order Bessel function, which Mr. Chowning did not want. The instrument (i.e., synthesis patch) generating the sounds is represented in Figure 2. The envelopes used for the oscillators have the shape shown in Figure 3.

There were two possibilities for the amplitude envelopes: a “normal” and an alternate. In the normal case, the sound started as modulated by the first and the second modulator (F3 and F4 in Figure 3), then the second modulating index (F4) decreased and the sound became modulated only by the first modulator and finished as only the carrier. (The F2 carrier amplitude envelope was used in this case.) The alternative case was used to produce a sort of “sssh-Boom” effect (as Mr. Chowning called it) during the



climax of the piece: this effect was obtained by a step variation of the carrier amplitude (envelope F7 in Figure 3). The description of the envelopes given in Figure 3 is slightly different with respect to what described in Meneghini (2003): The figure reported in the previous paper was based on a preliminary drawing prepared by John Chowning in 2002. The envelopes presented in this article were re-drawn by the composer on 30 November 2006 on the basis of the original drawings dated 8 September 1977; these

Figure 3. Amplitude envelopes used in *Stria*.



are the same that have been used for the rewriting of the piece (Baudouin 2007).

The skew function was defined by a small deviation at the attack of the sound that became even smaller in a short time. It is difficult to hear this variation on a single sound, but it is easily recognizable in the superposition of many musical elements as a kind of beating between components. The maximum amplitude of the skew function was determined starting from the frequency of the tone to play, by means of a low-pass function, to obtain a sharp deviation at low frequencies and a small deviation at high frequencies.

The human ear is more sensitive to frequency variations at low frequencies than high frequencies. This justifies the shape of the function defining the skew percent as a function of frequency. There is an additional justification for this low-pass function that is related to the overall structure of the piece, explained in the following section.

For each sound, a set of three spatialization parameters was defined: the reverb to be applied to the sound, the apparent angle of the source, and the apparent distance of the source. The last parameter was defined in accordance with the theory explained by Mr. Chowning (1971 and 1999). In both these papers, the composer explained that the perception of distance can be given by the ratio be-

tween the direct sound intensity (varying with the distance) and the reverberated sound one (fixed with the distance).

Spatial control used in *Stria* was not intended to be precise: The sounds resulting using these parameters were similar to the ones that would be obtained in a reverberant cathedral: amorphous, big, and undefined (Chowning 2003). Furthermore, no Doppler effect was used.

### The Temporal Structure

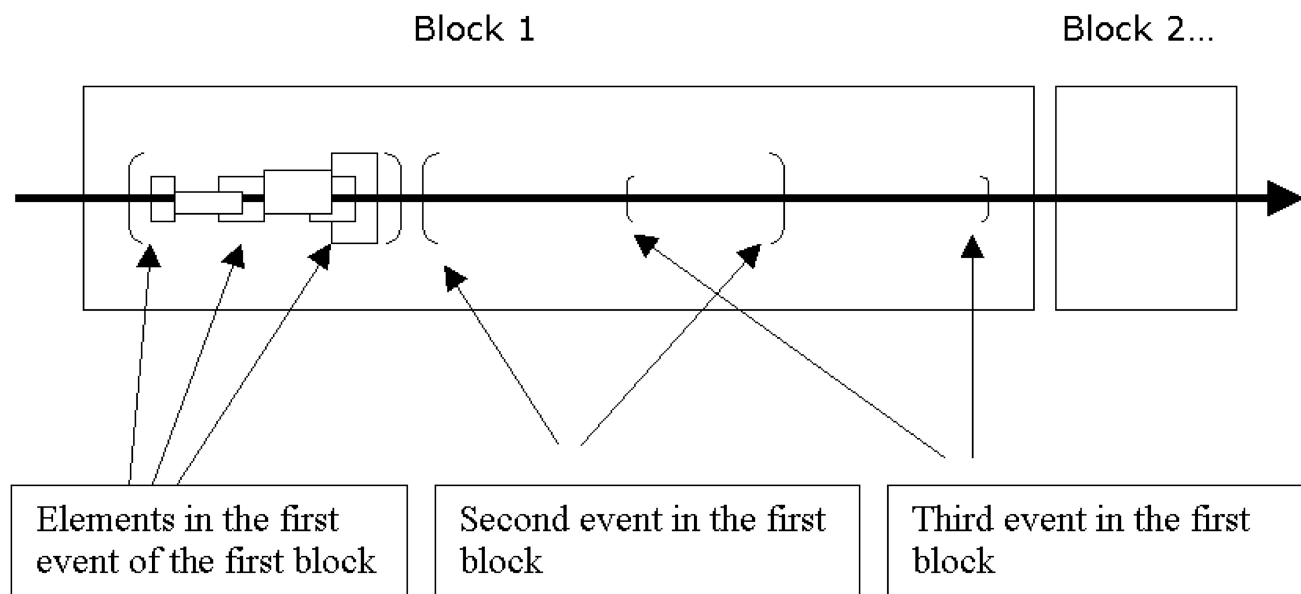
The scores of *Stria* were generated by the algorithms starting from several sets of data, which the composer chose as input to the programs. The piece can be considered composed in terms of both a microstructure and a macrostructure. The whole composition is divided in 6 blocks (or sections): Each block corresponds to a different input session during which Mr. Chowning inserted a sequence of data as program parameters. The output generated by the algorithms starting from each of these input sessions was saved into three different score files.

Each block is composed of few events: Each event is defined by a set of input data and is composed of a great number of elements (or single sounds) generated by the algorithms starting from the input values. Figure 4 illustrates the temporal structure of the piece.

The arrow represents the temporal evolution of *Stria*, the large rectangles are the blocks, and events are shown between brackets. The first event also shows the elements composing it (small rectangles). We can consider the elements as the elementary unit of which *Stria* is constituted. Each element is heard as a single sound or as a multiplicity of related partials depending upon the pitch height and the succession and the superposition of these sounds is the whole piece. This is the macrostructure of *Stria*: each element is defined by some microstructural parameters, defining its characteristics, as will be clear in a following section. Some of these parameters are generated according to the Golden Mean.

During the composition of the piece, Mr. Chowning used recursive algorithms to generate further

Figure 4. Temporal structure of *Stria*.



sounds ("child elements") superimposed onto the original ones ("parent elements"); this will be discussed in a succeeding section. The whole piece lasts 15'46". In the first part of *Stria*, the intensity increases, and after 9' (a number that approximately stands in the Golden Ratio with respect to the total length), there is a climax, followed by a quasi-silence moment, after which the intensity grows again. The organization of the sound events in the time-frequency space is opposite that of tradition: In most music, low-pitched events are longer in duration than the high-pitched events. Here, instead, the longer events have higher pitch, and vice versa (Dodge and Jerse 1985). Furthermore, the attack and decay time of each sound is determined by its frequency (e.g., a high-pitch sound has a slow attack). The pitch of the sounds here decreases towards the climax and increases after this moment. These points, in addition to the low-pass function applied to the frequency skew noted previously, have an importance related to the overall structure of the piece. Mr. Chowning explains:

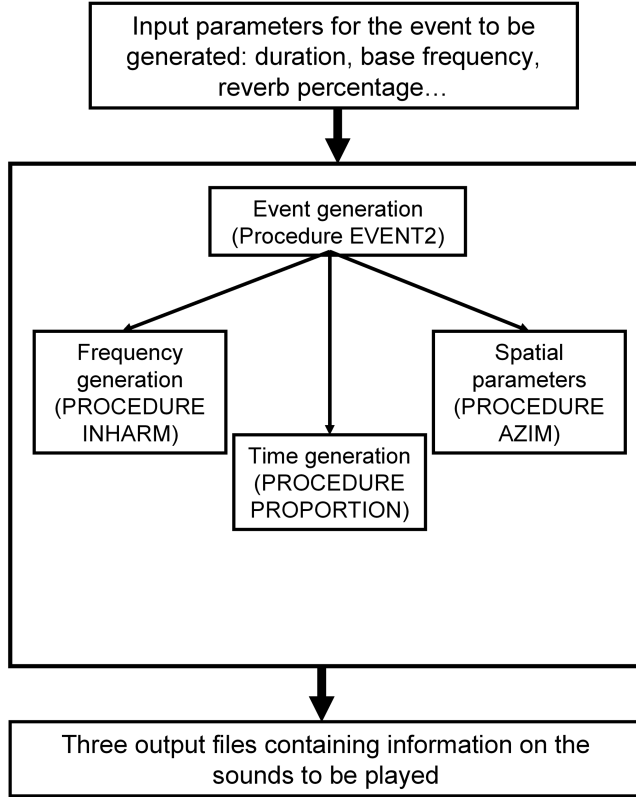
In the beginning, each individual partial or line (stria) of the high, extended sounds is clearly heard. The piece slowly unfolds with the pro-

gression toward lower pitches and an increase in the bandwidth of the pitch space. As the texture becomes increasingly dense, the individual partials within an element at the lower pitches tend to fuse or cohere because of the shorter attack times and increase in frequency skew, a perceptual phenomenon related to the Gestalt principle of "common fate." Thus, where the line at the beginning of the piece is the partial of the element, the line at the climax becomes the element of the block. The process reverses from the climax to the end (Chowning 2007).

## The Program

The program that generated the scores starting from the input data was written in SAIL (Stanford Artificial Intelligence Language; Smith 1976), a language similar to ALGOL and PASCAL. It consists of a few procedures (see Figure 5) called by a main program; sounds composing the events were generated by the procedure `EVENT2`, receiving data from input, and calling other service routines to generate the frequencies (`INHARM`), the times (`PROPORTION`), and the spatialization parameters (`AZIM`) of the sounds

Figure 5. Schematic of structure of the SAIL program used for the composition of Stria.



and to write the output files (WRITE). Each call of the program defines one block composed of a few events; each of these events is defined by a set of input parameters. The output of the program consists of three files, one of which, the score file (.SCR), contains 30 parameters for each sound played. The other files (.MEM and .REP) represent a memory of the input data used by Mr. Chowning for the composition of the piece.

### Frequency Generation

Frequency generation is managed by the INHARM procedure, called by EVENT2, when generating each sound. Each sound created in one event has a different frequency value in accordance with the variation of a variable (*num*) at every call of INHARM. It is the variation of this parameter that creates the pitch line of the event. In EVENT2, the frequency

of each sound (i.e., element) is generated by the expression

$$f = fff \cdot freq \cdot k \quad (7)$$

where *freq* is the base or reference frequency of the event, that is, the frequency of the fundamental note of the pseudo-octave on which the whole event is constructed (constant for each sound in the event); *fff* is the coefficient (scale frequency) that determines the note played by the current sound in the event by means of the product  $fff \cdot freq$ , varying from element to element in the event (at each call of INHARM); *k* is a coefficient used to calculate the frequency to be played on each oscillator by the product  $fff \cdot freq \cdot k$ . For example, the carrier frequency is calculated from the note  $fff \cdot freq$  by the expression

$$c = fff \cdot freq \cdot f_c \quad (8)$$

Here, *fff* and *k* are multiples of  $\Phi$ , and *freq* is given by an expression like  $\Phi \cdot 1000$ . The variable *fff* varies from sound to sound at every call of the INHARM procedure.

For each event, a frequency space variable is defined by the expression

$$space = ratio^{power} \quad (9)$$

where *ratio* is equal to  $\Phi \cong 1.618$ , and *power* is a positive or negative integer. This variable represents the frequency space to be occupied by the event, namely, the dimension of the spectral space occupied by the whole event. The variable *power* is the number of pseudo-octaves (above or below *freq*, in accordance with its sign) used in the event. Each element (sound) will play a note in one of these pseudo-octaves that will be divided into 9 or 18 tones.

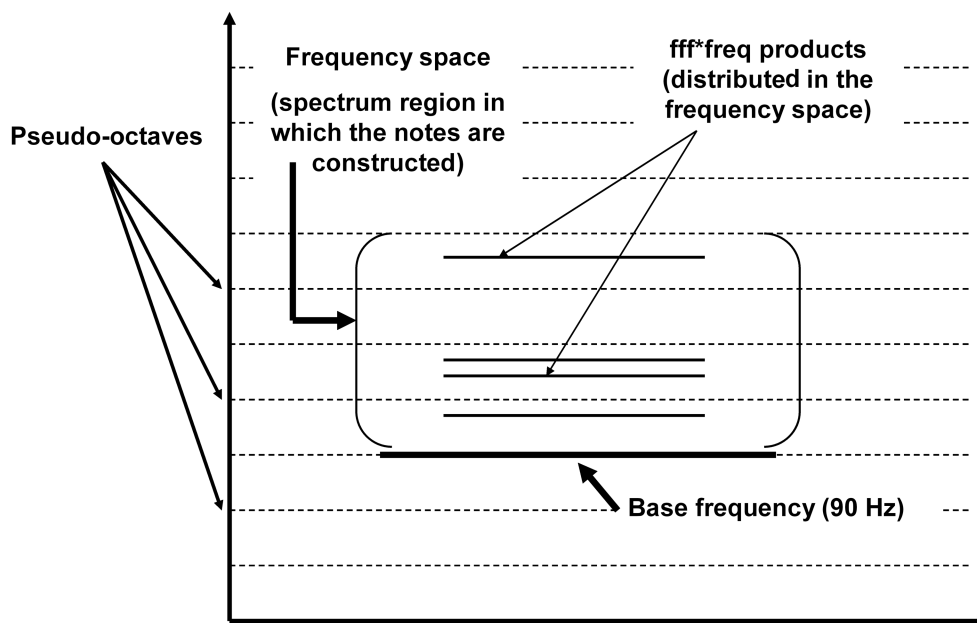
This division is done by means of variable *fff*, which defines the note to be played, and is calculated by the expression

$$fff = \left( ratio^{power} \right)^{num/divx} = ratio^{\frac{power \cdot num}{divx}} \quad (10)$$

Here, *num* is different for each element generated, and it is calculated by the service routine INHARM, called by EVENT2; *divx* is a variable that is equal to the number of divisions chosen for the frequency

Figure 6. Parameters used for the definition of each element (an example case in which  $\text{freq} = 90$  Hz,  $\text{power} = 4$ ). The spectral components related to the

$\text{fff} \cdot \text{freq}$  products are represented for four different values of  $\text{num}$ . Here,  $\text{freq}$  is the fundamental note of the indicated pseudo-octave. (See Equation 10.)



space (9 or 18 notes for pseudo-octaves). It is important to note that for  $|\text{power}| = 1$ , Mr. Chowning did not use the 18-tone division to avoid noisy spectra.

Figure 6 shows the relationships between the parameters used to define the frequencies of each element in *Stria*. The input data used by Mr. Chowning for the composition of the piece (Chowning 1977a, 1977b) determine the variation of the parameters  $\text{freq}$  and  $\text{power}$  during the piece. Therefore, using these data it is possible to analyze the frequency occupation of the events in *Stria*. The results of this analysis are summarized in Figure 7. As can be seen, the pitch of the events decreases towards the climax, and then it increases. This confirms the previous assertion that high-pitched events have longer durations whereas, on the other hand, lower-pitched events are shorter.

As stated above, the parameter  $\text{num}$  determines the frequency of each of the sounds generated in *Stria*. It is worth mentioning how  $\text{num}$  is determined by the algorithms: Because *Stria* is the succession of sounds with different values of  $\text{num}$ , it is  $\text{num}$  itself that defines the pitch line of the piece. The value of  $\text{num}$  is constructed from a table of 10 values: The elements of this table are read following specific rules so that a succession of values of  $\text{num}$

is generated. This succession, in the case of a 9-tone division of each pseudo-octave, is periodic with repetition period 40 (if not re-initialized), meaning that the melodic line would be repeated only every 40 elements. In *Stria*, there is no event with more than 40 (parent) elements; this means that the melodic line is not repetitive during an event. (For the elements generated by recursion, see herein; the base frequency is different from that of the parent, and no repetitive melodic line is possible, even though continuing the reading of the 40-period succession.) The generation of the values of  $\text{num}$  in the subsequent events can continue to follow the succession (creating the periodicity) or be re-initialized from the initial values. This choice (made by input parameters) could be used by Mr. Chowning to increase the pitch-line complexity of the piece. In the case of an 18-tone division of the frequency space occupied by the event, the table is read in a different way, generating a succession with period 20 of values of  $\text{num}$  in the range 0–18;  $\text{divx}$  is equal to 18, in this case. In this way, Mr. Chowning allowed some events to generate elements that play an interval analogous to a pseudo-octave semitone. It is important to note that the recursive sounds (child elements) can be constructed only on a 9-division

Figure 7. Representation of the temporal variation of the frequency occupation of the events in *Stria*.

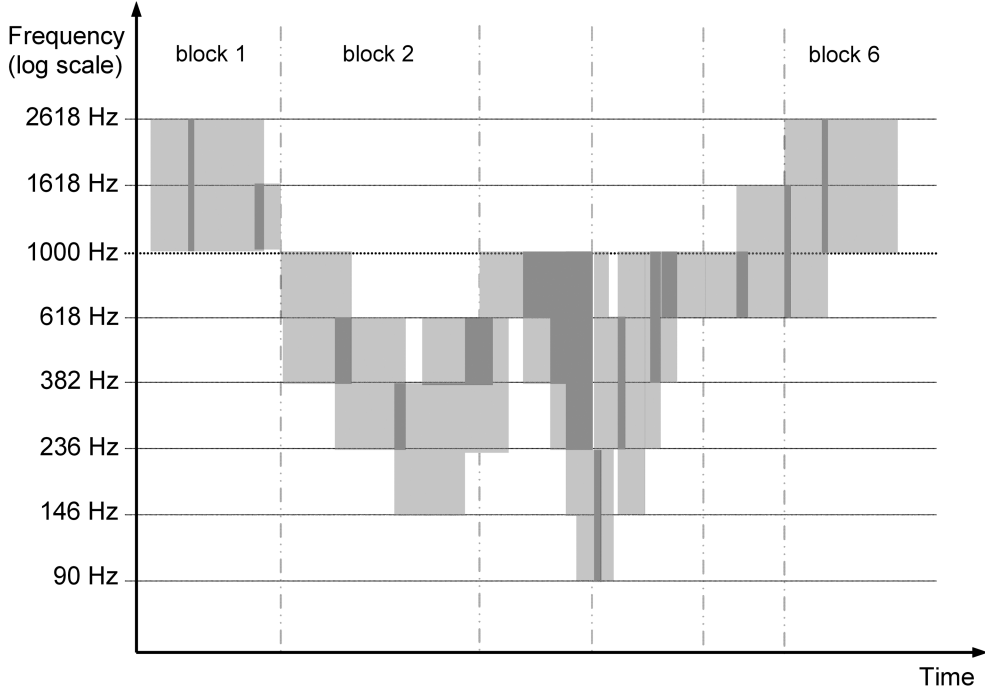
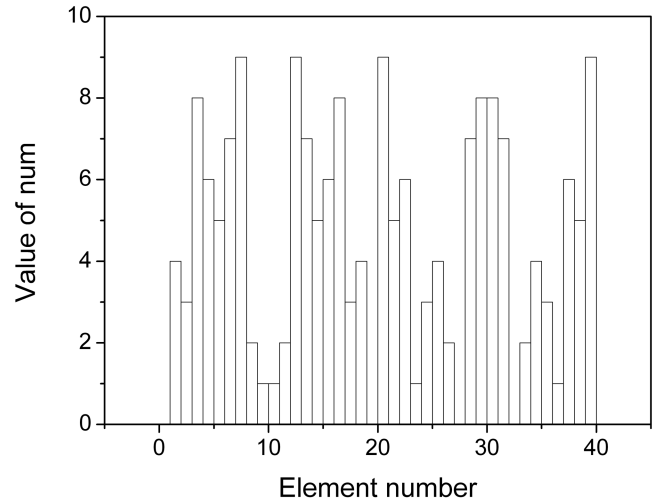


Figure 8. Pitch line (values of *num*) generated, starting from the default conditions by procedure *INHARM*. The values of *num* for 40 consecutive events are represented.



space, whereas the parent sounds can have *divx* equal to either 9 or 18. (Actually, the 9-note division of the pseudo-octave yields an interval close in size to the traditional semitone, so a *Stria* "semitone" is about the size of a traditional quarter-tone.)

Figure 8 represents a particular case of a melodic line (of values of *num*) generated starting by the default parameters of *INHARM* for 40 consecutive elements in an event for a 9-tone division of the frequency space.

From the values of *fff* (calculated from *num*) and *freq* for each element, the procedure *EVENT2* calculates the carrier frequency and the second modulator by the following:

$$c = fff \cdot freq \cdot f_c \quad (11a)$$

$$m_2 = fff \cdot freq \cdot f_{m2} \quad (11b)$$

where  $f_c$  and  $f_{m2}$  are the frequency coefficients already explained. The determination of the first modulator frequency is quite different. For this oscillator, the program could use an equation just like Equation 11b (which would create different components for all the elements in the event), but it also

had the option of maintaining the same first lower (or upper) side spectral line constant for all the elements in the event. In the latter case, the formula used (for a constant lower side) is

$$f_{m1} = fff \cdot freq \cdot \left[ -ratio \frac{(9 - num)^{(power - 1)}}{9} \cdot (f_c - f_{m1}) + f_c \right] \quad (12)$$

and remembering Equation 10, the first lower side spectral line is

$$f_{m1} = (f_c - f_{m1}) \cdot freq \cdot ratio\% \quad (13)$$

which is constant with *num*, that is, for each element in the event. An accurate reading of the parameter files used for the generation of *Stria* indicates that this option was never used by Mr. Chowning for the final version of the piece, even if it was included in the original program. However, the fact that the program contained such a possibility is an indication of the intensive research work carried out by the composer on the spectral properties of the sounds.

## Time Generation

Another important topic in the composition is time generation: Each instrument has time parameters, including start time, duration, and attack and decay times. For the determination of the first two parameters, the procedure `EVENT2` (generating the events) calls the service routine `PROPORTION` to calculate the temporal weight (*prop*) of each sound (i.e., element) with respect to the total attack duration of the event. Using this last parameter (*at\_dur*—i.e., the part of the event in which the elements can begin to play), the start time of the next element is calculated by the begin time of the current one by the formula

$$nextbeg' = nextbeg + prop \cdot at\_dur \quad (14)$$

The total attack duration of the event is therefore partitioned between all the (parent) elements in the event.

In each event, the elements are numbered by a counter variable *cnt*: The first sound that is played is taken as number one (i.e., *cnt* = 1), and so on. The duration (*el\_dur*) of each element (i.e., sound) is determined by weighting the time remaining from the beginning of the sound to the end of the event, with a factor directly related to the number (*cnt*) of the element generated, in accordance with the expression

$$el\_dur = (beg + dur - el\_beg) \cdot \left( \frac{cnt}{elements} \right)^{2 \cdot ext} \quad (15)$$

where *beg* and *dur* are, respectively, the start time and the duration of the event, *elements* is the number of parent elements in the event, and *el\_beg* is the start time of the current element. Here, *ext* represents a weighting factor for an exponential interpolation and lies in the range  $0.8 < ext < 1.5$ .

An important case occurs when there is no overlapping between two consecutive sounds. In this case, Mr. Chowning imposed an “overlapping condition” on the sounds, making the element with no overlap longer by means of the formula

$$el\_dur = (nextbeg - el\_beg) \cdot 1.25 \quad (16)$$

where *nextbeg* represents the begin time of the next sound and *el\_beg* the current one.

The duration of the attack of each element in the event is determined by an exponential interpolation between two values: the attack duration of the first element, and the attack duration of the last element in the event. To make this interpolation, a parameter *interp* is used to calculate the position of the element in the event. This parameter is obtained by

$$interp = \frac{el\_beg - beg}{at\_dur} \quad (17)$$

which is the distance between the beginning of the element and the beginning of the event, normalized to *at\_dur* (where  $0 < at\_dur < 1$ ). Using this parameter, the attack time of the current element is calculated by exponential interpolation between the initial and final values (*INITATT* and *ENDATT*) according to

$$attack\_time = el\_dur \cdot INITATT \cdot \left( \frac{ENDATT}{INITATT} \right)^{interp} \quad (18)$$

The same procedure is used for the decay time. By contrast, these parameters can also be determined as a function of frequency as a way to obtain short attacks for low pitches and long attacks for high pitches; the inverse applies for the decay time.

## Spatialization

In *Stria*, spatialization is given by three factors: the reverb percentage of all the sounds in the event, the

Figure 9. Amplitude diagram for the loudspeaker located at  $\text{deg} = 225^\circ$ .

apparent angle of the sound source, and the apparent distance of the sound source.

The reverberation is constant for every parent element in the event, and the apparent angle of the source is given by the quadraphonic diffusion of the sound, by calculating the four amplitude factors for the four playback loudspeakers. For the speaker positioned at the angle  $d$  (equal to  $45^\circ$ ,  $135^\circ$ ,  $225^\circ$ , or  $315^\circ$ ) that is to emit a sound apparently diffusing from the angle  $\text{deg}$ , the amplitude coefficient is calculated by the formulas

$$\text{amp\_fact} = \sqrt{\frac{\text{deg} - d + 90}{90}} \quad \text{for } d - 90 < \text{deg} < d \quad (19)$$

$$\text{amp\_fact} = \sqrt{\frac{d + 90 - \text{deg}}{90}} \quad \text{for } d < \text{deg} < d + 90 \quad (20)$$

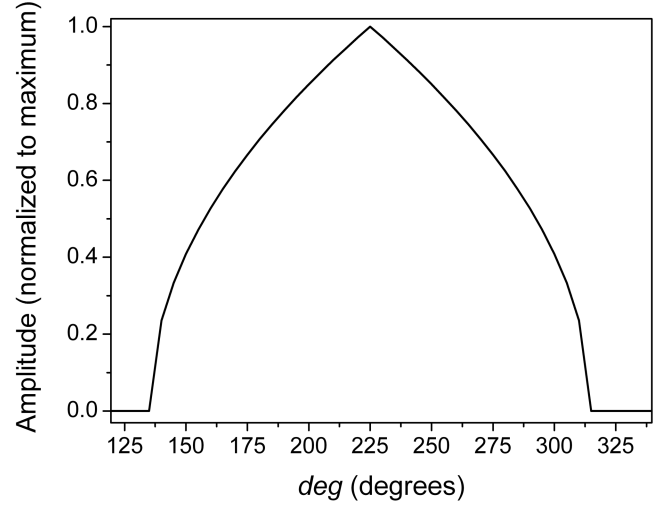
Figure 9 shows the amplitude diagram for the speaker located at  $d = 225^\circ$ , playing a sound apparently diffusing from angle  $\text{deg}$ .

Each event has many elements, numbered by the variable  $\text{cnt}$ : The reference angle  $\text{el\_deg}$  of each parent element in one event is calculated by the formula

$$\text{el\_deg} = \text{ev\_deg} + 360 \left( \frac{\text{cnt}}{\text{elements} - 1} \right)^{0.2} \quad (21)$$

where  $\text{ev\_deg}$  is the reference angle of the current event. This means that the elements in the event rotate around the listener more than  $360^\circ$  per event. Another rotation is introduced from event to event: At the end of the generation of an event (a parent or a recursive one), in fact, there is a rotation of  $-90^\circ$ : A slow rotation on the events is superimposed on the one owing to the elements, but in the opposite direction. As we will see in the following section, the child events, generated during recursion, generally spin around the listener faster than the parent events, generating dynamism.

The apparent distance of a sound from the source is given by varying the ratio between the intensity of the direct and reverberated sound components. The latter is kept fixed in the whole generation of parent elements, whereas the former can vary from element to element in an exponential way according to



$$\text{dis} = (\text{cnt} \cdot \text{DIS\_SCALE})^{0.8} \quad (22)$$

where  $\text{DIS\_SCALE}$  is the reference distance scale of the event, and  $\text{dis}$  is the distance parameter of the current element, numbered by  $\text{cnt}$ , obtaining an apparent movement of the source (and thus position changes exponentially). When  $\text{DIS\_SCALE}$  is negative, then  $\text{dis}$  is constant at the magnitude of  $-\text{DIS\_SCALE}$ .

## Recursion

The `EVENT2` procedure may use recursion for the generation of the sounds. After the creation of each element, `EVENT2` checks if two conditions are verified:

1. The number of recursive calls already done in the event is less than the maximum imposed by input (usually one recursion per event, and never more than one).
2. The value weighting factor *prop* (see the preceding section) for the current element should have a particular value. (This condition is verified on average one time every five elements.)

If both of these conditions are verified, the current element is the parent of a child event, and `EVENT2` calls itself with new modified parameters to con-

---

struct the new event, which has different characteristics from the parent one (i.e., a recursive call).

The child event has duration and attack duration shorter than that of the parent. (These parameters are scaled by  $prop < 1$  in the recursive calls.) This means that the events generated by recursion are shorter than the originals; in addition, because the attack duration of the child event is shorter, its elements are closer than the originals, increasing the dynamism (i.e., shorter events with closer elements).

The number of child elements played in an event is kept less than or equal to 9 (although the number of parent elements can be greater), in order not to exceed the number of available instruments. The base frequency for the new event is given by the expression  $fff * freq$ , that is, it is equal to the frequency of the note played by the parent element which generated the recursive call. The space variable is chosen by imposing  $|power| = 1$ , and this means that the recursive event occupies only one pseudo-octave above or below its base frequency (above if the original value of  $power$  was less than unity, and vice versa). In the case in which the new base frequency is greater than 1618 Hz,  $power$  is kept equal to  $-1$  to avoid frequency divergence owing to recursion.

The frequency space occupied by the child event is divided into 9 tones (i.e., no 18-tone division) to avoid creating spectra that are too "rich." The child event begins the moment in which the parent begins, and has attack and decay times calculated by the input values due to recursion. The reverb percentage for the children is 1.2 times the parent's, resulting in a greater reverberation percentage in a shorter duration. The rotation around the listener is generally faster than the parent's, depending on a smaller number of elements in a shorter event. The generation of parent elements continue after the creation of the child elements, starting, in the score, from the moment in which it was interrupted.

The meaning of recursion in *Stria* can be understood from the preceeding considerations. During the generation of an event, one element can create a recursion, which generates another shorter event, starting from the moment in which the parent element begins. This new event is shorter and has closer elements, spins around the listener faster

than the originals, occupies a frequency space one pseudo-octave wide and a base frequency equal to the note played by the parent element. The reverberation percentage is greater than the parent's, for a shorter time, to increment the acoustic weight of this event. This, in other words, means that the recursion generates an explosion of the sounds at the time of the recursive call, with many close and short sounds rotating fast around the listener, creating a dynamic sound environment.

## Conclusions

A detailed study of the process used by John Chowning for the composition of *Stria* has been presented. The analysis has been carried out starting from the study of the programming language used for the composition of the piece. Fundamental characteristics of the piece have been studied and described. The analysis of the compositional process has been conducted by means of the reading (line by line) of the algorithms used for the generation of the piece. The mathematical relationships and the parameters used as input for the programs have been described in detail; this analysis highlights the accurate formalism that stands at the basis of the piece, both in terms of micro-structural and global parameters, and complement the knowledge derived by the listening of the piece. The important use of recursion for the generation of superposition of sounds has also been addressed.

## Acknowledgments

I want to thank John Chowning for his help during these years of research. In particular, the clearness of his explanations, the helpfulness in answering to my electronic mails, and his enthusiasm in retrieving information on algorithms and function were of vital importance for the development of my analysis of the piece. Special thanks also to Professor Giovanni de Poli, who (in 2002) suggested that I carry out this work, and during the following years was always helpful and available for discussion.

## References

- Baudouin, O. 2007. "A Reconstruction of *Stria*." *Computer Music Journal* 31(3):75–81.
- Bossis, B. 2005. "Stria de John Chowning ou l'oxymoron musical: du nombre d'or comme poétique." In P. A. Castanet et al., eds. *John Chowning: Portraits polychromes*. Paris: INA/GRM and Michel de Maule, pp. 87–115.
- Chowning, J. 1971. "The Simulation of Moving Sound Sources." *Journal of the Audio Engineering Society* 19(1):1–6.
- Chowning, J. 1977a. T0.MEM, T286.MEM, T466.MEM, T610.MEM, T754G.MEM, and END.MEM. Computer file printouts. John Chowning Papers, Stanford University Archives.
- Chowning, J. 1977b. CPC4.INS. Computer file printout dated 8 September. John Chowning Papers, Stanford University Archives.
- Chowning, J. 1977c. COMP.SAI. Computer file printout dated 3 October. John Chowning Papers, Stanford University Archives.
- Chowning, J. 1999. "Perceptual Fusion and Auditory Perspective." In P. Cook, ed. *Music, Cognition, and Computerized Sound: An Introduction to Psychoacoustics*. Cambridge, Massachusetts: MIT Press, pp. 261–275.
- Chowning, J. 2003. Personal communication.
- Chowning, J. 2007. Personal communication, March.
- Dahan, K. 2007. "Surface Tensions: Dynamics of *Stria*." *Computer Music Journal* 31(3):65–74.
- Dodge, C., and T. A. Jerse. 1985. *Computer Music: Synthesis, Composition and Performance*. New York: Schirmer.
- Markowsky, G. 1992. "Misconceptions about the Golden Ratio." *The College Mathematics Journal* 23(1):2–19.
- Meneghini, M. 2003. "*Stria* by John Chowning: Analysis of the Compositional Process." Paper presented at the XIV Colloquium on Musical Informatics, Florence, Italy, 8–10 May. Available on-line at [www.dei.unipd.it/~menego/CIM2003\\_Meneghini.pdf](http://www.dei.unipd.it/~menego/CIM2003_Meneghini.pdf).
- Runion, G. E. 1990. *The Golden Ratio*. Palo Alto, California: Dale Seymour.
- Smith, N. W. 1976. "SAIL Tutorial." Available on-line at [pdp-10.trailing-edge.com/decuslib20-01/01/decus/20-0002/sail.tut.html](http://pdp-10.trailing-edge.com/decuslib20-01/01/decus/20-0002/sail.tut.html).