

Memorie del 1º Colloquio su:

"ASPETTI TEORICI DI INFORMATICA MUSICALE"

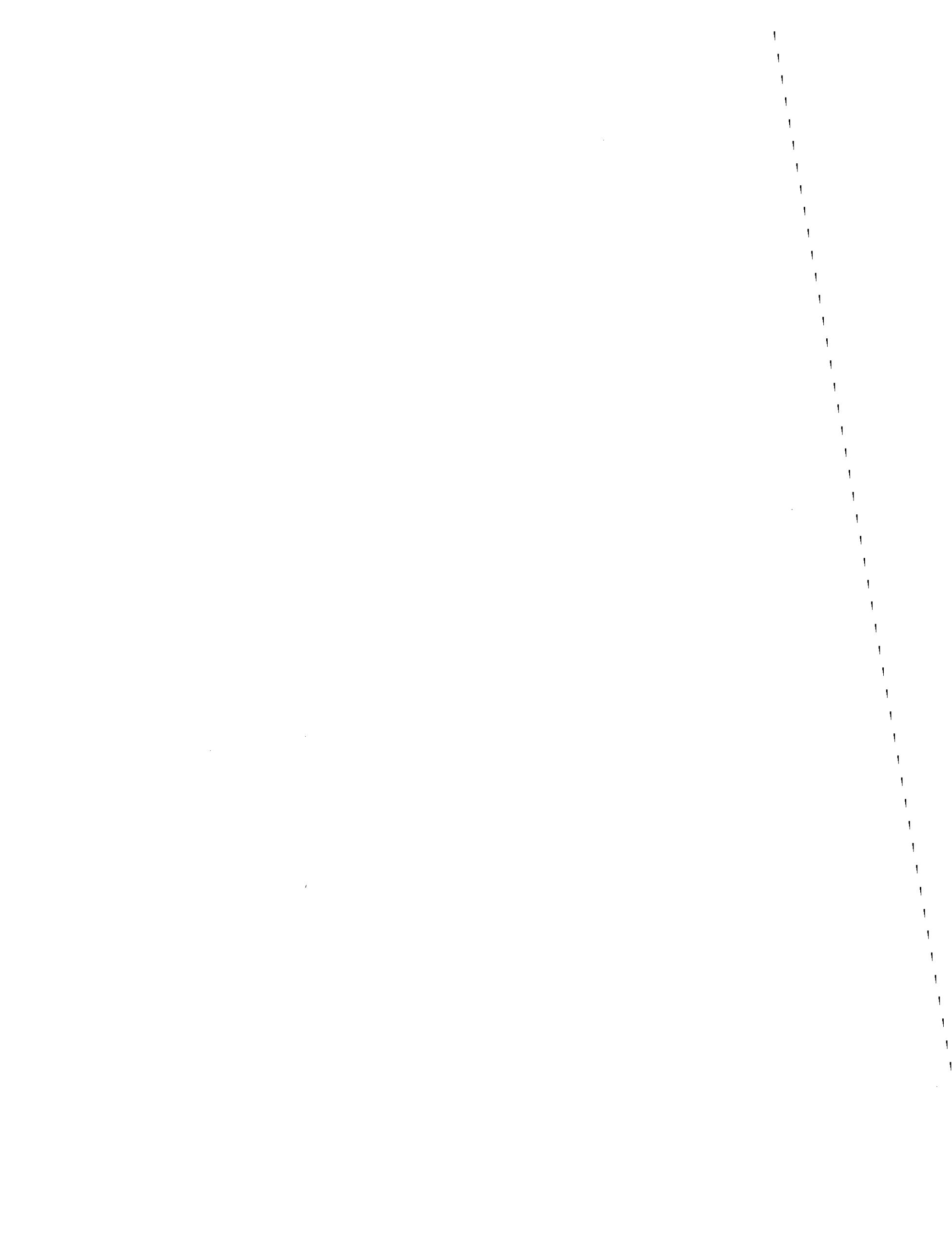
14 dicembre 1977

INDICE DELLE MEMORIE

- Baroni/Jacoboni - "Verso una grammatica della melodia"
- Bertoni/Haus/Mauri/Torelli - "Analisi e compattazione
 - di testi musicali"
- De Poli/Vidolin - "EMUS: un programma per l'elaborazione
 - di strutture musicali"
- Bolognesi - "Automazione di processi compositivi: criteri
 - di utilizzo della sorgente stocastica"
- Razzi - "Relazione costante tra "produttori" (musicisti
 - e scienziati) e pubblico"
- Melzi/Ozzola - "Hardware e software per la composizione
 - musicale automatica in tempo reale"
- Martissa - "Analisi e sintesi di processi pseudo-musicali"
 - estratto dalla tesi
- Gruppo DOC (RAI Milano) - "Computer music e produzione di
 - programmi radiotelevisivi"
-  Tisato - "Un sistema interattivo di software per la sintesi
 - del suono in tempo reale"
- Fugazza/Menchinelli - "Dispositivo per la generazione di
 - suoni mediante sintesi additiva
 - armonica operante per mezzo di peri-
 - feriche veloci controllate da un
 - microcalcolatore"

- Cavaliere/Di Giugno/Fedullo/Ortosecco - "Un calcolatore
veloce orientato per la sintesi di segnali
acustici in tempo reale"

Istituto di Cibernetica - Università di Milano



M. BARONI - E. SACOBONI

"VERSO UNA GRAMMATICA DELLA MELODIA"

Per due ragioni abbiamo apposto a queste note lo stesso titolo di un nostro studio pubblicato qualche mese fa⁽¹⁾; sia perché nella nostra relazione dovremo riferirci a quello studio e riassumerne i contenuti essenziali, sia perché le ricerche che abbiamo continuato anche dopo la sua pubblicazione - alle quali pure accenneremo in questa sede - continuano nella stessa direzione.

Il problema di cui ci stiamo occupando si trova all'incrocio fra due vie entrambe problematiche: una riguardante il concetto di grammatica e l'altra riguardante quello di melodia.

La grammatica: la musica, come tutti i prodotti della fantasia è retta da norme legate alla vita emozionale e non obbedienti a principi logici; tuttavia nel momento in cui la fantasia si incarna in oggetti sonori, in entità fisiche, acquista forme misurabili e i suoi procedimenti di costruzione assumono l'aspetto di regole di strutturazione. Col termine di 'grammatica' intendiamo appunto queste regole, solo in parte consapevoli al compositore stesso.

La melodia: le infinite forme di strutturazione del sonoro non sono tuttavia predeterminabili; alcune di esse hanno assunto nei secoli un ruolo preminente e una fisionomia distinata: vedi il caso della melodia. Ora, se è vero che col termine di 'grammatica' possiamo intendere un insieme di regole applicate dal singolo compositore, è altrettanto vero che la melodia in quanto tale - indipendentemente da ogni variabile stilistica o individuale - richiede essa stessa un insieme di regole che la strutturino come entità formale codificata e distinta da altre entità formali possibili.

Il nostro studio si colloca attualmente all'incrocio di queste due vie: tende infatti a trovare le regole di stile comuni ad un corpus il più possibile omogeneo (la parte del soprano nei corali a quattro voci di Bach) secondo un'ottica sincronica che tiene ai margini sia il problema concreto degli 'scarti dalla norma' - ossia delle regole individuali - sia quello della definizione in astratto delle regole

1) M.BARONI-C.JACOBONI, Verso una grammatica della melodia, Scuola di perfezionamento in musicologia dell'Università di Bologna, 1976.

generali della melodia. Si tratta di una scelta consapevole dei propri criteri che è giustificata dall'estrema complessità del tema e dalla necessità di partire dalla sua formulazione più semplice.

- - - - -

Nel testo pubblicato il problema era affrontato nei suoi termini più elementari: si trattava di esaminare le prime due frasi dei corali non modulanti, composti in 4/4 e in modo maggiore. Il corpus da esaminare era dunque ridotto a 56 coppie di frasi la cui analisi doveva corrispondere però a criteri precisi: in primo luogo doveva trasformarsi in regole di generazione automaticamente applicabili da parte di un computer, cioè doveva essere subordinata alla necessità di una successiva sintesi, e in secondo luogo doveva presentare caratteri di completezza cioè essere tale da permettere una ricostruzione stilisticamente attendibile delle frasi analizzate. A questo punto solo la 'competence' di un esperto in corali poteva giudicare dell'attendibilità del prodotto.

Come si vede, sia il procedimento induttivo-deduttivo, sia il ricorso alla 'competence' rivelano chiari legami del metodo adottato con i modelli della linguistica chomskyana. Tuttavia abbiamo cercato di tenerci il più possibile al riparo dalla indiscriminata tendenza tipica della semiologia musicale degli anni '60 - ad assumere i modelli di formalizzazione del linguaggio verbale come strumenti di indagine applicabili per principio al linguaggio sonoro. Così abbiamo preferito in questa occasione correre il rischio consapevole di ottenere una grammatica che ha assunto la forma di un insieme non organizzato di regole piuttosto che partire dal presupposto non dimostrato che anche una gerarchia di regole musicali dovesse assumere una forma simile a quella delle gerarchie sintattiche di Chomsky.

Ogni nota è stata esaminata in base a quattro tratti distintivi: la durata, la posizione (rispetto agli accenti della battuta), l'altezza e il grado della scala di appartenenza (a cui ha corrisposto la distinzione dei gradi forti - I, III, V - dagli altri gradi).

Ogni coppia di frasi è stata divisa in prima e seconda frase e

ogni frase in tre parti: nota iniziale, corpo centrale, cadenza. Questo è stato sufficiente a fornirci un' insieme di regole locali applicabili ad ogni nota singolare tali da permettere una generazione di tipo sequenziale (dalla prima all'ultima nota). Tale insieme di regole si è rivelato non sufficiente ad esaurire la complessità della struttura. E' stato dunque necessario introdurre una serie di regole globali (intendendosi con questo termine tutte le regole che implicano rapporti fra note non contigue e fra insiemi di note).

Alla divisione delle regole in due categorie ha corrisposto, in sede di generazione, una diversa utilizzazione del computer. Le regole locali sono state introdotte in forma sequenziale come modi di organizzare il passaggio da una nota alla nota successiva e dunque di generare le due frasi. Le regole globali sono invece state introdotte in forma di 'rigetto', cioè come approvazione o disapprovazione delle frasi generate. L'adozione di questa tecnica è strettamente connessa all'adozione del principio sequenziale: se infatti noi avessimo adottato le regole globali in forma generativa e non in forma di rigetto, avremmo rischiato di assegnare alla prima nota della frase una del tutto arbitraria funzione 'previsionale' nei confronti dell'ultima, cioè avremmo assegnato alla ~~p~~rima nota una libertà maggiore e la facoltà di determinare la natura delle altre.

Fra le note che compongono la frase è stata poi introdotta una forma di gerarchia sulla base della distinzione - accertata nel corpus stesso - fra condotta melodica semplice e condotta melodica 'fiorita'. Alcune melodie, infatti, si presentano chiaramente come 'varian-~~t~~à' di melodie più semplici fra una nota e l'altra delle quali sono state inserite note di passaggio, appoggiature, note di volta ed altri abbellimenti.

Questa osservazione non solo ci ha permesso di distinguere le note che abbiamo chiamato fondamentali da quelle di fioritura e di introdurre le regole corrispondenti, ma ci ha consentito di ipotizzare una norma riguardante i salti. Abbiamo notato, infatti, che le nostre melodie procedevano ~~per~~ ^{con} salti di ~~grado~~ per grado congiunto e che, nei casi in cui ciò non si verificava, esse procedevano per salti fra gradi forti o per intervalli successivamente 'riempiti' da inflessioni melodiche i cui profilo era esattamente corrispondente alla forma di certe

fioriture (e queste inflessioni abbiamo chiamato figure di fioritura).

Si delineava in questo modo una possibile ipotesi 'trasformazionale' che recava chiari indizi di come, in una melodia, si possano distinguere strutture sintatticamente più profonde' (come la melodia antecedente l'introduzione delle fioriture o antecedente all'introduzione del salto e della figura di fioritura che lo 'riempie') e strutture più superficiali' (quelle della melodia nella sua stesura definitiva).

Nella precedente ~~etapa~~^{etapa} del lavoro questa distinzione fra strutture profonde e strutture superficiali è rimasta allo stadio in cui ora l'abbiamo descritta. Nella fase successiva, alla quale stiamo attualmente lavorando, questo aspetto ha assunto un'importanza centrale anche dal punto di vista del metodo di analisi, tanto da indurci a mettere in dubbio l'ipotesi sequenziale precedentemente adottata e a sviluppare invece un'ipotesi generativo-trasformazionale più vicina ai modelli chomskyani oltre che a certi modelli ahalitici proposti qualche decennio or sono da musicologi come Riemann e Schenker.

In questa seconda fase ~~dall'elaborazione~~ si è ampliato anche il corpus dei corali da analizzare: vi abbiano inclusi, infatti, le melodie del soprano dell'intero corale e ci siamo proposti di affrontare i problemi della struttura melodica nella tonalità minore, quelli della modulazione e quelli della struttura formale dell'intero corale, ivi includendo le questioni riguardanti la coerenza tematica dei materiali melodici. Non è mutata invoca l'intenzione di continuare ad usare il computer come strumento di verifica delle nostre ipotesi, come di mantenere strettamente congiunti il lavoro di analisi e ~~all'inizio~~^{quale} di sintesi, nonché di sottoporre la sintesi stessa al giudizio della nostra 'competence'.

Questa seconda fase del nostro studio - lo accennavamo in precedenza - è ancora in elaborazione, e possiamo descriverla solo per quanto ci è consentito dalla provvisorietà delle nostre ipotesi attuali.

La prima ipotesi è che la melodia (o per lo meno la melodia dell'epoca tonale a cui appartiene il nostro corpus) si configuri come un itinerario strutturato in tappe successive ciascuna delle quali si conclude con una formula 'portatrice di conclusività' (la cadenza) che conduce a una nota (la nota finale di ogni frase) che sta in rela-

con
zioni determinate della tonica della tonalità d'impianto la quale conclude l'itinerario generale. Compito dell'analisi sarà in questo caso di individuare la rete strutturale che organizza l'insieme ordinato delle note del corale.

La seconda ipotesi è che ciascuna frase di una melodia si configuri come un itinerario che, a partire da un intervallo fondamentale (o nucleare come proponiamo di chiamarlo) che è costituito da una nota 'iniziale' e da una nota 'finale', arricchisca questo intervallo con altre note poste al suo interno e alla sua sinistra e strutturate fra loro da regole di concatenazione semplici e ben determinate. In questo caso l'analisi dovrà individuare l'intervallo nucleare di ogni frase e stabilire in quale ordine e seguendo quali regole le note seguenti l'abbiano arricchito.

Secondo questa ipotesi la grammatica assume la forma di una serie di trasformazioni dall'intervallo nucleare alla frase finale; ognuna di queste trasformazioni corrisponde a una 'figura' determinata; con non più di dieci figure è possibile ricostruire interamente i 123 corali, cioè le più di mille frasi che ora costituiscono il nostro corpus.



ANALYSIS AND COMPACTING OF MUSICAL TEXTS

A. Bertoni - G. Haus - G. Mauri - N. Torelli

ABSTRACT

We show that it is possible to analyze and describe the structure of musical texts in a compact form by identifying the passages which can be obtained from preceding passages by applying suitable operators. We consider a number of operators of musical interest and report some preliminary results of the analysis of a few pieces of music.

INTRODUCTION

The first part of this paper is devoted to the theoretic analysis and the formalization of the description in a compact form of generic sequences of symbols without a specific reference to musical structures: more precisely, we have shown (by using only the operator of repetition, but results can be generalized) how it is possible to associate the sequence of symbols to a shorter one, which we call "processor", from which it is possible

BERTONI - HAUS - MAURI - TORELLI

"ANALISI E COMPATTAZIONE DI TESTI
MUSICALI"

to re-construct univocally the original sequence. The most interesting aspects, also from the musical point of view concern the relations between the length of a sequence and that of the shortest processor, called "optimal processor", which can be associated with it; in fact, their ratio is an indicator of the "difficulty of description" of a sequence or, if you want, of its "grade of repetition". Starting from this, it is possible to define in formal way, a measure of "descriptive complexity" of a sequence, generalizing a similar concept given by Kolmogorov (1968).

For this reason, we have paid particular attention to the determination of upper and lower bounds for the "descriptive complexity" of a sequence of a given length. Some theorems have been proved that give, for the lower and upper bound respectively, the values:

$$\underline{\psi}(n) = \log_2 n + 1 \quad \overline{\psi}(n) = O(n/\log_2 n)$$

Afterwards an algorithm is described which permits the determination of the descriptive complexity of sequences of length n in a time $t(n) = O(n^2/\log_2 n)$, operating in a way similar to the algorithm given by Paterson (Aho et al., 1974).

In the second part of the paper, these results are applied to the analysis of musical sequences. First of all it has been necessary to identify the class of operators best suited to this type of structure, on the basis of historical motivations and of some former works (Schoenberg, 1954).

At this point, the implementation of the described algorithm permits, through the operators determined in this way, to analyze sequences describing musical texts. The program provides us with the optimal processor for the analysed sequence, the ratio between the length of the optimal processor and that of the original sequence, as well as the graph of the complexity as a function of the length of the sub-sequence already analyzed. The aspect of such a graph provides information on the regularity of the musical text in question, allowing one to identify also visually the structural elements which constitute the sequence, and to detect the introduction of new structural elements, such as, for example, new melodic themes, in the musical process.

FUNDAMENTAL DEFINITIONS

Def. 1 - Given a finite alphabet Σ , the set Σ^+ of the sequences of letters of Σ is defined recursively as follows:

i) $\sigma \in \Sigma$ is a sequence of length $l(\sigma) = 1$;

ii) if x is a sequence and $\sigma \in \Sigma$, $x\sigma$ is a sequence of length $l(x\sigma) = l(x) + 1$.

If a sequence contains sub-sequences repeated more than once, it is possible to represent explicitly only the first of these sub-sequences and to indicate synthetically the others with an operator symbol, $\tilde{\wedge}$ (in this paragraph, as already said, we consider only the operator of repetition), applied to the original sub-sequence, represented through the positions of its initial and final letters in the sequence. If we call a "processor" one of the synthetic representations obtained in this way, we can associate to it a length and a weight accordingly to the following definition.

Def. 2 - i) $\sigma \in \Sigma$ is a processor of length $L(\sigma) = 1$ and of weight $\omega(\sigma) = 1$;

ii) if p is a processor and $\sigma \in \Sigma$, $p\sigma$ is a processor of length

$L(p\sigma) = L(p) + 1$ and of weight $\omega(p\sigma) = \omega(p) + 1$;

iii) if p is a processor and $0 < s \leq m \leq \omega(p)$, then $p\tilde{\wedge}(s, m)$ is a processor of length $L(p\tilde{\wedge}(s, m)) = L(p) + 1$ and of weight $\omega(p\tilde{\wedge}(s, m)) = \omega(p) + m - s + 1$.

Let's denote by $P(\Sigma)$ the set of processors of sequences of Σ^+ . Obviously, given a processor, it is possible to reconstruct in a univocal way the complete sequence which it represents.

Def. 3 - We say interpretation of the processors the function $W : P(\Sigma) \longrightarrow \Sigma^+$ defined by:

i) $W(\sigma) = \sigma$;

ii) $W(p\sigma) = W(p)\sigma$;

iii) $W(p\tilde{\wedge}(s, m)) = W(p)y$, where y is the subsequence of $W(p)$ included between the positions s and m .

Example: $W(aab\tilde{\wedge}(2,3)\tilde{\wedge}(1,4)c) = aababaabac$.

If we indicate by C_W the range of W , it is immediately proved, by induction on the construction of the processors (Def. 2), the following theorem:

Th. 1 - The interpretation $W: P(\Sigma) \rightarrow \Sigma^*$ is a total function, that is every processor generates a sequence; moreover, $C_W = \Sigma^*$ and $\omega(p) = l(W(p))$ for every $p \in P(\Sigma)$.

This means that every sequence is an interpretation of at least one processor (in case the sequence itself: in fact a sequence is also a processor); but in general it is an interpretation of many processors of different lengths. The minimum length of these processors depends on the structure of the sequence, besides its length, and characterizes in one sense the "difficulty of description" or descriptive complexity of the sequence.

Def. 4 - We say descriptive complexity of a sequence $x \in \Sigma^*$ of length n the value:

$$H(x, n) = \min \{L(p) / p \in P(\Sigma) \text{ and } W(p) = x\}.$$

It is straightforward to prove that $L(p) \leq \omega(p)$ for every $p \in P(\Sigma)$, so that $H(x, n) \leq n$.

If now we consider the class of all the sequences of equal length n , we can define the greatest lower and least upper bounds of the descriptive complexity inside this class:

$$\bar{\Psi}(n) = \max \{H(x, n) / l(x) = n\}$$

$$\underline{\Psi}(n) = \min \{H(x, n) / l(x) = n\}$$

The values of these bounds give us the limits within which it is possible to compact a sequence, while the comparison of the descriptive complexity of an analysed sequence with these values can give an indication of the degree of "regularity" of the sequence itself.

Th. 2 - $\underline{\Psi}(n) = \lceil \log_2 n \rceil + 1$, where $\lceil k \rceil$ represents the least integer not less than k .

Proof. The descriptive complexity for sequences of length n is obviously minimum for the sequences of type σ^n , with $\sigma \in \Sigma$. In such case an optimal processor p (that is one of minimum length) such that $W(p) = \sigma^n$ is:

$$p = \begin{cases} p' \tilde{w}(1, n/2) & \text{if } n \text{ is even} \\ p'' \tilde{w}(1, (n-1)/2) & \text{if } n \text{ is odd} \end{cases}$$

where p' is an optimal processor for the sequence $\sigma^{n/2}$ and p'' for the sequence $\sigma^{(n-1)/2}$.

Therefore $\underline{\Psi}$ can be defined recursively by:

i) $\underline{\Psi}(1) = 1$;

ii) $\underline{\Psi}(n) = \underline{\Psi}(n/2) + 1$ if n is even, $\underline{\Psi}((n+1)/2) + 1$ otherwise.

For example, $\underline{\Psi}(18) = \underline{\Psi}(9) + 1 = \underline{\Psi}(5) + 2 = \underline{\Psi}(3) + 3 = \underline{\Psi}(2) + 4 = \underline{\Psi}(1) + 5 = 6$.

Then, immediately follows:

$$\psi(n) = k \text{ where } 2^{k-2} < n \leq 2^k - 1.$$

Therefore:

$$k-2 < \log_2 n \leq k-1 \implies \log_2 n + 1 \leq \psi(n) < \log_2 n + 2, \text{ and finally:}$$

$$\psi(n) = \lceil \log_2 n \rceil + 1.$$

Corollary - $\psi(n) \sim \log_2 n$ ($n \rightarrow \infty$).

Example: The sequence $a \dots a$ of length 16 will be represented by the processor
 $\pi(1,1) \pi(1,2) \pi(1,4) \pi(1,8)$

of length $5 = \log_2 16 + 1$.

More complex is the determination of the least upper bound. A particular case immediately evident is:

Th. 3 - If $|\Sigma| = 1$, then $\bar{\psi}(n) = \psi(n) = \lceil \log_2 n \rceil + 1$.

In general, an upper bound to the value of $\bar{\psi}(n)$ is given by the following theorem.

Th. 4 - There exists a constant c such that $\bar{\psi}(n) \leq cn/\log_2 n$.

Proof. Let $|\Sigma| = 2$ (the proof is analogous for $|\Sigma| > 2$).

Given a sequence of length $n \geq 4$, determine the integer k for which:

$$a) k \cdot 2^k \leq n < (k+1) \cdot 2^{k+1}.$$

Since $2^k > k \geq 1$, a) immediately implies $2^{2k} \leq n \leq 2^{5k}$, so that we have:

$$b) (1/5) \cdot \log_2 n \leq k \leq (1/2) \cdot \log_2 n.$$

Now divide the sequence of length n in sub-sequences, all of length k , except, at most, the last; since the sub-sequences obtained in this way are at least 2^{2k} (being $n \geq k \cdot 2^{2k}$ by a), while the distinct sequences of length k are only 2^k , there exist, at least, two identical sub-sequences; substitute the second of these with $\pi(s, sk - 1)$, where s is the position of the beginning of the first sub-sequence. Repeat the argument for the remaining sequence, till its length h verifies $h \geq k \cdot (2^k + 1)$. When $h < k \cdot (2^k + 1)$ terminate the described procedure. In this way a processor is obtained, p , whose length is given by:

$$L(p) = h + (n-h)/k \leq n/k + (k-1) \cdot (2^k + 1) = n/k + k \cdot 2^k + k - (2^k + 1)$$

Since $k \leq 2^k + 1$, we conclude:

$$c) L(p) \leq n/k + k \cdot 2^k.$$

Since $k \cdot 2^{2k} \leq n$ implies that $k \cdot 2^k \leq \sqrt{nk}$, from c) follows $L(p) \leq n/k + \sqrt{nk}$,

and finally, from b):

d) $L(p) \leq 5n/\log_2 n + \lceil \frac{n\log_2 n}{2} \rceil \sim 5n/\log_2 n \leq (5+\varepsilon')n/\log_2 n \quad (n \rightarrow \infty),$
which is the thesis.

The following theorem shows that the asymptotic value n/\log_2 is not ulteriorly improvable.

Th. 5 - If $|\Sigma| \geq 2$, there exists a sequence x of length n such that:

$$H(x, n) \geq c'n/\log_2 n \quad (c' \text{ constant}).$$

Proof. Let $\Sigma = \{a, b, \dots\}$. Given n , take the integer k such that:

a) $2^k > n \geq 2^{k-1}$.

Then construct a sequence x of length n with the following characteristics:

1) $x = \langle x_1, x_2, \dots, x_m, x_{m+1}, \dots \rangle$, with $x_s \in \{a, b\}^*$ ($1 \leq s \leq m+1$);

2) $I(x_s) = k$ for $1 \leq s \leq m$; $I(x_{m+1}) \leq k-1$;

3) All the x_s are distinct (this is possible by a).

Let p be any processor of x . By construction, in x there are no repetitions of sequences of length $2k+1$, since that would imply the existence of two subsequences x_i and x_j , identical, against 3). Then $L(p) \geq n/2k$.

From a) follows $\log_2 n \leq k \leq \log_2 n + 1$, therefore $L(p) \geq n/(2\log_2 n + 2) \sim n/2\log_2 n$, and finally:

$$H(x, n) \geq (1/2 - \varepsilon) \cdot n/\log_2 n.$$

From the two preceding theorems immediately follows:

Th. 6 - If $|\Sigma| \geq 2$, then $\bar{\psi}(n) = O(n/\log_2 n)$.

A last question of a general character is the determination of the number $N(n)$ of distinct processors for which $\omega(p) = n$.

First of all, it is evident that $N(1) = |\Sigma|$. For $n > 1$, a processor p for which $\omega(p) = n$ is of type:

a) $p = p' \sigma$ with $\omega(p') = n-1$, or

b) $p = p'' \tau$ (s, m) with $1 \leq s \leq m \leq \omega(p'')$, $m-s+1 = k \leq n/2$ and $\omega(p'') = n-k$.

If $|\Sigma| = q$, there exist exactly $q \cdot N(n-1)$ possible processors of type a) and $N(n-k) \cdot (n-2k+1)$ of type b) for every k such that $1 \leq k \leq n/2$, so that it is possible to write the recurrence relation:

1) $N(1) = q$

2) $N(n) = \sum_{k=1}^{\lfloor n/2 \rfloor} (n-2k+1)N(n-k) + q \cdot N(n-1).$

From 2) follows that, for n large enough,

$$3) (n-1+q)N(n-1) \leq N(n)$$

from which $N(n) \geq n \cdot N(n-1)$, and finally:

$$4) N(n) \geq q \cdot n! \sim q(n/e)^n \sqrt{2\pi n}.$$

By 4) the number of processors whose interpretations are the q^n sequences of Σ^+ of length n is very large. In particular, there exists a n -sequence which is an interpretation of at least $(n/cq)^n$ distinct processors.

AN ALGORITHM TO DETERMINE THE DESCRIPTIVE COMPLEXITY

Let us now deal with the problem of finding an efficient algorithm to determine $N(x, n)$ for every sequence x . The fundamental observation is the following:

Lemma - With every processor p , of $x\sigma$ it is possible to associate a processor p' such that $W(p') = x$ and $L(p') \leq L(p)$.

In fact, if p is of type $p = p'' \text{Tr}'(s, m)$ let:

$$p' = \begin{cases} p'' & \text{if } s = m, \\ & : \\ & : \\ p'' \text{Tr}'(s, m-1) & \text{if } s < m. \end{cases}$$

If, on the contrary, p is of type $p''\sigma$ let $p' = p''$.

In both cases one has $W(p') = x$ and $L(p') \leq L(p)$, as is easy to verify.

An immediate extension of the preceding lemma is:

Th. 7 - With every processor p of xz (where z may be empty) we can associate a processor p' of x such that $L(p') \leq L(p)$.

Now given a sequence x , the following decomposition, if existing, is unique:

A) $x = ayb$, where $a, b \in \Sigma^*, y \in \Sigma^+$ and furthermore:

i) a does not admit y as a sub-sequence.

ii) y is the maximum suffix of x for which a decomposition of type A) holds.

If a decomposition of type A) does not exist, we shall consider the decomposition:

B) $x = z\sigma$ with $z \in \Sigma^*$, $\sigma \in \Sigma$

Now construct a processor of x , $p^\wedge(x)$, with the following rules:

i) If there exists a decomposition of x of type A), then

$p^\wedge(x) = p^\wedge(ayb) \text{Tr}'(s, m)$, with $s = l(a)+1$ and $m = l(ay)$.

ii) If such a decomposition does not exist, then, from B) $p^\wedge(x) = p^\wedge(z)\sigma$.

The length of $p^\wedge(x)$ is exactly the descriptive complexity of x . In fact, the following theorem holds:

Th. 8 - $L(p^*(x)) = H(x, n)$.

Proof. By induction on the length of x one has:

1) $L(p^*(\sigma)) = L(\sigma) = 1 = H(\sigma, 1)$

2) Let $p^*(x) = p^*(ayb) \sqcap^*(s, m)$; then $L(p^*(x)) = L(p^*(ayb)) + 1$.

Now, taken any processor p such that $W(p) = x$, p is of the form $p = p' \sqcap^*(s', m')$ or $p = p' \sigma^*$; in both cases there exists a processor p'' of ayb such that

$L(p'') \leq L(p')$, since y is the maximum suffix of x satisfying A.

Then, by the induction hypothesis and Th. 7, one has:

$$L(p) = 1 + L(p') \geq 1 + L(p'') \geq 1 + L(p^*(ayb)) = L(p^*(x)).$$

If, on the contrary, $p^*(x) = p^*(z)\sigma^*$, then the letter σ does not appear in z ; this means that every other processor p of x is of the form $p = p'\sigma^*$, therefore, by the induction hypothesis:

$$L(p) = L(p') + 1 \geq L(p^*(z)) + 1 = L(p^*(z\sigma^*)) = L(p^*(x)). \text{ And from this the thesis.}$$

To calculate $H(x, n)$ it is then enough to construct $p^*(x)$ and compute its length.

Let us now detail an algorithm that performs such computations.

Let us define, first of all, the application $T: \Sigma^+ \rightarrow \Sigma^*$ given by:

$$T(x) = \begin{cases} ayb & \text{if } x \text{ admits the decomposition A)} \\ z & \text{otherwise, that is considering the decomposition B).} \end{cases}$$

Furthermore, given a sequence x , we shall say that a sequence $y \neq x$ is a pre-suffix of x if it is at the same time a suffix and a prefix of x . Now we can give the desired algorithm in a Algol like language:

ALG 1

Input: sequence $x = x_1 \dots x_n$; integer $n \geq 1$.

Output: integer $H(x, n)$.

```
begin
    H ← 0 ;
    h ← n ;
    repeat ALG2 until T(x) = A ;
    H(x, n) ← H
end
```

Where ALG2 is simply the serial execution of two others algorithms.

ALG 2

Input: sequence $x \equiv x_1 \dots x_n$

Output: $T(x)$

begin

ALG 3

ALG 4

end

In these algorithms, ϕ_i^j ($1 \leq i \leq h$) is the length of the longest presuffix of the sequence $x_{h-i+1} \dots x_h$, since ALG 3 transposes the input sequence before executing any other manipulations.

ALG 3 is intended to determine ϕ_j^j for every j , and is a version of an analogous algorithm presented in Aho et al. (1974), par. 9.3.

ALG 3

Input: sequence $x \equiv x_1 \dots x_h$; integer $h \geq 1$.

Output: vector $\phi \equiv (\phi_1, \dots, \phi_h)$

begin

$y \leftarrow x^T$;

$i \leftarrow 0$;

for $j \leftarrow 1$ until h do $\phi_j \leftarrow 0$;

for $k \leftarrow 2$ until h do

while $y_{i+1} \neq y_k$ and $i \neq 0$ do $i \leftarrow \phi_i$

if $y_{i+1} = y_k$ then

begin

$\phi_k \leftarrow i + 1$;

$i \leftarrow i + 1$;

end

end

ALG4

Input: sequence $\beta = \beta_1, \dots, \beta_h$; integer $h \geq 1$.

Output: integer H ; sequence $x = x_1, \dots, x_h$; integer h .

```

begin
    for  $k \leftarrow h$  step -1 until 2 do
        while  $\beta_k > k/2$  do
            begin
                 $i \leftarrow \beta_k$  ;  $\beta_k \leftarrow \beta_i$  ;  $\beta_i \leftarrow 0$  ;
            end
             $p \leftarrow \text{Max}(1, \text{Max}(\beta_1, \dots, \beta_h))$  ;
             $h \leftarrow h-p$  ;
             $T(x) = x_1 \dots x_h$  ;
             $H \leftarrow H+1$  ;
             $x \leftarrow T(x)$  ;
    end

```

ALG4 must verify that the presuffixes do not overlap and choose the maximum length of the presuffixes which satisfy this requisite.

Let us now proceed to an analysis of the program to evaluate its time complexity.

ALG3, as shown by Aho et al. (1974, Th.9.7), requires $O(h)$ steps of calculation; for ALG4, $h-2$ steps are necessary if $\beta_k \leq k/2$ for every k , otherwise h other steps at most are needed to set all the β_i 's to 0; therefore ALG4 also requires $O(h)$ steps.

Let now $t(x)$ be the number of steps necessary to evaluate $H(x,n)$ following ALG1, and let $\tau(n) = \text{Max}\{t(x)/l(x) = n\}$. Considering the program given for ALG1, we see that ALG2 is executed m times, where $m = H(x,n)$. $T(x)$ is determined by the sequence of the two algorithms ALG3 and ALG4 which, as we have seen, require $O(l(x))$ steps, x being the input sequence to ALG2.

Since these sequences have successively decreasing lengths, starting from the value n of the length of the assigned sequence x , one will have, for a suitable constant g ,

$$t(x) \leq g \cdot n \cdot h(x, n)$$

But if the sequence x is assumed to be the one described in the proof of Th.4, evidently one will have, for a suitable constant $g' > 0$,

$$\therefore t(x) \geq g' \cdot n \cdot \frac{n}{\log_2 n}$$

therefore concluding that $\mathcal{T}(n) = O(n^2 / \log_2 n)$.

THE OPERATORS

A summary analysis of a few musical texts and some considerations derived from composition practice (Schoenberg, 1954) have enabled us to identify a set of operators related to some common musical functions. By this set of operators it is possible therefore to analyse a musical text and identify in it the musical functions the operators represent; in this way it is also possible to give a compact description of the analysed text considering a sequence of notes as represented by the operator alone that describes it as a function of the preceding text.

As a first step, musical processes are represented by sequences of numerical symbols (Haus 1976); for example, it is possible to take as alphabet the set of the first seven integer numbers and put them into correspondence with the seven degrees of a major scale, or take the set of the first twelve integer numbers and put them into correspondence with the twelve semitones of an octave; likewise it is possible to code the other parameters which describe the sound (duration, width, pitch, timbre etc.).

The different operators that correspond to musical functions can be grouped into distinct classes; the operators that we have utilized are referred to two of these classes.

The class of operators on values has the function of altering the values of the parameters of a sequence; the class of operators on order has the function of altering the order of the parameters in a sequence.

Associated with every parameter of sound there is the language of the sequences of values of the parameter with which will be associated a suitable set of

operators, constituted at first by the operators more frequently employed according to the indications of composition practice, later constituted by the most effective operators sorted out by means of the same musicological analysis itself we are making.

To the class of operators on values belong the operators that correspond to the algebraic operations of addition, subtraction, multiplication, division, exponentiation etc. These operators take a different name according to the parameters they refer to; for example, addition is called transposition for the pitch, crescendo for the width etc.

To the class of operators on order belong the operators which correspond to particular types of permutations (for example, reversing a parameter sequence).

The operators that we have utilized for our analysis are:

- a) Transposition: it uniformly changes the parameter values; the operator of repetition coincides with the operator of transposition of zero degree; it is used for musical functions like repetitions, progressions, modulations.

EXAMPLE

The operator of TRANSPOSITION of 7th degree applied to the sequence of pitches S gives the new sequence $T_7(S)$.

- b) Specular inversion: it changes the parameter values symmetrically with respect to a certain value or degree; applied to the pitch sequences it fulfills the function of ^{inverting} all the harmonic intervals of the original sequence; this function has great importance in tonal music, in dodecaphonic music and particularly in polyphonic music.

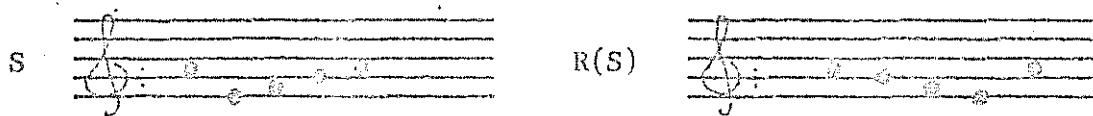
EXAMPLE

The operator of SPECULAR INVERSION of 5th degree applied to the sequence of pitches S gives the new sequence $I_5(S)$.

c) Retrogradation: it reverses the sequences of parameter values; applied to the pitch sequences it has an importance similar to that of the operator of specular inversion; applied to durations it is utilized in all types of music.

EXAMPLE

The operator of RETROGRADATION applied to the sequence of pitches S gives the new sequence R(S).



The operators of transposition and of specular inversion belong to the first class, the operator of retrogradation to the second.

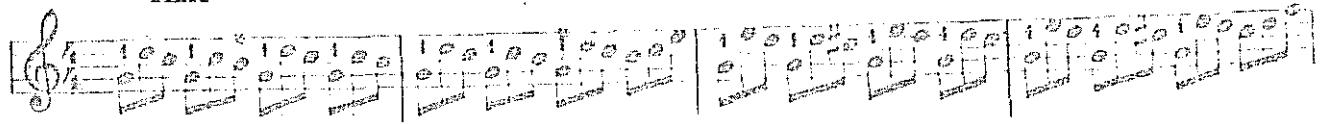
Combinations of these operators have also been utilized (for example the retrogradation of the specular inversion); these combinations of operators fulfil functions of higher complexity. It is easy to show that the possible combinations of these operators are a finite number, precisely 48.

The operators obtained by composition of simple operators permit further compaction of the sequences.

It is understood that all the functions represented by these operators transform sequences of symbols of a given alphabet into sequences on the same alphabet, taking, where necessary, the modulo with respect to the cardinality of the alphabet itself.

For example, we analyze a short sequence of pitches, the first six bars from Czerny's Studio 849 No. 6 right hand, by the operators of transposition, specular inversion, retrogradation, and their combinations.

TEXT



SEQUENCE

7 2 0 7 0 11 7 2 0 7 0 11 7 2 0 7 0 11 7 2 0 11 0 4 7 4 2 7 2 1 7 4 2 7 2
7 4 2 7 2 1 7 4 2 1 2 5 7 5 4 7 2 0 7 0 11 7 2 5 7 5 4 7 2 0 7 0 11 7 2 5

PROCESSOR

7 $T_7(1,1)$ $T_{10}(2,2)$ $I_1(1,2)$ $T_{11}(5,5)$ $T_0(1,6)$ $T_0(1,8)$ $T_0(17,18)$ $R \circ T_5(18,$
 $T_3(24,24)$ $T_9(25,25)$ $T_7(16,17)$ $T_2(21,22)$ $T_0(25,30)$ $T_0(25,31)$ $T_2(20,22)$ $T_{10}(24,$
 $T_3(44,46)$ $T_0(13,19)$ $T_{10}(51,52)$ $T_0(49,60)$

The analyzed text gives one of the graphs in figure 1, where a horizontal step represents an element of the processor above.

We report, as an example, two graphs obtained by analysing some short sequences relative to musical piece; in particular, in the first there are pitch sequences and the whole set of operators has been used, while in the second there are width sequences and only the operators of repetition and retrogradation have been utilized.

PERSPECTIVES

The concepts we have introduced can be extended to parallel and interactive processes such as musical processes actually are. We have implemented some programs which give significant evaluations on the harmonic development of musical parts (up to 4) described by several sequences of height-duration pairs.

In a more general view, the present work and its developments aim to contribute to the following aspects of musical problems:

- a) Storing: musical texts (as well as other data structures) can be compacted and stored with remarkable saving of storage, provided that one identifies, as we have partly done in this work, some suitable operators.
- b) Musicology: the musicological analysis of musical texts becomes extremely easier using procedures of this type, as much as one can associate a particular operator with the characteristic under study.

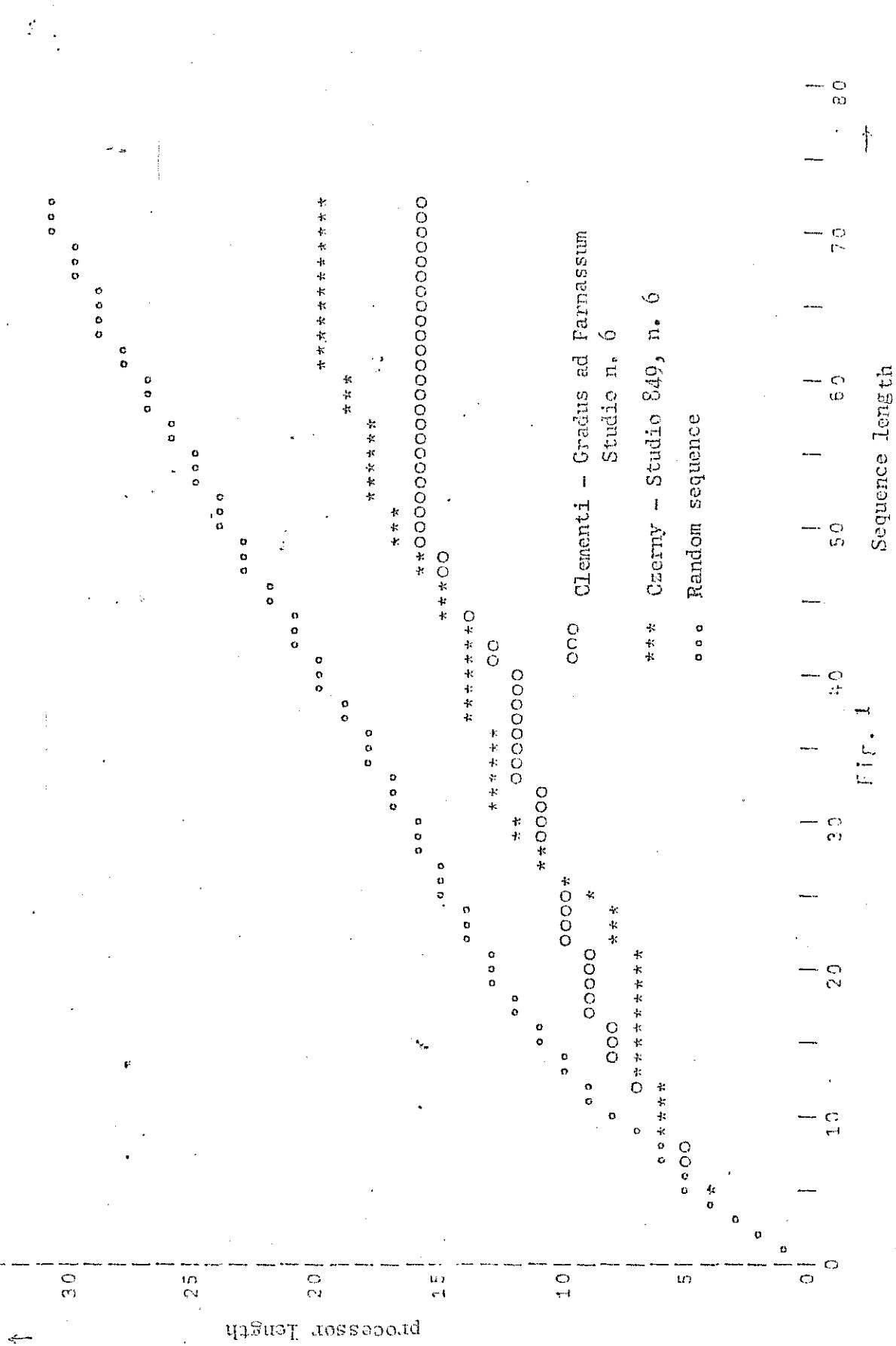
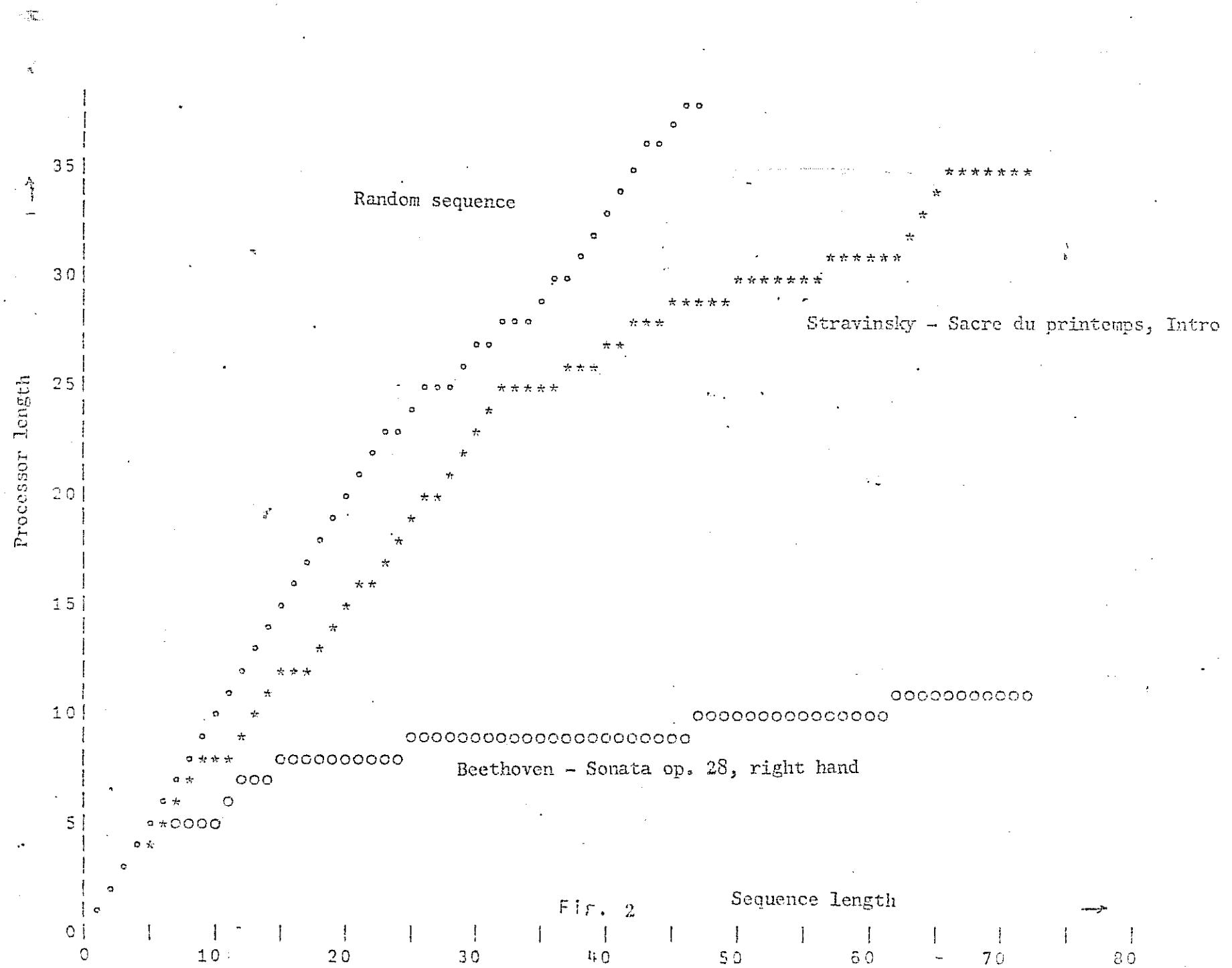


Fig. 1

Sequence Length



c) Text composition: the instruments of description which have been formalized in the present paper induce the definition of a musical language structurally organized and particularly suited for the purpose of a satisfactory automatic composition of musical texts.

* Sincere thanks are due Prof. Giovanni Begli Antoni, Institute of Cybernetics, for his continuous encouragement and suggestions.

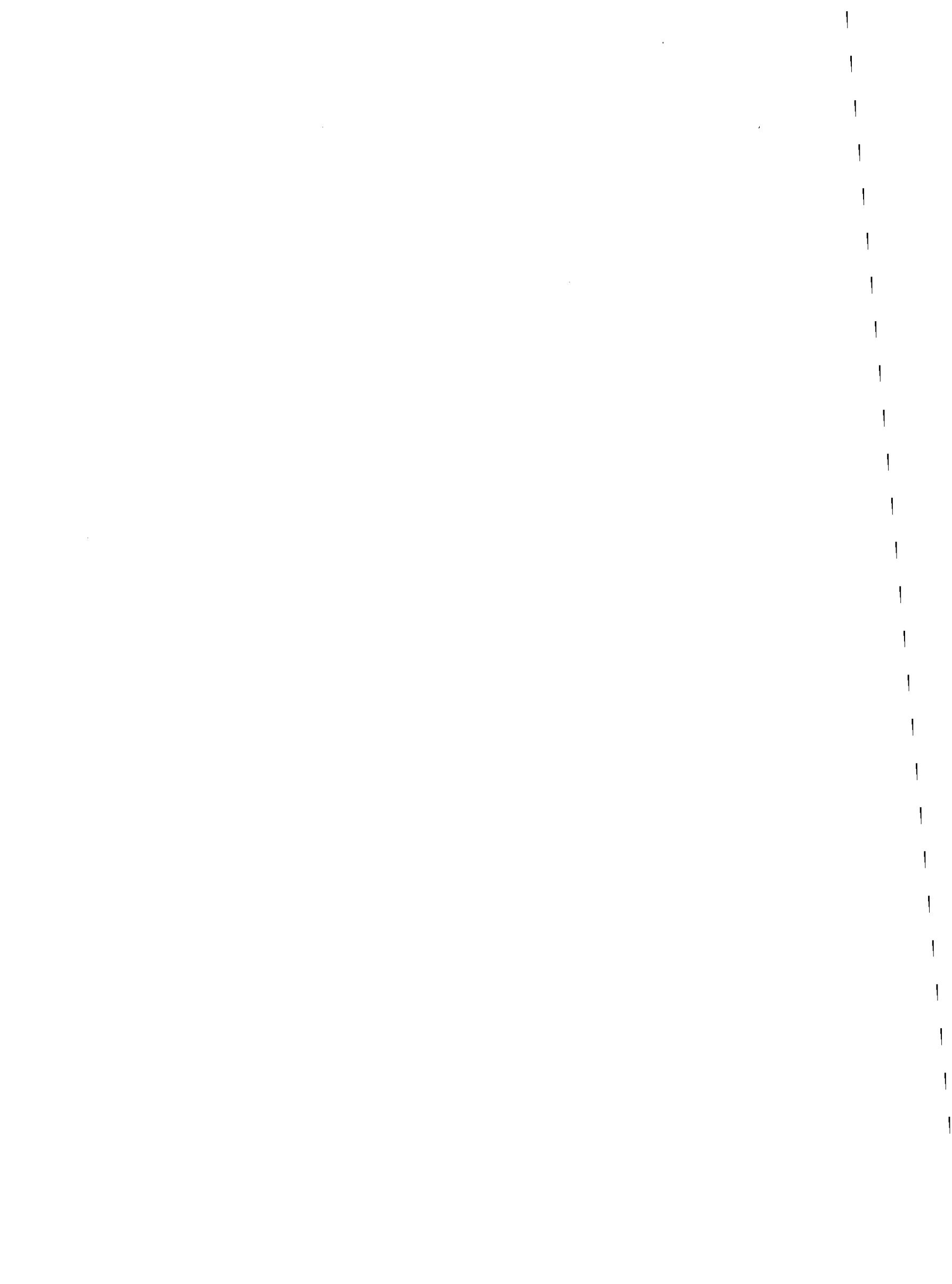
** This research has been developed in the frame of the Communication and Programming Project of the University of Milan and Honeywell Information Systems Italia.

REFERENCES

- Aho, A.V.-Hooperst, J.E.-Ullman, J.B. : "The Design and Analysis of Algorithms" - 1974 - Addison-Wesley - Reading Massachussets
- Coguen, J.A. : "Complexity of Hierarchically Organized Systems and the Structure of Musical Experience" - 1975 - UCLA Computer Science Department Quarterly - Vol. 3 - n. 4 - p. 51-88
- Haus, G.M. : "Descrizione formale di processi musicali" - Thesis - 1976 - University of Milan
- Kolmogorov, A.N. : "Logical Basis for Information and Probability Theory" - 1968 - IEEE Transactions on Information Theory - IT14 - n. 5
- Laske, O. : "Toward a Theory of Musical Cognition" - 1975 - Interface - Vol. 4 - p. 147-208
- Schoenberg, A. : "Structural Functions of Harmony" - 1954 - Ernest Benn - Wien

A.Bertoni, G.Haus, G.Mauri, M.Torelli
Università di Milano - Istituto di Cibernetica
Via Vietti 5

I 20133 MILANO
(Italy)



GIORGIO DE POLI - ALVISE VIDOLIN

"EMUS: UN PROGRAMMA PER L'ELABORAZIONE
DI STRUTTURE MUSICALI"⁴

EMUS: un programma per l'elaborazione di strutture musicali

Introduzione

La composizione di opere musicali che normalmente avviene con metodologie di tipo artigianale, da alcuni anni può essere realizzata avvalendosi dell'elaboratore elettronico numerico. Tale mezzo, pur inizialmente pensato per svolgere calcoli matematici, si è rivelato efficace nel trattamento di informazioni simboliche. In base alle istruzioni che vengono fornite mediante il programma, l'elaboratore può svolgere i compiti più diversi. È stato quindi utilizzato anche nel campo musicale in particolare come aiuto alla composizione di opere musicali, per l'analisi musicologica e per la sintesi del suono.

Quando viene usato per comporre musica, la partitura che ne deriva può essere eseguita da strumenti tradizionali oppure da strumenti elettronici e anche direttamente da l'elaboratore stesso.

In questo lavoro ci occuperemo dell'impiego dell'elaboratore come aiuto alla composizione di opere musicali.

La composizione consiste nel disporre suoni nel tempo in base a determinate intenzioni espressive o formali. Si possono distinguere due aspetti: la struttura generale per analizzare le caratteristiche globali e la disposizione topologica degli eventi; e le relazioni fra i suoni per evidenziare l'aspetto puntuale e le caratteristiche sonologiche. Possiamo pensare la struttura generale di un'opera musicale come un insieme di strutture formali in stretto rapporto con l'insieme delle relazioni fra i suoni impiegati, per cui, fissata la struttura generale, vengono fissate inequivocabilmente anche le relazioni fra i suoni e viceversa.

Si può considerare che in alcuni casi il compositore parte dall'insieme delle relazioni per determinare la struttura generale dell'opera, in altri agisce nella maniera opposta oppure, come spesso accade, usa alternativamente entrambe i metodi. In ogni caso possiamo dire che la composizione di un'opera musicale avviene tramite la elaborazione di strutture musicali intendendo per quest'ultime sia le strutture formali che le relazioni fra i suoni.

Tali strutture musicali vengono sviluppate dal compositore in base alla sua creatività artistica ed alle sue conoscenze tecniche. La prima è la parte meno formalizzabile nel procedimento compositivo in quanto attiene alla sfera inventiva.

* Università di Padova - Istituto di Elettronica

** Conservatorio "B. Marcello" - Venezia

va dell'attività umana. Le conoscenze tecniche, invece, si possono esprimere mediante regole musicali più o meno esplicite che dipendono da fattori culturali, da scelte stilistiche, dalla realtà fisica, etc.; queste regole musicali possono essere condivise da più persone e costituire il substrato comune a più opere.

Nell'elaboratore si può inserire quella parte di conoscenze tecniche che può essere formalizzata e resa esplicita. Val la pena osservare che le regole possono essere relative a dei particolari molto dettagliati nella composizione come pure molto generali, tali da condizionare regole a livelli gerarchici inferiori.

Le attività che portano alla realizzazione globale di una opera di musica si possono schematizzare in vari modi. Se si impiega l'elaboratore queste possono essere divise in quattro funzioni base anche se non sempre sono nettamente separate fra loro. Si individuano così la funzione di generazione di un materiale simbolico di base, di elaborazione di tale materiale, di interpretazione del linguaggio simbolico e di produzione sonora vera e propria.

Un programma generale per la composizione può essere scomposto in queste quattro funzioni come mostrato in fig. 1. In tale schema sono indicati gli ingressi ed evidenziati i risultati di ogni parte. Questo è uno schema teorico del tutto generale a cui possono ricondursi i vari modi di utilizzare l'elaboratore per la produzione di opere musicali.

Spesso l'organizzazione del programma è fatta in modo che le varie funzioni non appaiano all'esterno così distinte e i risultati intermedi non siano accessibili. In certi casi i dati di ingresso possono essere determinati o dalle fasi precedenti o da scelte insite nel programma. Altre volte alcune funzioni sono effettuate senza l'aiuto dell'elaboratore essendo il musicista ad utilizzare o fornire i risultati intermedi.

Ad esempio, considerando alcuni dei più caratteristici programmi di composizione si osserva che quelli che impiegano metodi generativi grammaticali (come Lindblom-Sundberg; Lidov-Gabura; etc.) consistono essenzialmente in un modello di generazione della partitura simbolica praticamente senza elaborazione. Il Project 2 di G. M. Koenig utilizza materiale base definito espressamente dal musicista e lo organizza nella partitura simbolica sfruttando così ampiamente la funzione di elaborazione. Il programma Score di L. Smith sviluppa con particolare cura la funzione interpretativa mentre il Music5 di M. Mathews quella di produzione del suono.

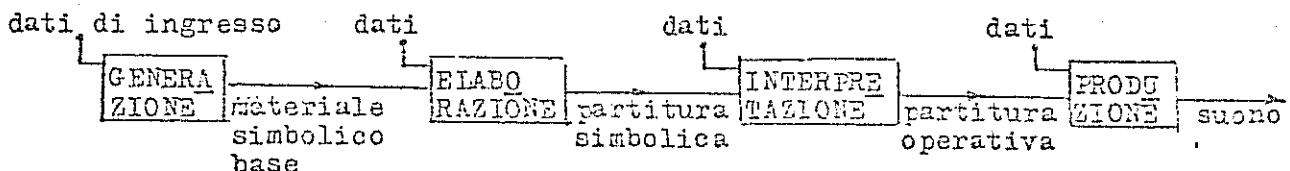


fig. 1

-Caratteristiche generali

Abbiamo visto che nella composizione di opere musicali ha una forte influenza la componente creativa che, non solo permette la realizzazione di opere sempre nuove in dato sistema di regole, ma rinnova tali regole permettendo l'evoluzione del pensiero musicale. Si ha così un sistema in continua evoluzione ed estremamente composito. E' perciò impossibile un approccio sistematico che pretenda di risolvere il problema dell'impiego dell'elaboratore nella composizione musicale in modo generale e definitivo.

allo stato attuale delle conoscenze non è pensabile la realizzazione di un programma per la composizione di uso completamente generale; è utile invece realizzare programmi più limitati che affrontino il problema da punti di vista diversi. I risultati di questi lavori ci forniranno una visione sempre più completa del problema della composizione musicale come pure costituiranno utili strumenti di indagine per la comprensione dei processi creativi umani.

In questa visione si è ritenuto opportuno elaborare un programma, denominato EMUS, che, oltre a portare un contributo alla ricerca più generale, permetta ai musicisti la produzione di opere musicali avvalendosi dell'aiuto dello elaboratore.

Vediamo alcune caratteristiche di questo programma. Il musicista è in grado di realizzare una composizione di tipo deterministico, in cui fissa la struttura generale dell'opera fino al particolare (singolo suono), come pure può lasciare alcuni processi o alcuni parametri del tutto aleatori o parzialmente controllati.

Bisogna precisare che ogni evento sonoro viene scomposto in una serie di parametri che ne definiscono le caratteristiche. Questi parametri non sono necessariamente i parametri fisici del suono, ma essendo espressi mediante simboli possono rappresentare entità musicali. Il compositore definisce separatamente l'evoluzione dei singoli parametri e solo in un secondo momento, con una operazione di aggregazione viene fissata la successione nel tempo degli eventi sonori.

Si è posta particolare attenzione nel fissare un linguaggio il più semplice ed intuitivo possibile per favorire l'interazione musicista-elaboratore. Si è evitato perciò l'impiego di soli numeri in favore di definizioni simboliche alfabetiche o grafiche. Inoltre per soddisfare alle diverse esigenze ogni compositore può aggiungere le proprie regole di definizione ed interpretazione dei simboli.

Nel programma si possono distinguere le funzioni indicate nello schema funzionale di fig. 1: generazione, elaborazione, interpretazione.

La generazione prevede la creazione del materiale simbolico di base e può essere fatta direttamente dal musicista con

una operazione di definizione oppure da un processo generativo totalmente automatico o controllabile parametricamente. Questo materiale può essere elaborato per mezzo di operazioni di selezione grafica o numerica generando del nuovo materiale base.

L'organizzazione di questo materiale base avviene per mezzo della struttura intermedia che permette la aggregazione dei simboli prima definiti in modo da costituire una rappresentazione di sequenze di eventi sonori. Queste sequenze sono gli elementi che costituiscono la struttura intermedia e vengono ordinate dal musicista in vista della loro utilizzazione definitiva. Inoltre possono essere disposte in linee diverse ottenendo così un numero di voci, inteso in senso tradizionale, non limitato.

L'organizzazione finale si ottiene scandendo in modo opportuno le sequenze della struttura intermedia e disponendole nel tempo con diverse possibilità di sincronismo fra le varie linee.

L'interpretazione consiste nell'esplicitazione dei simboli che definiscono gli eventi musicali secondo regole prefissate in generale o definibili di volta in volta dal musicista. Il significato di ogni simbolo può dipendere dal complesso dei simboli e quindi non è frutto di una semplice operazione di traduzione bensì dipende dal contesto. E' interessante notare che in questa fase è possibile apportare correzioni automatiche o segnalazioni di errore in base a regole di compatibilità fissate in precedenza dal compositore.

La partitura operativa così ottenuta viene eseguita da un qualsiasi programma o dispositivo per la sintesi del suono.

Queste funzioni non sono sempre necessariamente susseguenti nel tempo, ma è possibile, una volta definita la struttura intermedia, definire o ridefinire del materiale base, variare la stessa, generare la partitura operativa e ascoltare il risultato acustico in qualsiasi successione sfruttando così convenientemente i risultati intermedi. Il musicista è messo in condizioni di poter espletare le proprie scelte e il controllo di ogni singola funzione con particolare riferimento a quella di elaborazione. E' questa, infatti, la fase in cui prende forma l'opera nella sua consistenza globale ed in cui più intervengono quelle componenti non formalizzabili delle regole musicali.

Con questo programma non si intende sostituire l'uomo con la macchina, ma si vuole fornire al musicista un valido aiuto nella composizione. Non è pensabile, comunque, che sia la macchina a risolvere problemi musicali che non siano stati precedentemente risolti dal musicista.

In conclusione è ancora il compositore l'unico responsabile del prodotto finale.

- Caratteristiche operative

Il programma EMUS genera la partitura operativa seguendo le istruzioni che vengono fornite dal compositore. Queste istruzioni controllano le operazioni di creazione del materiale base e di organizzazione dello stesso nel tempo. Ogni istruzione è caratterizzata da un codice operativo, da un identificatore, fissato a piacere dall'utilizzatore e dalla serie di operandi richiesta dal tipo di istruzione.

I parametri in cui sono scomponibili gli eventi sonori, come ad esempio, altezza, dinamica, densità, tessitura, etc., variano di valore nel susseguirsi degli eventi. Il materiale base è costituito dall'evoluzione dei parametri fondamentali scelti dal compositore e può essere determinato con vari tipi di istruzione. Il musicista può definire direttamente le sequenze di simboli che rappresentano l'evoluzione di uno o più parametri e avere così il controllo puntuale dell'andamento della successione e delle relazioni fra gli eventi sonori.

Molto spesso il musicista non intende definire in modo rigoroso l'evoluzione di tali parametri, ma preferisce sfruttare la generazione automatica nei vari gradi: da generazione totalmente aleatoria a generazione mediante regole. Questa tecnica può essere applicata sia a tutti i parametri, e quindi avere la generazione totale del materiale base come pure è possibile definire in modo rigoroso alcuni parametri e usare la generazione automatica per i rimanenti. Inoltre non è necessario avere definiti nel materiale base tutti i parametri ritenuti fondamentali dal musicista. Alcuni di essi possono essere ricavati direttamente dal valore di altri o dal contesto mediante regole e senza ulteriori specificazioni. Tutto ciò può realizzarsi nella funzione di interpretazione che prevede l'introduzione delle regole personali richieste dal compositore. In questo modo non si ha solo l'interpretazione dei simboli ma anche la generazione dei parametri secondo regole contestuali.

Per la creazione del materiale base è stata realizzata una serie di istruzioni che può essere facilmente variata o incrementata in base alle esigenze. Esiste una istruzione caratterizzata dal codice operativo IST che permette la specificazione di una successione di simboli alfanumerici. Tale istruzione può essere impiegata per la definizione di qualsiasi parametro, sia simbolico che numerico. Il significato dei simboli impiegati viene determinato solo dalla funzione di interpretazione. Si possono così utilizzare gli stessi simboli per parametri diversi.

Accanto a questa istruzione di uso totalmente generale ne sono previste altre in cui il codice operativo definisce direttamente la regola di interpretazione.

L'istruzione FRZ prevede la specificazione di una sequenza di numeri espressi come rapporto e può essere direttamente impiegata per la definizione delle durate metriche o di intervalli di frequenza. L'istruzione ALT permette la definizione di una successione di altezze della scala cromatica con la notazione adottata nel linguaggio MUSICA .

Accanto a queste istruzioni che permettono una definizione deterministica del materiale base ve ne sono altre che impiegano metodi di generazione e di selezione da un repertorio. Possono essere utilizzate sia per la creazione del materiale base sia per la generazione di simboli appartenenti ad eventi con una precisa collocazione temporale. Nel primo caso si ottiene una successione di simboli indipendenti dal tempo che costituiscono una sequenza determinata una volta per tutte. Nel secondo caso si fissa una legge che determina di volta in volta la sequenza in dipendenza dagli istanti di inizio degli eventi sonori. Questa operazione di generazione può essere effettuata solo quando vengono aggregati i simboli nella struttura intermedia e verrà , quindi, esaminata più avanti.

L'istruzione RNF genera una successione di numeri aleatori compresi fra un massimo ed un minimo. Può essere utilizzata ad esempio, per la generazione di successioni di durate metriche, di frequenze espresse in Hz, etc. L'istruzione RNS, invece, crea una successione di simboli mediante selezione aleatoria da un'altra successione fissata ad esempio con una istruzione LST . Partendo quindi da un repertorio di simboli è possibile ottenere un nuovo ordinamento di una parte o di tutti i simboli del repertorio.

Oltre che con un criterio aleatorio è possibile operare una selezione con un criterio di scelta espresso graficamente. E' necessario innanzitutto definire una funzione grafica per mezzo delle due istruzioni FXY ed FEX. La prima esegue l'interpolazione lineare fra coppie di punti di un piano cartesiano e la seconda disegna un esponenziale passante per due estremi. Assumendo poi come riferimento una successione numerica, in corrispondenza di ogni elemento di questa successione viene determinato il valore della funzione grafica.

Mediante l'istruzione FNV si ottiene la sequenza di valori assunti dalla funzione e riscalati fra un massimo ed un minimo. Con l'istruzione FNS i valori della funzione servono per operare una scelta nel repertorio definito come successione di simboli.

Infine l'istruzione VAR permette di trasformare una sequenza di simboli mediante regole diverse, distinte da un codice operativo. Accanto alle regole di trasformazione previste dal programma, quali inversione di una sequenza simbolica; riscalamento, moltiplicazione, trasposizione di una sequenza numerica; selezione di una sottosequenza; etc., è possibile

aggiungere le regole di trasformazione che il compositore desidera.

Abbiamo descritto l'evento sonoro dal punto di vista del musicista, scomponendolo, quindi, nei parametri musicali simbolici. Per la sintesi del suono, invece, ci si avvale di un modello di produzione sonora descritto da un insieme di regole che dipendono da una serie di parametri fisici. Tale modello noi lo chiameremo, in analogia al mezzo di produzione tradizionale, strumento, inteso nel senso più generale del termine. E' lo strumento che determina le caratteristiche sostanziali del suono ed al variare dei suoi parametri produce risultati diversi.

Il compositore sceglierà un certo numero di strumenti che verranno impiegati per l'esecuzione dell'opera. Per ogni strumento è opportuno definire i valori di riferimento dei parametri fisici; questi valori saranno sostituiti o modificati dalla funzione di interpretazione nell'esplicitare il significato dei simboli. La specificazione dello strumento adottato e dei suoi parametri di riferimento costituisce l'evento sonoro base che viene associato ad un nome nella fase di creazione del materiale base.

Definito così il materiale base, una prima organizzazione di questo viene fatta nella struttura intermedia.

Questa struttura può essere divisa in linee: ogni linea è divisibile in celle ed ogni cella è divisibile in righe.

La cella definisce una sequenza di eventi sonori. Tale sequenza è scomponibile nelle sequenze-andamento dei singoli parametri; come abbiamo visto, queste vengono normalmente definite nel materiale base e identificate mediante un nome. E' proprio questo nome che viene introdotto nelle righe della cella. Ci sono vari tipi di righe (R, S, I, C) aventi compiti diversi.

La riga R1 è riservata alla successione degli intervalli temporali fra gli eventi sonori. Essa determina il numero degli eventi sonori contenuti nella cella e viene assunta come riga di riferimento. La riga R2 contiene la successione degli eventi sonori base. Con le righe R1 ed R2 si ha il minimo di informazione necessaria per ottenere una successione di eventi sonori. Per la loro trasformazione si ricorre alle righe simboliche S, I e C.

Le righe S contengono sequenze di simboli esprimenti i parametri musicali. In particolare la riga S0 può essere riservata alla successione di altezze e va ad agire direttamente sul corrispondente parametro di frequenza dello strumento.

Molto spesso è utile associare ad un evento un insieme di parametri raggruppati sotto uno stesso nome. Una sequenza di questi nomi può essere contenuta nelle righe I. Nelle righe C viene introdotto un unico simbolo che fa riferimento a tutti gli eventi della sequenza.

Per riassumere: una cella contiene una sequenza di eventi sonori; ogni evento è composto da una durata metrica indicata in R1; dal tipo di strumento adottato e da una serie di parametri fisici di riferimento, indicati in R2; da uno o più simboli di interpretazione, ciascuno in una riga S; da uno o più gruppi di simboli di interpretazione, ciascuno in una riga I; da uno o più simboli di interpretazione, comuni a tutti gli eventi della sequenza, ciascuno in una riga C.

L'aggregazione dei simboli appartenenti alle varie sequenze, viene effettuata prendendo come riferimento la riga R1. Scandendo ordinatamente la riga R1, al primo elemento viene associato il corrispondente primo simbolo di tutte le altre righe, e così via. Se una sequenza di simboli ha un numero di elementi inferiore al necessario viene ripetuta ciclicamente; viceversa, se superiore, vengono ignorati gli ultimi elementi.

Le celle vengono ordinate numericamente in più linee. Ogni linea, quindi, contiene una successione di eventi sonori ed è caratterizzata dallo stesso ordinamento numerico. Si può far riferimento a celle appartenenti a linee diverse mediante lo stesso numero. Questa separazione in linee è utile sia per la organizzazione del lavoro, in quanto si possono disporre su linee diverse materiali differenti, come per la sincronizzazione delle varie sequenze. In questo ultimo caso, però, le celle devono avere una durata metrica tale da consentire i sincronismi voluti.

La struttura intermedia viene riempita con l'istruzione caratterizzata dal codice operativo FUT. Questa istruzione permette l'organizzazione del materiale base agendo sia sui singoli parametri che sui gruppi di essi. Le sequenze-andamento del materiale base possono essere raggruppate in modo tale da riempire celle successive. Questo raggruppamento viene chiamato Kolon. Anche il Kolon può essere formato da uno o più parametri. Sia Kolon che singole sequenze possono essere introdotti nelle celle più volte e in posizioni diverse.

L'istruzione opera su righe di celle appartenenti ad una o più linee. Vengono anzitutto specificate le linee e le righe interessate all'operazione. Se si indica più di una linea, l'istruzione ~~è~~ in ognuna di queste ed allo stesso modo. Le righe delle celle vengono riempite con i nomi delle sequenze di simboli o con i simboli stessi se si tratta delle righe C. E' quindi specificato nell'istruzione un nome per ogni riga. Più nomi formano un gruppo quando sono inseriti in righe diverse nella stessa cella; formano un Kolon quando vanno inseriti nella stessa riga di celle successive. E' possibile avere anche Kolon di gruppi. Un gruppo o un Kolon possono essere inseriti automaticamente

in celle a distanza definita o a partire da alcune celle definite espressamente. E' possibile inoltre specificare il numero della cella in cui inserire ciascun gruppo di nomi componenti il KOLON.

Le righe C vengono riempite anche dall'istruzione PCx in cui x indica il numero della riga C. In essa vengono indicati le linee ed il nome di una sequenza di simboli. Questi vengono via via assegnati alle celle specificate nella istruzione. Se il nome corrisponde ad una istruzione FNS, viene assunto come riferimento per il grafico la successione delle celle indicate. Se corrisponde all'istruzione RNS viene selezionato aleatoriamente un simbolo per ogni cella.

Mediante l'uso della struttura intermedia si ottiene una prima organizzazione del materiale base. Gli eventi sonori, infatti, sono organizzati in sequenze e le sequenze sono organizzate in linee. Si può sfruttare questa disposizione per evidenziare le relazioni esistenti fra gli eventi e tra i gruppi di eventi.

La disposizione nel tempo delle sequenze di eventi sonori, per preparare la partitura simbolica, avviene tramite la istruzione SCN. Tale istruzione permette di scandire la struttura intermedia nelle celle e nelle linee volute. Ad ogni cella così individuata viene attribuito un tempo iniziale. Si fissa così l'esatta collocazione temporale di ogni evento componente la sequenza in base agli intervalli temporali contenuti nella riga R1 della cella.

Scandendo la struttura intermedia ordinatamente in tutte le sue celle e in tutte le sue linee si ottiene una partitura simbolica che rispecchia fedelmente l'organizzazione della struttura intermedia. Normalmente invece, vengono utilizzate alcune parti in ordine permutato, riorganizzando così la successione delle sequenze.

Potendo specificare in maniera arbitraria il tempo iniziale di ciascuna sequenza, queste si combinano e sovrappongono nella maniera più varia. Per favorire la sincronizzazione delle sequenze, ogni linea ha un proprio tempo di riferimento che viene incrementato automaticamente al procedere delle scansioni. Il tempo iniziale della cella può essere relativo al tempo di riferimento della propria linea o può essere espresso con un valore assoluto rispetto l'inizio dell'opera.

E' possibile sincronizzare i tempi delle varie linee con l'istruzione TMP. Con essa i tempi di riferimento di alcune linee vengono aggiornati ad un valore assoluto oppure al massimo o al minimo dei tempi di altre linee.

Abbiamo visto che fra le istruzioni che generano simboli, alcune definiscono una legge che fissa di volta in volta una sequenza in dipendenza degli istanti di inizio degli eventi sonori. Questa regola viene applicata al momento

della realizzazione della partitura simbolica. Le istruzioni FNS e FNV possono assumere come riferimento del grafico gli istanti iniziali degli eventi sonori della sequenza. Ad esempio con FNS attraverso legge grafica viene selezionato da un repertorio un simbolo in corrispondenza di ogni evento.

Le istruzioni RNV e RNS possono anch'esse agire in questa fase, generando una nuova sequenza ogni volta che sono impiegate.

Esiste poi l'istruzione MSK che opera solo in questa fase. Essa specifica una sequenza maschera di distanze temporali ed una sequenza di simboli da selezionare. Ad ogni intervallo temporale è associato ordinatamente un simbolo della sequenza che viene ripetuta ciclicamente se il numero degli elementi è inferiore al necessario. Assunto lo stesso tempo di inizio per le due sequenze (istanti iniziali e maschera) si determina in quale intervallo della maschera cade l'istante iniziale di ogni evento. Viene così selezionato il simbolo corrispondente.

Come abbiamo già detto in queste istruzioni si fa riferimento alle sequenze attraverso il nome. Ridefinendo quindi una sequenza, viene cambiata automaticamente l'azione delle istruzioni che la impiegano in questa fase.

Con l'istruzione SCN si è preparata la partitura simbolica. In essa compaiono, per ogni evento, l'istante di inizio, l'evento sonoro base, nonché i simboli o gruppi di simboli relativi ai parametri musicali. Per ottenere la partitura operativa deve venir interpretato il complesso di questi simboli e tradotto nei parametri fisici specifici dello strumento che produrrà il suono. Questa operazione dipende strettamente dalle caratteristiche degli strumenti previsti nonché dalle scelte musicali effettuate dal compositore.

Non è possibile fissare delle regole generali di interpretazione che soddisfino a tutte le possibili esigenze. Accanto ad alcune operazioni più frequenti ve ne sono altre le cui regole variano di volta in volta. Si è ritenuto opportuno, quindi, che l'utilizzatore possa variare o aggiungere le proprie regole di interpretazione con appositi sottoprogrammi.

Conclusione

Il programma qui descritto non ha avuto ancora un impiego tale da trarre delle conclusioni generali e sufficientemente rappresentative. Dai primi esperimenti, comunque, si è ri-

velata particolarmente utile l'idea di mantenere tutta l'informazione musicale a livello simbolico. In questo modo nella fase di organizzazione si è completamente svincolati dalle caratteristiche degli strumenti, che possono usare algoritmi di sintesi completamente differenti, a favore di un'elaborazione strutturale dell'opera musicale a livelli sempre superiori. Non si perde tuttavia la possibilità di comporre il suono; anch'esso viene composto facendo riferimento più all'aspetto musicale che alla tecnica di produzione.

La struttura intermedia, già sperimentata nel programma CELLE con risultati positivi, è stata qui ampliata ed arricchita di nuove possibilità. Anche se il musicista è costretto ad una fase preliminare di organizzazione del materiale base, tutto il lavoro viene notevolmente facilitata in quanto si procede con ordine e con una chiara visione generale della struttura. Tale rigore, comunque, non limita le applicazioni ed anzi da un ampio spazio alla sperimentazione a tutti i livelli: dalla struttura generale al singolo suono.

Può sembrare scomodo aver lasciato al singolo utente la possibilità di ampliare la funzione di interpretazione mediante un sottoprogramma, in quanto raramente il compositore sa programmare. Tale scelta è stata imposta dalle caratteristiche di generalità volute e non si è rivela~~ta~~ta particolarmente gravosa in quanto esiste una collaborazione sempre più stretta fra il tecnico ed il musicista soprattutto nel campo della computer-music.

In conclusione si ritiene che il programma così impostato possa favorire la composizione di opere ed un'estesa sperimentazione di nuove forme di espressione musicale.

Bibliografia

- B. LINDBLOM, J. SUNDRERG, "Music Composed by a Computer Program", Speech Transmission Laboratory-Quarterly Progress and Status Report, n° 4, 1972, p. 20-28
- D. LIDOV, J. GABURA, "A Melody Writing Algorithm Using a Formal Language Model", Comp. Stud. Hum. & Verbal Behaviour, vol. 4, n° 3-4, 1973, p. 138-148
- G.M. KOENIG, "PROJECT 2: Computer Programme for the Calculation of Musical Structure Variants", Electronic Music Reports, n° 3, 1970
- L. SMITH, "SCORE: A Musician's Approach to Computer Music", Journal of Audio Eng. Soc., vol. 20 n° 1, Jan-Feb 1972

- M.V. MATHEWS, "The Technology of Computer Music", MIT press.
1969
- G. DE POLI, A. VIDOLIN, "Computer Music: proposte per una impostazione procedurale", Atti del III e IV Seminario di Studi e Ricerche sul Linguaggio Musicale, Vicenza 1975, p. 132-133
- G. DE POLI, A. VIDOLIN, "CELLE: un programma per l'elaborazione di strutture ritmiche musicali", Atti del VI Seminario di Studi e Ricerche sul Linguaggio Musicale, Vicenza 1976 (in corso di pubblicazione)

BOLOGNESI - CNRCE DI PISA

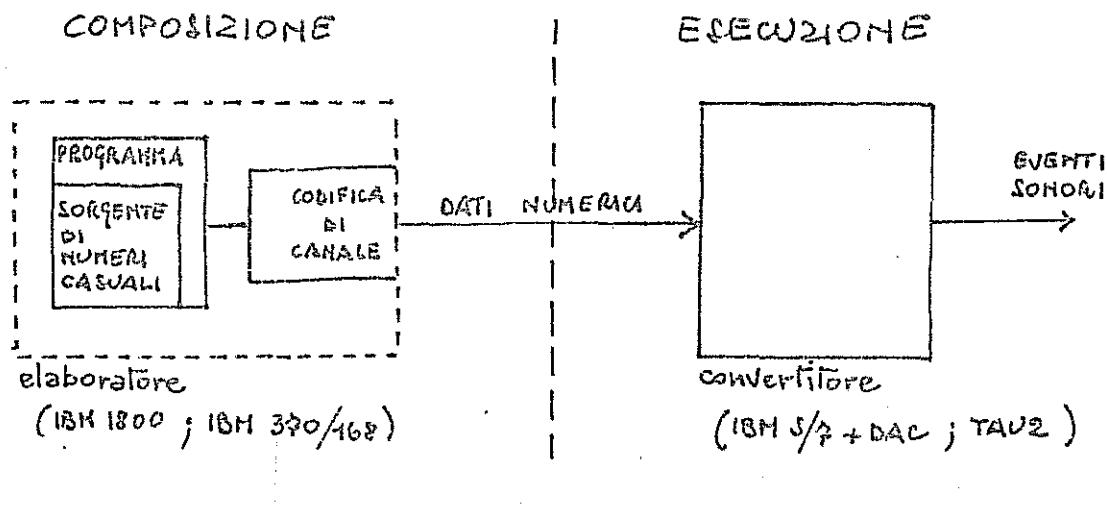
"AUTOMAZIONE DI PROCESSI COMPOSITIVI:
CRITERI DI UTILIZZO DELLA SORGENTE
STOCASTICA"

Indice

1. Introduzione	1
Schema del sistema di composizione (hardware e software)	2
Principi cui i programmi si sono ispirati	2
a) esecuzione <u>senza limiti di tempo</u>	2
b) preponderante utilizzo dell'elemento <u>casualità</u>	2
c) ciò che non può essere ottenuto se non sfruttando la macchina	2
d) possibilità di <u>intervento umano a diversi livelli</u>	2
2. Caratteristiche dei programmi di composizione	3
Tabella dei programmi già esistenti , suddivisi in <u>orientabili</u> e non <u>orientabili</u>	3
Un esempio di programma <u>orientabile</u> : l'ALGOR-DCMP	3
- le opzioni	3
- osservazioni	3
Un esempio di programma <u>non orientabile</u> : il CREATE-DCMP	3
- Play	3
- Options	3
- Elaboration	3
Alcune osservazioni sugli altri programmi:	3
FLAY 1800	3
CREATE TAUMUS	3
RANDOM TAUMUS (il tempo reale)	3
Osservazioni generali	3
3. Possibili linee di sviluppo per i programmi di composizione automatica	15
1) Prima osservazione	15
2) Varietà oggettiva e varietà soggettiva	15
3) Due modi di limitare la casualità (invarianza e non invarianza per inversione temporale)	16
4) Pregi e limiti della sorgente puramente probabilistica	18
Tre esigenze: a) rappresentazione riassuntiva del passato	18
b) memoria di particolari eventi del passato remoto	20
c) le attese create dalle lacune strutturali.	20
5) Criteri di utilizzo della sorgente casuale	20
6) Problemi relativi al progetto di una nuova sorgente	22
- Memoria	22
- Algoritmi = I approssimazione	23
II "	23
III "	24
IV "	24
- Sorgente casuale	28
Bibliografia	29

1. INTRODUZIONE

Una grossa parte delle attività di ricerca e delle realizzazioni effettuate a partire circa dal 1970 a Pisa riguarda l'automazione di processi di composizione musicale; la creazione cioè di programmi che generano sequenze numeriche da convertire in sequenze di eventi sonori.



-fig. 1-

L'intero sistema di produzione musicale può essere schematizzato come in fig. 1.

L'hardware è costituito da un elaboratore (IBM 1800, IBM 370/168) dove avviene la composizione delle sequenze numeriche, e da un convertitore (IBM S/7 + DAC; TAU 2) nel quale comprendiamo anche il sistema di amplificazione e diffusione, dove avviene l'interpretazione, o, si potrebbe dire, l'esecuzione delle sequenze stesse che vengono convertite in sequenze di segnali acustici, o eventi sonori.

Il software è costituito dal programma (PLAY 1800; ALGOR e CREATE del DCMP; CREATE e RANDOM del TAUMUS) che compone le sequenze numeriche secondo una codifica che diremo interna, e da un programma che realizza una seconda codifica, che diremo canale, orientata alle caratteristiche tecniche del convertitore, e che fornisce le sequenze numeriche che pilotano quest'ultimo.

Non ci occuperemo dell'hardware e neppure ci occuperemo, per quanto riguarda il software, del problema della codifica di canale; tratteremo invece delle caratteristiche generali e delle più specifiche caratteristiche funzionali dei programmi di composizione, dei criteri che sottostanno ai procedimenti compositivi che essi complementano, e prospetteremo alcuni sviluppi del discorso sulla composizione automatica.

Assunta l'idea di utilizzare l'elaboratore come compositore ed esecutore, i lavori realizzati fino ad oggi si qualificano e giustificano come la prima esplorazione delle possibilità offerte dalla programmazione per dare alle esecuzioni la maggiore varietà, flessibilità ed

agilità.

La realizzazione dei programmi si è ispirata grosso modo a quattro requisiti fondamentali:

- a) possibilità, se richiesto, di una esecuzione senza limiti di tempo: si vuole cioè che la macchina possa suonare senza interruzione. Ciò implica che l'elaborazione dei dati deve avvenire o in tempo reale, cioè durante l'esecuzione della musica elaborata negli istanti immediatamente precedenti, oppure fra una nota e l'altra, ma a rapidità tale da non disturbare l'esecuzione.
- b) Preponderante utilizzo dell'elemento casualità. I programmi contengono sempre e utilizzano nei vari modi, come vedremo, una sorgente di numeri casuali, o meglio pseudocasuali; contengono cioè un algoritmo (metodo di Lehmer o metodo usato nella subroutine RANDU della IBM) che partendo da un numero iniziale detto seme, fornisce una sequenza periodica ma di periodo T così grande ($T = 2^{29}$ per la RANDU) che all'uomo appaia imprevedibile. Il seme poi, a seconda del programma, può essere assegnato dall'uomo o ricavato a cura del programma stesso da un orologio interno della macchina.
- c) I programmi non si pongono il problema della simulazione di processi compositivi già noti, implementando grammatiche ottenibili con procedimento deduttivo a partire da testi di armonia musicale (prof. E. Gagliardo), o con procedimento induttivo a partire dall'analisi di insiemi di brani stilisticamente coerenti (prof. C. Jacoboni e Mtro. M. Baroni); si è invece puntato sullo sfuttamento delle risorse che differenziano l'elaboratore elettronico dall'uomo, per ottenere ciò che non può essere ottenuto se non sfruttando la macchina. Si è tenuto dunque in grande considerazione l'alta velocità di elaborazione del calcolatore, e la sua capacità di simulare molto bene una sorgente di numeri (eventi) casuali.
- d) Possibilità di intervento umano a diversi livelli: l'uomo non è responsabile della musica prodotta solo in quanto redattore del programma ma, data la flessibilità di quest'ultimo può intervenire di volta in volta, prima o durante l'esecuzione del programma cioè della musica, per limitare o indirizzare la casualità; il programma infatti prevede e predispone delle scelte l'esito delle quali determina l'aspetto del flusso musicale in uscita. Di fronte a tali scelte l'uomo può comportarsi in due modi: decidere di propria iniziativa, assegnando alla macchina dati che ne influenzano il comportamento nel modo voluto, oppure delegare questa decisione alla macchina stessa, come vedremo più in dettaglio.

2. CARATTERISTICHE DEI PROGRAMMI DI COMPOSIZIONE

Possiamo classificare i programmi di composizione esistenti a Pisa in base alla possibilità o meno che l'uomo ha di intervenire, prima dell'esecuzione, allo scopo di "orientare" (oltre che, naturalmente, avviare) il processo compositivo. Il funzionamento dei programmi orientabili è dunque flessibile e può essere programmato dall'utente di volta in volta; ciò non è vero per i programmi non orientabili.

PROGRAMMI ORIENTABILI		ALGOR	CREATE
PROGRAMMI NON ORIENTABILI	PLAY 1800	CREATE	RANDOM
CONVERTITORE	DAC		TAU 2 (12 voci)

Tale distinzione è, peraltro, più di comodo che sostanziale; vedremo infatti come anche il PLAY 1800 e il CREATE appartenente al DCMP (Digital Computer Music Program) siano parzialmente orientabili, seppure l'intervento dell'uomo, in questi casi, avvenga in modo inconsapevole.

I programmi orientabili elaborano sequenze di eventi sonori per una durata che deve necessariamente essere assegnata, via terminale, all'inizio dell'elaborazione.

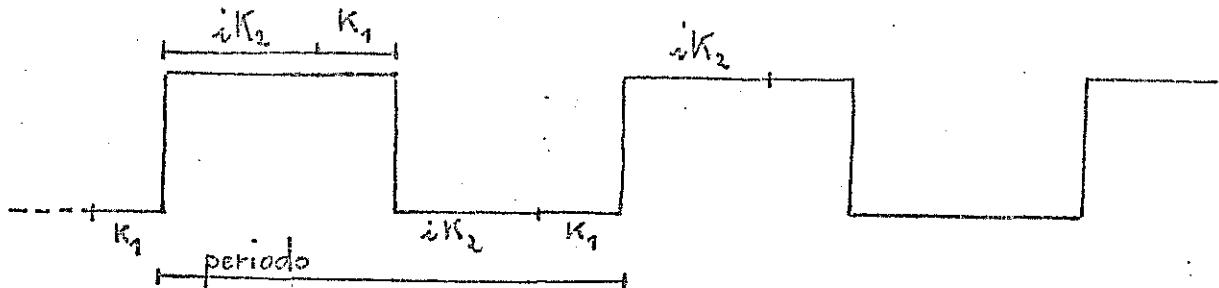
I programmi non orientabili elaborano senza limiti temporali, e la cessazione dell'elaborazione, dunque dell'esecuzione, può avvenire solo per espressa richiesta dell'utente.

I parametri gestiti dai programmi sono FREQUENZA, DURATA, TIMBRO e VOLUME; l'originalità dei programmi, tuttavia, riguarda soprattutto il trattamento dei primi due, frequenza e durata, dei quali ci occuperemo in particolare.

Esponiamo con un certo dettaglio le funzioni dei programmi ALGOR e CREATE del DCMP, dato che essi consentono di mettere in luce i principi comuni anche agli altri programmi citati.

L'ALGOR è un programma di composizione orientabile che genera sequenze musicali monodiche, temporalmente limitate, in base a un processo di generazione di numeri pseudocasuali (metodo di Lehmer) e a una serie di dati, o "OPZIONI", che vengono assegnati dall'utente, via telescrittore, prima dell'inizio dell'elaborazione.

Gli algoritmi operano su 2842 frequenze e altrettanti valori di durata, entro i limiti di 27 e 4900 Hertz per le frequenze, e di 1/100 di secondo e 9.9 secondi per i tempi. Poiché il segnale è ricavato da uno dei dispositivi binari contenuti nei circuiti della macchina, il controllo sul periodo avviene nel modo schematizzato in fig. 2:



-fig. 2-

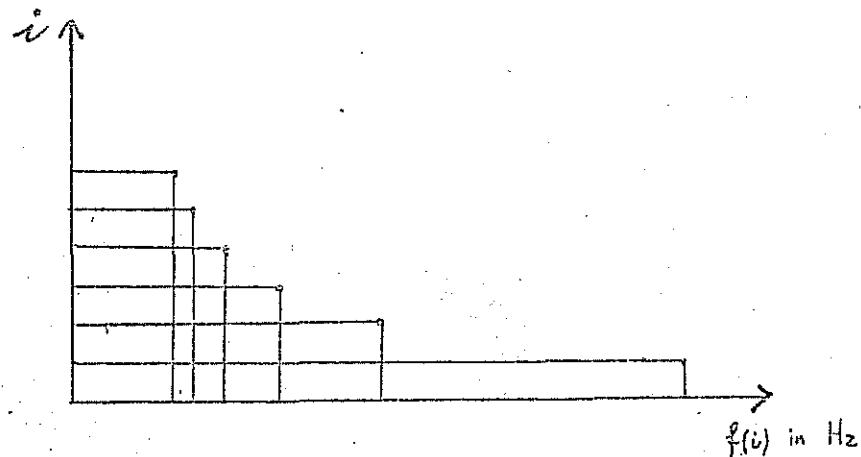
dove i è un indice discreto (intero) variabile, e K_1 , K_2 ed N sono costanti che dipendono dalla macchina.

Ne consegue che la gamma frequenziale è la seguente:

$$f_i = \frac{1}{2(K_1 + iK_2)} \quad i=1, 2, \dots, N$$

essendo 3 e 20.000 Hz i valori estremi.

Dalla relazione fra f ed i risulta che le frequenze si vanno addensando verso i valori più bassi (come avviene, tra l'altro, per tutti gli strumenti "classici" a gamma discreta.).



-fig. 3-

La densità delle frequenze è tale che, a tutti gli effetti, è come disporre della gamma continua.

A frequenze e durate vengono assegnati valori casuali generati dal programma, e opportunamente rettificati dallo stesso qualora le opzioni assegnate dall'utente contengono indicazioni in tal senso.

Le opzioni che l'utente può assegnare sono 23 e sono state studiate e introdotte come correttivo o alternativa alle facoltà decisionali dell'ALGOR; esse sono distribuite, in base alle loro funzioni, come segue:

7 per le frequenze
 6 per i tempi
 5 per le pause
 2 per la durata delle elaborazioni
 2 destinate all'algoritmo dei numeri pseudocasuali
 1 per la stampa dei dati elaborati.

L'elenco dei parametri e relative funzioni che presentiamo è tratto dal manuale di impiego del DCMP (1).

ELENCO DELLE OPZIONI

Simbolo	Numero	Funzione
/D1/	intero	determina la durata dell'elaborazione in secondi
/D2/	"	determina la durata dell'elaborazione in numero di suoni
/N1/ e /N2/	intesti	richiedono complessivamente un numero da 1 a 8 cifre - 4 in /N1/ e 4 in /N2/ - quale valore iniziale della serie di numeri pseudocasuali che verranno dati dall'apposito algoritmo nel corso dell'elaborazione.
/F1/	reale	è un fattore moltiplicativo della frequenza.
/F2/	intero	se diverso da zero rettifica le frequenze raggruppandole in zone frequentziali la cui ampiezza e durata d'impiego saranno stabilite dal programma stesso nel corso dell'elaborazione.
/F3/ e /F4/	reali	sono valori numerici assegnati dall'operatore per indicare il limite massimo (in /F3/) e minimo (in /F4/) entro cui devono essere rettificate le frequenze. Le due opzioni possono essere impiegate separatamente.
/F5/	intero	provoca la rettifica delle frequenze secondo rapporti intervallari i cui valori e la loro durata d'impiego vengono direttamente controllati dal programma. Le frequenze vengono rettificate al valore più vicino appartenente ad una serie geometrica dove la ragione è uguale al rapporto trovato e il primo elemento è il limite minimo frequentziale operante nel momento.
/F6/	reale	è un rapporto intervallare dato dall'operatore. Le frequenze vengono rettificate secondo la procedura indicata per l'opzione /F5/.
/F7/	intero	se diverso da zero le frequenze trovate, tranne la prima, vengono sostituite con altre date dai valori reciproci dei rapporti intervallari originari.

simbolo	numero	funzione
/T1/	reale	è un fattore moltiplicativo dei tempi.
/T2/	intero	se diverso da zero provoca la rettifica dei tempi entro limiti la cui ampiezza e durata d'impiego vengono controllate dal programma.
/T3/ e /T4/	reali	sono valori numerici assegnati dall'operatore per definire i limiti massimo (in /T3/) e minimo (in /T4/) entro i quali devono essere rettificati i tempi. Le due opzioni possono essere impiegate separatamente.
/T5/	intero	se diverso da zero provoca la rettifica dei tempi secondo unità temporali i cui valori e la loro durata d'impiego vengono controllati dal programma. I tempi vengono rettificati al valore più vicino appartenente ad una serie aritmetica dove la ragione è uguale alla unità di misura data e il primo elemento è dato dal limite minimo temporale operante nel momento.
/T6/	reale	è una unità temporale assegnata dall'operatore. I tempi vengono rettificati secondo la procedura indicata per l'opzione /T5/.
/T7/	intero	se diverso da zero ordina la conservazione dei dati elaborati che potranno poi essere stampati fuori linea al termine dell'impiego del DCMP.
/P1/	intero	se diverso da zero provoca la sostituzione di alcune frequenze del testo con pause. L'inserimento delle pause avviene in modo casuale.
/P2/	intero	se diverso da zero inverte il rapporto suoni-pause di un testo elaborato con l'opzione /P1/ sostituendo i suoni con pause e viceversa. /P2/ può essere impiegato solamente insieme a /P1/.
/P3/	intero	è un valore assegnato dall'operatore per stabilire la percentuale di pause voluta nell'elaborato.

Ricevute le opzioni assegnate dall'utente, il programma elabora il testo e, ad elaborazione ultimata, spedisce i dati al convertitore ottenendone l'esecuzione immediata. E' possibile inoltre, con l'uso di un ulteriore parametro, ottenere una esecuzione illimitata di sequenze tutte generate nel rispetto della configurazione parametrica assegnata dall'utente.

Terminata l'esecuzione di una sequenza ne viene elaborata ed eseguita una successiva, che potrà risultare somigliante ma non identica alla precedente, dato che il processo di generazione di numeri pseudocasuali non viene re-inizializzato al termine di ogni esecuzione. La rapidità di elaborazione è tale, poi, da non provocare soluzione di continuità nella esecuzione: è forse legittimo, già a questo livello, parlare di elaborazione in tempo reale, anche se solo apparente.

Osservazioni.

1) E' da notarsi la rigorosa simmetria che viene rispettata nel trattamento dei valori delle frequenze e delle durate.

2) Si veda la funzione di T_2 rispetto a T_3 e T_4 , T_5 rispetto a T_6 , e gli stessi parametri per le frequenze.

E' questo un modo di impostare i rapporti uomo-macchina molto tipico di questo e degli altri programmi citati: molte scelte (come le singole frequenze e le singole durate) vengono effettuate casualmente dalla macchina; alcune scelte (come la determinazione del campo frequenziale) vengono predisposte dal programma e proposte all'uomo; i parametri T_2 e T_5 , f_2 ed f_5 rappresentano una ulteriore possibilità per l'uomo: quella di rimandare alla macchina anche le scelte che egli stesso potrebbe compiere.

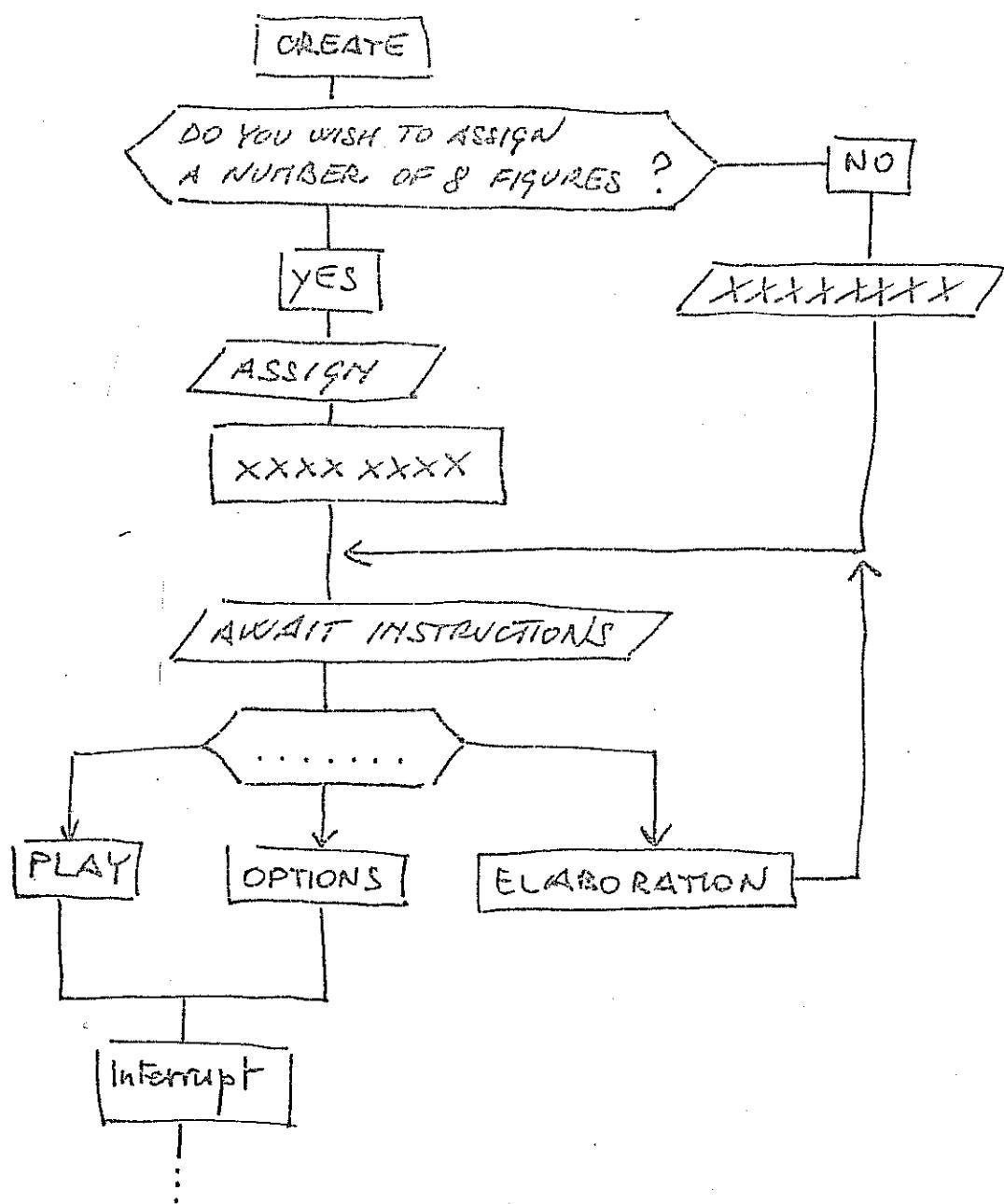
Come esempio di programma non orientabile descriviamo il CREATE - DCMP.

Esso è un programma di composizione che genera sequenze musicali monodiche, temporalmente illimitate, almeno in linea di principio, in base a un processo di generazione di numeri pseudocasuali (metodo di Lehmer).

L'intervento dell'uomo consiste quasi esclusivamente nell'avviare l'elaborazione, via telescrittore, e nel richiederne l'arresto.

Il diagramma di flusso che segue rappresenta il dialogo utente macchina che avvia il processo.

Le istruzioni PLAY, OPTIONS, ELABORATION corrispondono ai tre diversi modi di procedere del programma.



- fig. 4 -

-PLAY-

Questo comando mette in funzione alcuni algoritmi studiati per assicurare ampie possibilità combinatorie dei parametri frequenza e durata, riducendo al minimo le probabilità di ripetizione delle possibili configurazioni. Allo stesso tempo il programma limite, per così dire, la casualità pura, scegliendo di tanto in tanto fra alcune differenti modalità di elaborazione dei valori di frequenza e durata e decidendo autonomamente per quanto tempo trattenersi ad elaborare secondo la modalità considerata, prima di effettuare una nuova scelta.

I modi di elaborazione per le frequenze sono 5:

- 1) Random - Il programma genera valori frequentziali a caso, scegliendo nella gamma, già descritta, del quasi-continuo.
- 2) Range 1 - Il programma decide gli estremi inferiore e superiore entro il quale effettua la scelta casuale delle frequenze, e decide per quanto tempo effettuare questa ricerca prima di cambiare Range, scegliendo 2 nuovi estremi.
- 3) Range 2 - Il programma assume per un certo periodo di tempo una certa frequenza come base per la determinazione dei valori frequentziali; questi vengono ottenuti aggiungendo incrementi relativamente piccoli, e casuali, alla base. Poiché l'incremento è una variabile casuale la cui funzione densità (piatta) ha varianza fissa (indipendente dalla frequenza di base) si ottiene che i Range gravi appaiono più ampi di quelli acuti. Anche la durata di utilizzo della frequenza di base viene decisa casualmente di volta in volta dal programmatore.
- 4) Serie aritmetiche - Per un certo periodo di tempo, deciso casualmente dal programma, alle frequenze vengono assegnati, ordinatamente, i valori di una serie aritmetica di ragione pure casuale.
Le scale che si ottengono risultano compresse verso gli acuti.
Raggiunti gli estremi della gamma, viene invertito il moto.
- 5) Serie geometriche - Per un certo periodo di tempo, deciso casualmente dal programma, alle frequenze vengono assegnati, ordinatamente, i valori di una serie geometrica di ragione pure casuale. Le scale che si ottengono risultano, in un certo senso, "ben temperate". Raggiunti gli estremi della gamma, viene invertito il moto.

Una possibile variante ai due ultimi modi è quella per cui il programma sceglie casualmente, nota per nota, la ragione da utilizzare per

il calcolo della frequenza successive.

Per fare riferimento ai criteri di generazione di frequenze, durate e pause nel seguito continueremo ad usare i termini "modi", o "modalità", di generazione dei parametri sonori.

Per fare riferimento ai modi 2) e 3) in seguito verrà usato il termine "campi frequenziali provvisori".

I modi di elaborazione per le durate sono tre:

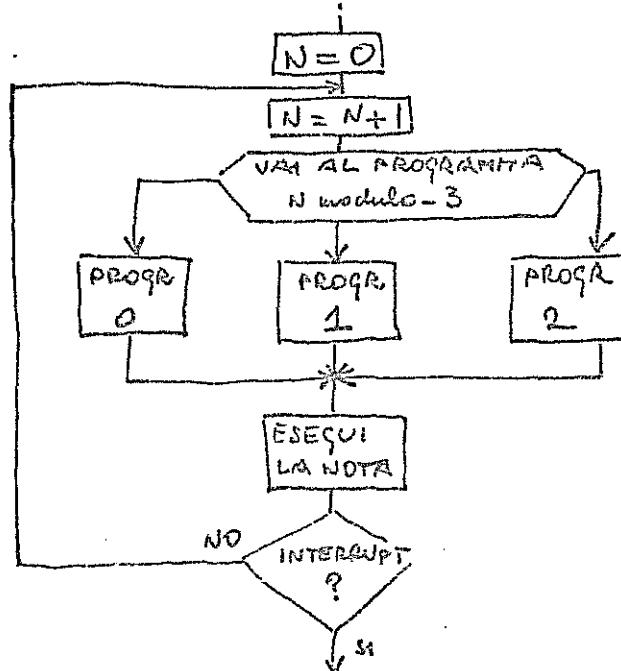
- 1) Periodi fissi - Il programma assegna ad ogni frequenza la durata di due periodi. Risulta che i suoni si accorciano salendo verso gli acuti.
- 2) Periodi random - Il programma assegna casualmente ad ogni frequenza una durata compresa fra 1 e 9 periodi. Mediamente si ottiene lo stesso effetto del caso precedente.
- 3) Durate random - Il programma assegna casualmente ad ogni frequenza una durata, compresa fra 3 e 100 centesimi di secondo.

Come per le frequenze, anche per le durate il programma decide casualmente per quanto tempo lavorare secondo uno dei modi esposti, prima di sceglierne un altro.

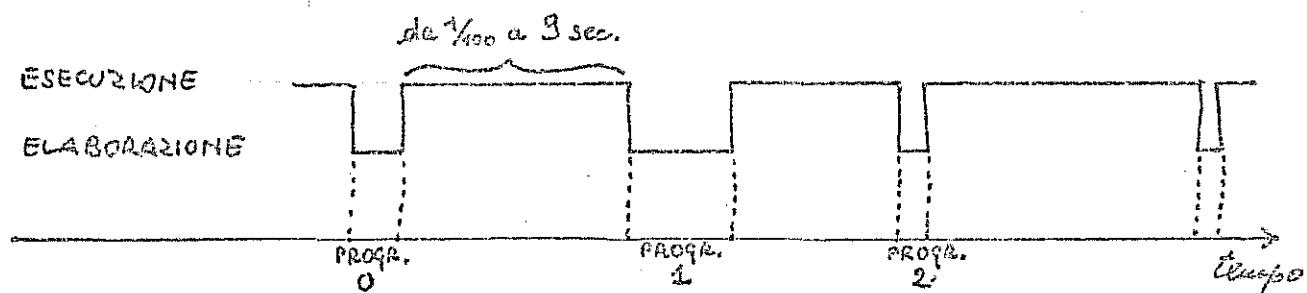
Sono previste anche tre modalità di inserimento delle pause nel testo elaborato:

- 1) niente pause
- 2) pause distribuite casualmente
- 3) pause periodiche con periodo che cade dopo un certo tempo, deciso dal programma.

Per ottenere poi una maggiore varietà nei flussi generati dal comando PLAY, sono state realizzate 3 copie dello stesso programma; ognuna di esse funziona, sempre a tempo limitato, secondo una delle modalità descritte e indipendentemente dalle altre; la sequenza finale è codificata dalle note generate ciclicamente dai tre programmi, come mostrato nel seguente diagramma di flusso:



L'esecuzione di ogni nota avviene subito dopo la sua elaborazione, cioè le tre copie del programma elaborano una nota alla volta. Il funzionamento dell'intero programma nel tempo può essere così schematizzato:



-fig. 6-

Come nel caso dell'ALGOR-DCMP relativo al parametro che ricicla l'utilizzo di una certa configurazione di opzioni assegnate dall'utente, anche in questo caso si potrà parlare di tempo reale solo apparente: i tempi di elaborazione, "morti" dal punto di vista dell'esecuzione, introducono un ulteriore elemento casuale, dato da una serie di pause quasi impercettibile e comunque indistinguibile dalla serie di pause creata dal programma.

-OPTIONS-

Questo comando, che a differenza del PLAY mette in funzione una sola copia del programma, leggermente modificata, pone l'operatore in condizione di compiere scelte libere, ma non consapevoli, fra i caratteri della telescrivente. Appare infatti il messaggio: "Type as you wish the Alphabetic Keys". La scelta e il numero dei caratteri impiegati determineranno parzialmente le caratteristiche morfologiche del tessuto sonoro. L'utente può battere fino a 120 caratteri sulla tastiera ma ciò che conta (e l'utente non lo sa) è il numero di battiti per ognuna delle tre principali righe della tastiera, essendo la prima riga associata alle frequenze, la seconda alle durate, la terza alle pause. Tali numeri, che sono considerati modulo 13, modulo 9 e modulo 3 per la prima, seconda e terza riga rispettivamente, determinano la scelta e la alternanza delle modalità di generazione dei parametri sonorici che verranno adottati dal programma, e che sono le stesse modalità viste per il PLAY. Dando il comando di interrupt e ribattendo a caso sulla tastiera si potrà ottenere un flusso sonoro di carattere diverso, ma imprevedibile.

Se poi fra i caratteri battuti ne compaiono alcuni prefissati, l'alternanza dei modi di generazione sarà casuale e le sequenze appariranno ancora meno strutturate. Anche per questo comando, come per il comando PLAY, le note vengono elaborate ed eseguite una alla volta.

-ELABORATION-

Questo comando provoca la stampa, alla telescrivente, del messaggio:

TIME ELABORATION IN YEARS, DAYS, HOURS, MINUTES MAXIMUM: 2043 YEARS
LESS 5 MINUTES, MINIMUM : 5 MINUTES.

e pone il programma in attesa di dati.

I dati assegnati a questo punto dall'utente, secondo un semplice formato, rappresentano la qualità di musica, in anni, giorni, ore e minuti che egli non vuole ascoltare.

Il programma compone le sequenze numeriche allo stesso modo che con il PLAY, ma senza eseguirle, quindi più rapidamente: è possibile, dopo questa fase ed assegnando il PLAY, ascoltare la musica che, senza l'opportunità del comando ELABORATION, si sarebbe ascoltata dopo anni di esecuzione ininterrotta.

Alcune osservazioni sugli altri programmi.

-PLAY 1800-

L'originalità delle sequenze generate dal PLAY 1800 risiede soprattutto nella timbrica, della quale non discutiamo; il processo compositivo è peraltro molto simile a quello del CREATE-DCMP. E' previsto, in aggiunta, un tipo di intervento da parte dell'utente più diretto di quello della battitura casuale sulla tastiera: se infatti nel CREATE-DCMP per questa operazione era necessario interrompere l'esecuzione-elaborazione, nel PLAY 1800 è possibile intervenire "in diretta" azionando gli interruttori della tastiera della telescrivente. Tali interruttori impostano i valori binari di alcune variabili che il programma analizza periodicamente: in base a tali valori l'aspetto delle sequenze in uscita viene modificato, in maniera però sempre imprevedibile per l'utente.

-CREATE TAUMUS-

Il CREATE del TAUMUS lavora con un massimo di 12 voci e secondo procedure del tutto simili a quelle dell'ALGOR DCMP, vale anche in questo caso il criterio di assegnazione (consapevole) di opzioni per la determinazione di alcune caratteristiche delle singole voci. Le voci per le quali non sono state assegnate opzioni vengono elaborate casualmente.

-RANDOM TAUMUS-

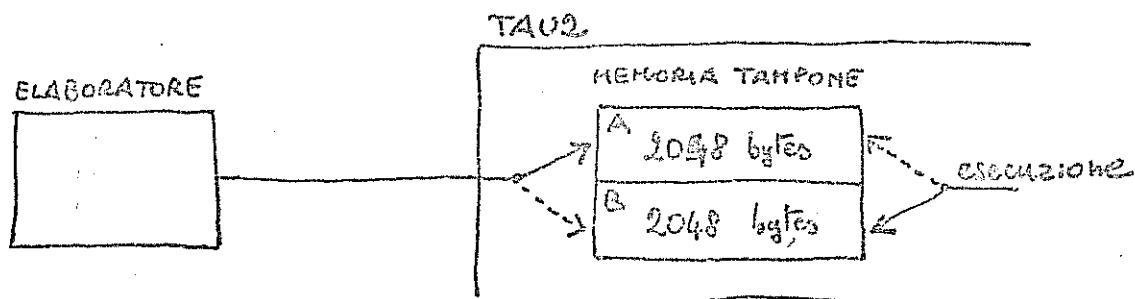
Il RANDOM del TAUMUS lavora a sei voci fisse, raggruppate a due a due, secondo criteri del tutto simili a quelli del CREATE-DCMP. L'accoppiamento delle voci consiste nel fatto che mentre le prime due voci eseguiranno (indipendentemente fra loro) scale a passo fisso, le seconde due lavoreranno, ad esempio, entro campi frequenziali, e le terze due in modo completamente casuale e così via.

Il RANDOM-TAUMUS lavora in tempo reale, e non solo apparente come avviene per il DCMP.

Ciò è possibile grazie al fatto che il TAU 2 è un convertitore abbastanza intelligente: esso è dotato di un controllo micropogrammato e di una memoria tampone di 4096 byte, divisa in due parti uguali

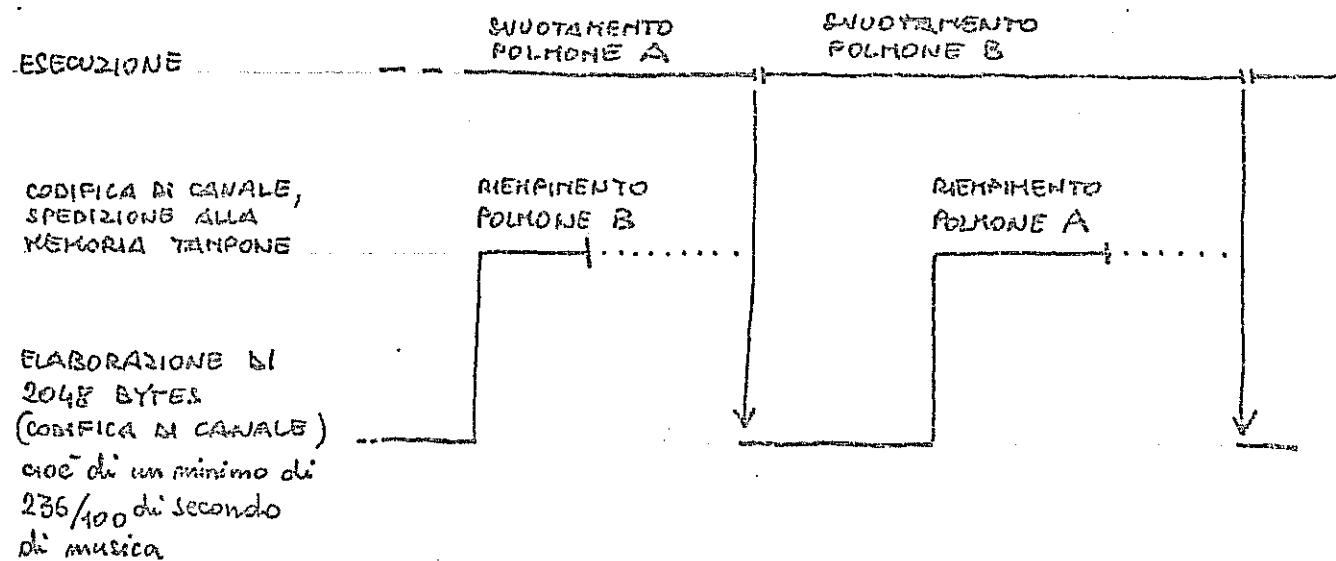
('polmoni') che gestiscono le code di dati, ricevuti dall'elaboratore secondo la codifica di canale, con il criterio FIFO (first in first out).

Il funzionamento dei 'polmoni' è così schematizzabile:



-fig. 7-

Mentre avviene l'esecuzione dei dati contenuti nel 'polmone' B, che, in base alla concezione del programma e della codifica di canale, può contenere musica per un minimo di $236/100$ di secondo, l'elaboratore ha il tempo di preparare, codificare e spedire i dati successivi, che riempiranno il 'polmone' A. Dal punto di vista del programma il procedimento è il seguente:



-fig. 8-

La freccia verticale indica che quando un polmone è stato svuotato il TAU 2 segnala all'elaboratore che è possibile riprendere l'~~esecuzione~~^{elaborazione}.

Osservazioni generali.

Riconsiderando in blocco tutte le funzioni dei programmi esposti,

possiamo osservare che i principi cui essi si sono ispirati si riferiscono a poche semplici operazioni fondamentali, dalla immediata interpretazione in chiave insiemistica:

- | | |
|---|--|
| 1) DEFINIZIONE DELL'INSIEME
(per enumerazione degli elementi,
cioè dei possibili valori dei parametri acustici) | SEQUENZE RANDOM |
| 2) INTRODUZIONE DI UNA RELAZIONE
D'ORDINE
(l'ordinamento naturale) | SEQUENZE A SCALE:
PASSO RANDOM |
| 3) INDIVIDUAZIONE DI SOTTOINSIEMI
(sfruttando l'ordinamento preesistente) | SEQUENZE RANDOM A CAMPI |
| 4) INTRODUZIONE DI UNA METRICA | SEQUENZE A SCALE:
PASSO FISSO |
| 5) INDIVIDUAZIONE DI ULTERIORI
SOTTOINSIEMI
(sfruttando l'ordinamento e la metrica preesistenti) | SEQUENZE RANDOM A CAMPI
CON INTERVALLI MULTIPLI
DI UN INTERVALLO BASE
FISSO |

Sideve anche osservare che tali operazioni, utilizzate per variare e rendere più interessanti gli effetti sonori prodotti dalla pura casualità, sono molto "orientate" alla programmazione, nel senso che sono tipicamente realizzabili con gli strumenti che quest'ultima può fornire; addirittura si potrebbe forse sostenere, rovesciando il discorso, che le stesse risorse della programmazione Fortran abbiano in alcuni casi suggerito quali criteri di generazione adottare.

E' evidente l'utilità, ad esempio, dei confronti aritmetici del Fortran per la generazione di frequenze comprese fra un estremo superiore ed uno inferiore, come chiara risulta la parentesi fra l'istruzione DO con inizio, fine, passo del Fortran, e i sottoinsiemi di tipo 4) della precedente tabella (il criterio di "inizio, fine, passo" è anche presente in quasi tutti i comandi dei programmi "convenzionali" di Pisa).

Ma, anche indipendentemente dalla facilità di programmazione, i sottoinsiemi di tipo 3 e 4 sono fra i più "economici" essendo sufficienti rispettivamente due numeri (estremo superiore, estremo inferiore) e tre numeri (i precedenti più il passo) per costruirli.

Chiaramente meno economica sarebbe, ad esempio, la individuazione, nell'insieme delle frequenze, di un sottoinsieme diatonico, per giungere poi al caso limite di sottoinsiemi di sottoinsiemi di complessità tale da richiedere, per la loro individuazione, l'enumerazione degli elementi uno per uno.

3.

POSSIBILI LINEE DI SVILUPPO PER I PROGRAMMI DI COMPOSIZIONE AUTOMATICA

Fra i più evidenti propositi dei programmi di composizione automatica vi è quello di fornire musica il più possibile varia e capace di interessare l'ascoltatore.

L'esperienza e i limiti dei procedimenti compositivi esposti suggeriscono alcune considerazioni sul modo di impostare il lavoro futuro.

1) Come già osservato, fra i requisiti più importanti che si voleva no soddisfatti nei programmi c'è il poter sfruttare le risorse più tipiche dell'elaboratore, e principalmente la velocità, per ottenere e mettere in luce ciò che non è possibile ottenere con i limitati mezzi umani.

A posteriori si può osservare che ciò è stato realizzato soprattutto nel senso dell'esecuzione: si è cioè ottenuto un esecutore di raro "virtuosismo" ma un compositore ancora incompleto. Nasce dunque l'esigenza di approfondire la ricerca specialmente a monte, cioè nel senso della composizione, per migliorare e rendere più complessi i processi di generazione di strutture musicali.

2) Ciò che è ricco e vario dal punto di vista matematico, cioè, se così si può dire, per la macchina, non lo è necessariamente dal punto di vista percettivo, cioè per l'uomo che ascolta. Il fatto di ridurre al minimo la probabilità di ripetizione di eventi sonori, o loro combinazioni, in un processo di composizione, ad esempio, non è sufficiente a garantire che le sequenze risultanti appaiano variate. Riferiamoci, per facilitare il discorso, al modello più semplice e generale di processo compositivo, cui i processi descritti sono chiaramente riconducibili: una sorgente discreta S , cioè un dispositivo ideale in grado di emettere sequenzialmente i simboli di un alfabeto finito A , detto alfabeto della sorgente.

$$A = \{ a_1, a_2, a_3, \dots, a_n \}$$

Ogni processo compositivo corrisponde a un modo di funzionare della sorgente. Il processo compositivo random, caratteristico di tutti i programmi esposti, corrisponde a una sorgente senza memoria e equiprobabile, cioè una sorgente che emette i simboli dell'alfabeto con probabilità

$$p(a_1) = p(a_2) = \dots = p(a_n) = 1/m$$

indipendente dai simboli precedentemente ammessi. In questo caso il funzionamento della sorgente, dunque il processo compositivo, è completamente definito dai valori di probabilità assegnati.

Per una sorgente probabilistica; di cui quella equiprobabile senza memoria è un caso particolare, la "varietà matematica" può essere

definita con più precisione, potendo assegnarne un metodo di misura: sorge spontaneo utilizzare a questo scopo la "quantità di informazione", o Entropia, introdotta da Shannon e basata appunto sulle probabilità di emissione dei simboli:

$$H_S = - \sum_{i=1}^n p(a_i) \log_2 p(a_i)$$

dove H_S è l'entropia della sorgente, o quantità di informazione mediamente associata ad un simbolo emesso (la formula è facilmente estensibile al caso di probabilità condizionali).

E' possibile ora riformulare l'osservazione fatta all'inizio di questo paragrafo in questi termini: una grande "varietà matematica" (oggettiva) intesa come alta entropia della sorgente non corrisponde necessariamente ad una grande "varietà percepita" (soggettiva).

Di fatto, tra tutte le possibili sorgenti probabilistiche, che emettono cioè i simboli del proprio alfabeto esclusivamente in base a vettori di probabilità, eventualmente condizionati dai simboli precedentemente emessi, la sorgente random

$$p(a_1) = p(a_2) = \dots = p(a_m) = 1/m$$

è quella che produce la massima entropia, generando sequenze completamente casuali: le probabilità

$$p(0) = p(1) = 1/2$$

ad esempio, determinano un valore dell'entropia

$$H = 1 \text{ bit/simbolo emesso}$$

che è il massimo valore possibile per sorgenti binarie; tuttavia tale sorgente non è certamente fra quelle che realizzano le sequenze più varie e degne di interesse.

Con ciò si vuole affermare che una evoluzione nei processi di composizione automatiza dovrà ricercare una definizione del concetto di "varietà" meno astratta di quella posta dall'equazione:

varietà & casualità (cioè massima entropia)
ma più orientata a ciò che l'uomo intende per varietà, cioè più consapevole delle caratteristiche e dei limiti della percezione.

3) In questa prospettiva e mantenendo come modello la sorgente dal comportamento esclusivamente probabilistico, il passo che appare più ovvio per la limitazione dell'effetto di monotonia nelle sequenze generate è quello di ridurre "la varietà in senso matematico", che per addosso è ancora definita come entropia della sorgente.

Ciò può essere fatto sostanzialmente in due modi:

a) Diminuendo il numero di simboli dell'alfabeto, ma conservando l'equiprobabilità. Questo procedimento è ampiamente adottato nei programmi di composizione automatica descritti; esso corrisponde al criterio di scelta casuale dei valori da assegnare ai parametri sonori ma all'interno di intervalli fissi (come avviene per i "campi frequenziali").

b) Condizionando le probabilità di emissione dei simboli ai simboli emessi precedentemente, cioè ai contesti.

Supponendo di limitarsi a contesti di lunghezza 1, cioè di far dipendere l'emissione di un simbolo dalle 1 emissioni precedenti, la sorgente risulterà completamente definita assegnando le probabilità di emissione di tutte le possibili $(\ell + 1)$ -uple ordinate realizzabili con gli n simboli dell'alfabeto (esse sono $n^{(\ell+1)}$):

$$p(a_{i_1}, a_{i_2}, \dots, a_{i_{\ell+1}})$$

Tale conoscenza permette infatti di ricavare tutte le probabilità condizionali:

$$p(a_i | \alpha)$$

essendo $\alpha \in \{\lambda\} \cup A^1 \cup A^2 \cup \dots \cup A^\ell$

dove λ è il contesto vuoto (e si hanno, come caso particolare, le probabilità assolute di emissione) ed A^* è l'insieme delle x -uple ordinate di lettere dell'alfabeto.

Questo procedimento non è stato adottato precisamente nei programmi di composizione automatica, ma in un programma della parte conversazionale del DCMP, che viene posto in esecuzione con il comando MARKOV. Grazie a questo comando l'utente può chiedere la composizione di testi sulla base di probabilità condizionali (viene anche assegnata la lunghezza del contesto) di emissione degli eventi sonori (frequenze, durate, intervalli frequentziali ...), probabilità che vengono ricavate dall'analisi eseguita dal programma stesso su un pezzo musicale contenuto in biblioteca e proposto dall'utente.

Si può osservare che tra i due metodi di riduzione dell'entropia esposti esiste la seguente differenza concettuale: il metodo a) non tiene conto della sequenzialità temporale dell'esecuzione musicale, il metodo b) ne tiene conto; in altri termini, il primo genera sequenze invarianti per inversione temporale, il secondo no. Per evitare una definizione di invarianza per inversione temporale troppo legata alla soggettività del giudizio di somiglianza tra una sequenza e la sua inversa, quindi imprecisa ed ambigua, possiamo assumere questa definizione:

una sequenza è invariante per inversione se ad essa e alla sua inversa compete la stessa probabilità di emissione.

Questa definizione è debole, ma sufficiente al nostro scopo di stabilire la differenza tra a) e b).

L'efficacia dei due metodi di riduzione entropica ai fini pratici di una maggiore varietà delle sequenze è sufficientemente comprovata dalle esperienze musicali con i programmi che implementano i metodi stessi. La discussione che segue sui pregi e sui limiti teorici di tali metodi consente tuttavia di progettare processi compositivi più complessi.

4) Questa osservazione riguarda i pregi e i limiti della sorgente puramente probabilistica come modello di processo compositivo. Un processo compositivo è buono se produce musica giudicata varia e interessante dagli ascoltatori. Così un metodo matematico di analisi delle strutture musicali è buono se è, diciamo così, "sensibile" a ciò che l'ascoltatore intende per "vario" o "degno di interesse", cioè può rilevare e misurare queste grandezze. Ora sulla base dell'osservazione 3, va sottolineato che il comportamento puramente probabilistico della sorgente e il metodo di misurazione dell'entropia come indice di varietà, o complessità, oricchezza, delle sequenze, sono strettamente collegati, rappresentando le due facce, rispettivamente sintetica ed analitica di uno stesso metodo, basato esclusivamente su matrici probabilistiche. Ne consegue che in questo caso, la bontà o meno del metodo di sintesi riflette la bontà o meno del metodo di analisi, per non dire addirittura che le due cose coincidono. Possiamo parlare dunque semplicemente di bontà o meno del metodo probabilistico. La pertinenza di tale metodo con i fenomeni musicali è ampliamente dimostrata: basterà citare L.B.Meyer (2) :

"Uno stile musicale, una volta che sia entrato a fare parte delle risposte abituali di compositori, esecutori ed ascoltatori, può essere considerato un sistema complesso di probabilità. ... Da questi sistemi intrinsecizzati di probabilità sorgono le attese-tendenze - sulle quali si edifica il significato musicale". Non scendiamo in una dettagliata esposizione dell'articolo citato, che analizza con notevole accuratezza i meccanismi psicologici elementari entranti in funzione durante l'ascolto musicale: creazione dell'attesa; valutazione, in una situazione di incertezza, dei possibili eventi sonori conseguenti; conferma o non conferma delle previsioni effettuate; reinterpretazione del passato alla luce dell'evento presente (vedi anche(3)).

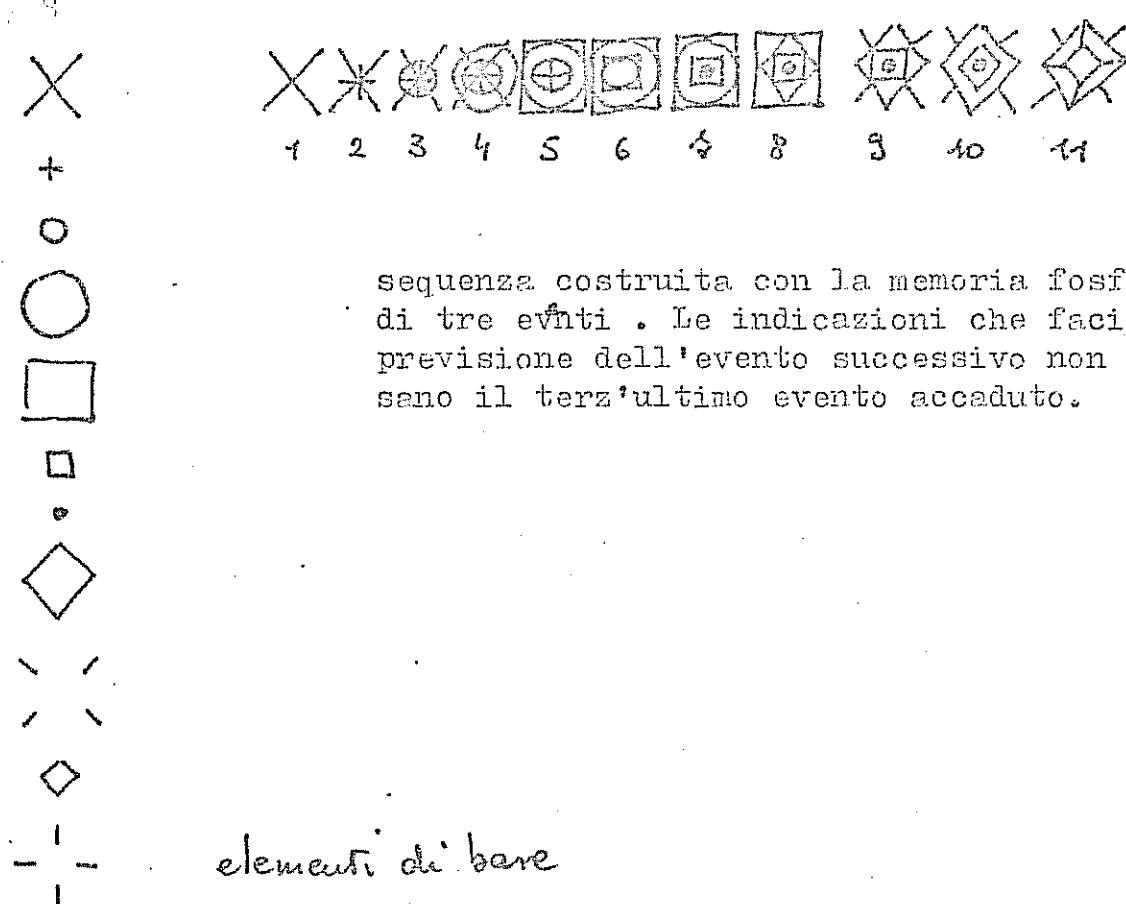
Diciamo invece che il metodo probabilistico non è sufficiente (o comunque, non è necessario, cioè non è il solo possibile) per simulare o valutare tali meccanismi psicologici, in una parola, per renderne conto.

Per almeno tre motivi:

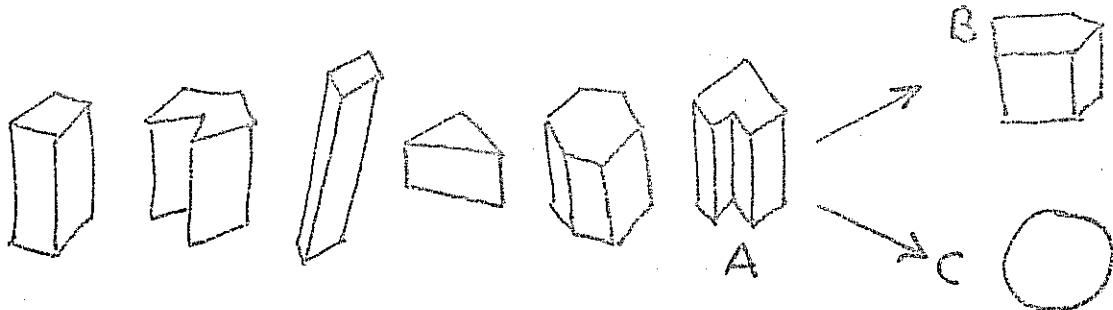
a) Se è vero che gli eventi sonori immediatamente precedenti al momento presente (contesti) hanno una funzione determinante nel configurare la situazione di attesa dell'evento nuovo, è anche vero che tale funzione viene esplicata in maniera diversa dai contesti "brevi" e da quelli "lunghi". Cioè il significato che un evento nuovo assume risolvendo una attesa dipenderà dall'esatta configurazione particolare delle precedenti poche note (fig. 9a) e dall'andamento

generale delle precedenti molte (fig.9b).

E' impensabile ritenere che il compositore faccia dipendere la scelta di un evento sonoro da tutti i minimi particolari degli episodi musicali precedenti come è impensabile ritenere che nel percepire un evento sonoro l'ascoltatore possa assegnargli un significato dipendente dalla memoria di tutti i minimi particolari degli episodi musicali precedenti. Inoltre la applicazione del metodo probabilistico ai contesti lunghi richiederebbe la memorizzazione e la gestione, da parte del compositore/ascoltatore, delle matrici delle probabilità condizionali, le cui dimensioni raggiungono ben presto proporzioni esagerate, dato che il numero dei possibili contesti cresce esponenzialmente rispetto alla lunghezza del contesto stesso. Se dunque il metodo probabilistico può rendere conto dei fenomeni relativi ai contesti brevi, cioè della cosiddetta memoria fosforescente, non è adeguato al trattamento dei contesti lunghi. Si pone la necessità di un metodo che possa rappresentare e memorizzare il passato in maniera in qualche modo riassuntiva.



- fig. 9a -

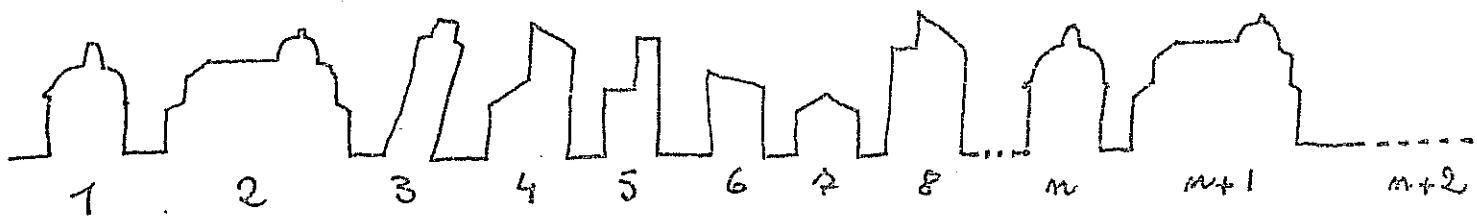


- fig. 9b -

Il fatto che le previsioni per l'evento successivo di A favoriscano B rispetto a C, non è dato dalla memoria precisa di un contesto breve, ma dalla "memoria riassuntiva" di tutto il passato, che è visto come un insieme di forme poliedriche.
 (E' questo un caso di previsione per analogia)

b) Se durante l'ascolto di una sequenza prende inizio un episodio nello stesso modo in cui ha preso inizio un episodio molto indietro nella sequenza, la situazione che si viene a creare è fortemente caratterizzata da come l'episodio passato si sviluppava. Il metodo probabilistico non è ovviamente in grado di rendere pienamente conto di questa situazione, che è esemplificativa di tante altre analogie. Si pone la necessità di un metodo che possa memorizzare localmente ma in maniera completa determinati episodi (fig. 10).

- fig. 10 -



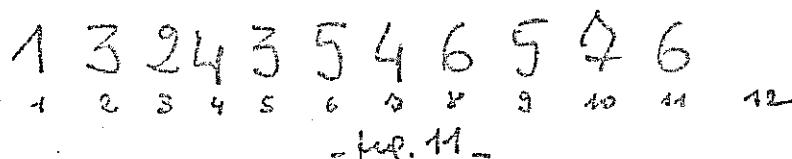
La previsione dell' $\frac{m+2}{m+1}$ ° evento si fonda sulla memoria della breve sottosequenza remota degli eventi 1, 2 e 3.

c) Sempre secondo L.B.Meyer : "Ciò che rimane costante da stile a stile non sono le scale, i modi, le armonie, i tipi di esecuzione, ma la psicologia dei processi mentali umani".... La mente umana in cerca di stabilità e completezza, "aspetta, ad esempio, delle lacune strutturali da riempire".

Ora sempre riferendoci ad un esempio singolo, ma intendendo una classe di situazioni, si consideri una sequenza di frequenze crescenti, per fissare le idee DO RE MI FA; la nota successiva più probabile appare il SOL, ma è abbastanza evidente che tale attesa, ponendosi ad esempio, dal punto di vista dell'ascoltatore, non è giustificata dal fatto che l'intera sequenza DO RE MI FA SOL è mantenuta altamente probabile, essendo stata ascoltata molte altre volte. Esiste un motivo più profondo: l'attesa del SOL è l'attesa

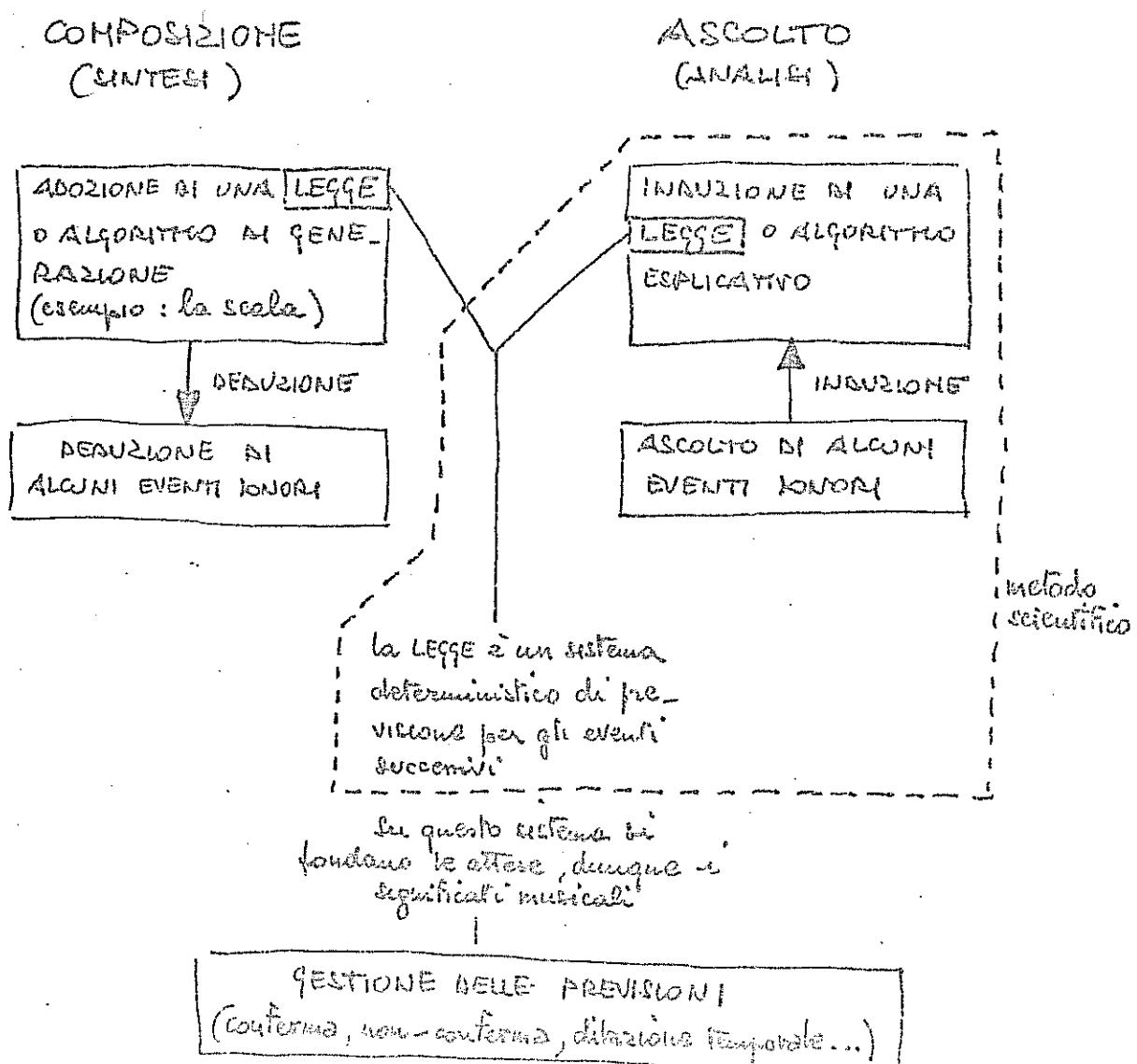
di un elemento che tende a completare, o almeno a proseguire, un processo (sequenza diatonica ascendente a gradi congiunti) di cui l'ascoltatore ha intuito l'andamento dalle prime quattro note DO RE MI FA.

E' facile immaginare altri esempi, anche completamente nuovi (si potrebbero citare le strutture generate col calcolo combinatorio del comando COMB del TAUMUS) i quali, pur non facendo parte dell'esperienza musicale di alcun ascoltatore, quindi non rifacendosi ad alcun sistema probabilistico, presentano in alto grado di prevedibilità, cioè creano delle precise attese.



La previsione del 12º evento si fonda sulla riconoscibilità della struttura della sequenza, cioè dell'algoritmo che la genera.

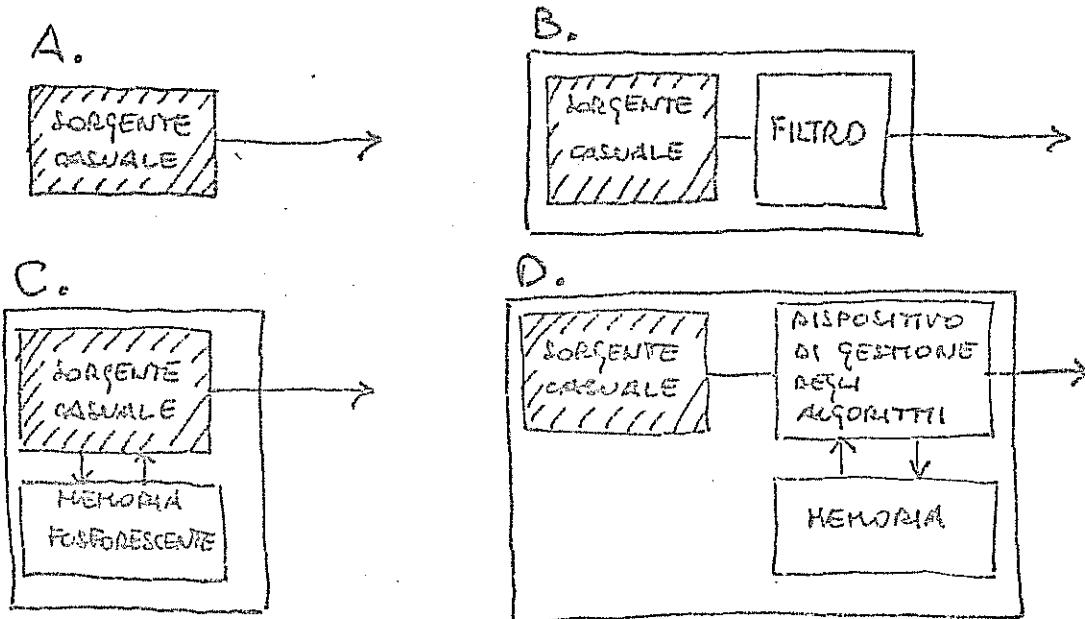
Ciò che accade in questicasi, a livello di composizione e di ascolto, può essere così schematizzato:



Tutto ciò appare molto rigido, ma basta immaginare la possibilità di leggi provvisorie che decadono nel corso della sequenza, o di segmenti della sequenza deducibili da (o che inducono) più leggi, cioè immaginare situazioni ambigue per convincersi del contrario. Si pone dunque la necessità di un metodo che possa maneggiare sequenze o sottosequenze di eventi sonori in senso deterministico e non probabilistico, e che adotti, dal punto di vista compositivo, che più interessa in questo scritto, algoritmi di generazione di strutture più o meno complesse (si osservi che ciò non significa necessariamente muoversi in direzione di una musica più ordinata perché dotata di leggi deterministiche. Una legge può avere senso anche solo in quanto rende possibile la sua NON osservanza).

5) I limiti insiti nel metodo puramente probabilistico indicano, come abbiamo discusso nell'osservazione precedente, che l'automazione di processi compositivi più evoluti dovrebbe porsi i seguenti obiettivi:

- 1) Miglioramento qualitativo della memoria
- 2) Creazione e gestione di algoritmi, cioè di leggi deterministiche di generazione di strutture
- 3) Viene da sé che memoria e dispositivo di produzione-gestione degli algoritmi dovranno essere costruiti in modo da permettere l'interazione, cioè lo scambio di dati: la memoria, se non sarà richiesto il dettaglio, dovrà conservare informazioni, più generiche, relative agli algoritmi che hanno operato nel passato; gli algoritmi, a loro volta, verranno via via scelti, orientati e combinati, durante il processo di composizione, in base alla conoscenza del passato, cioè del contenuto della memoria. Nella fig. che segue è presentato lo schema di memoria del processo compositivo (sorgente D) che stiamo progettando in confronto con i processi precedentemente considerati.



- Fig. 13 -

(Ripetiamo che in questo articolo le sorgenti rappresentano modelli di processi compositivi; più in generale si potrebbe dire che le sorgenti complesse B,C e D rappresentano diversi criteri di utilizzo della sorgente casuale A).

A è la sorgente probabilistica completamente casuale (equiprobabilità).

B è la sorgente probabilistica senza memoria che limita l'entropia nel modo a) discusso all'osservazione n.3 (campi frequenziali provvisori...). Abbiamo chiamato "filtro" il dispositivo posto in serie alla sorgente casuale intendendo sottolineare che esso crea strutture senza elaborare i dati casuali che riceve dalla sorgente, ma appunto "filtrandoli".

C è la sorgente probabilistica con memoria, il cui comportamento dipende dai contesti (probabilità condizionali), e che limita l'entropia nel modo b) discusso all'osservazione n.3.

D è la sorgente progettata per ultima, che utilizza una memoria più complessa e affianca all'aspetto probabilistico (sorgente casuale) l'aspetto deterministico (uso degli algoritmi).

E' chiaro che la funzione del dispositivo posto in serie alla sorgente casuale nella sorgente D non è assolutamente paragonabile a quella di un filtro (come avviene per la sorgente B), data la sua capacità di elaborazione; potrà semmai essere paragonata, ma solo in termini quantitativi, a quella di un amplificatore.

Si osservi a questo proposito, che il particolare processo di generazione di frequenze per serie aritmetiche o geometriche a ragione fissa del CREATE DCMP è perfettamente modellato dalla sorgente D (privata della memoria): l'amplificazione in questo caso avviene nel senso che pochi dati casuali in entrata, indicanti l'inizio, la fine e la ragione della serie verranno convertiti in molti dati in uscita costituenti la serie nel suo completo sviluppo.

6) Concludiamo con una discussione su alcuni problemi relativi alle funzioni e alla progettazione dei tre dispositivi che formano la sorgente D introdotta al paragrafo precedente. Non discuteremo invece il problema della gestione delle attese (cioè i criteri con i quali le possibili previsioni dell'evento successivo vengono confermate, non confermate o differite) che può essere affrontato solo dopo aver ragionevolmente risolto quello della creazione delle attese stesse.

Memoria

La memoria deve funzionare a diversi livelli. Oltre alla memoria fosforescente (utilizzo di matrici probabilistiche) deve esistere la possibilità, come si è visto, di rappresentare e memorizzare in forma riassuntiva il passato, e di conservare invece alcuni episodi nella loro interezza e permanentemente. Per quest'ultimo scopo può essere sufficiente un insieme di registri di memoria, una piccola biblioteca che può essere riempita e consultata nel corso del processo di composizione. Per il primo scopo, invece, potrebbe essere preso in considerazione, ad es., l'"automa a stati finiti" la cui memoria presenta caratteristiche abbastanza rispondenti alle

richieste avanzate.

(Automi a stati finiti.)

Un automa a stati finiti è infatti un dispositivo di analisi che riconosce determinati linguaggi formali, cioè determinati insiemi di stringhe di simboli su di un alfabeto finito, detti "linguaggi regolari"; per un automa a stati finiti in fase di analisi di una sequenza non è possibile una memoria completamente dettagliate del passato, ma tutti i possibili passati si raggruppano in classi di equivalenza (in numero finito) che corrispondono biunivocamente agli "stati" dell'automa. La memoria del passato è interamente contenuta (riassunta) nello stato attuale dell'automa.

Come la generazione di sequenze in base a matrici di probabilità condizionali e la misurazione dell'entropia della sorgente (caso C di fig. 13) da cui le sequenze provengono rappresentano le due facce, sintetica e analitica di uno stesso metodo, quello che abbiamo definito probabilistico, così un teorema dei linguaggi formali (4) stabilisce che lo strumento analitivo degli automi a stati finiti trova il suo corrispettivo sintetico nelle grammatiche generative che la classificazione di Chomskj definisce "regolari" (vedi la tabella di fig. 15): L'applicazione di tali grammatiche a processi di composizione automatica potrebbe rivelarsi efficace. Si potrebbe mostrare, utilizzando la rappresentazione dei "grafi di transizione", che l'adozione di una grammatica regolare come processo compositivo è perfettamente compatibile con la struttura della sorgente D, di cui stiamo discutendo.)

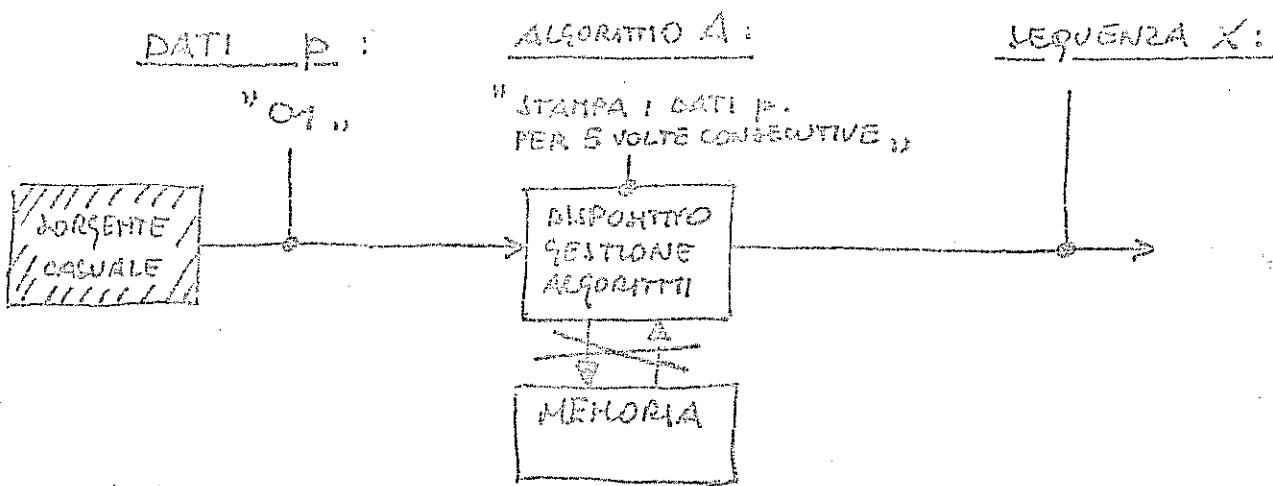
Algoritmi

La discussione di ciò che può avvenire all'interno di quello che nello schema della sorgente D (fig. 13) è stato chiamato "dispositivo di gestione degli algoritmi", è tutta da farsi. Per chiarire alcuni aspetti generali, senza pretendere qui di presentare progetti dettagliati, affrontiamo il discorso per approssimazioni successive, partendo dai casi più semplici.

- I^o approssimazione.

Il funzionamento più semplice del dispositivo di gestione degli algoritmi è quello per cui esso riceve in entrata i dati della sorgente casuale e li elabora in base a d un unico algoritmo, fisso, senza interagire con la memoria. La struttura della sequenza uscita dipenderà dai dati casuali e dall'algoritmo.

Ecco, in fig. 14, un semplice esempio :



- fig 14 -

Si dirà che "p è un programma che computa x in base ad A" e si indicherà questo dato con

$$x = A(p)$$

Questo tipo di funzionamento della sorgente consente di introdurre un metodo di analisi delle sequenze di gran lunga preferibile a quello della misurazione dell'entropia: la "varietà" di una sequenza non verrà più valutata in termini di entropia della sorgente, ma di complessità della sorgente stessa.

In base ad una definizione di A.N.Kolmogorov:

La complessità di una sequenza x rispetto ad un algoritmo A, indicata con $K_A(x)$, è la lunghezza della più corta sequenza binaria p (o "programma") che computa x in base ad A, cioè (5)

$$K_A(x) = \min \ell(p) ; A(p) = x$$

In riferimento all'es. precedente, ovviamente si avrà $K_A(x)=2$.

Ora, individuato il lato analitico del metodo algoritmico possiamo riassumere i metodi visti, nella loro doppia faccia sintetica-analitica, con la seguente tabella:

METODO	SINTESI	ANALISI	TIPO DI VALUTAZIONE
probabilistico	matrice delle probabilità condizionali	calcolo dell'entropia	CONTINUA
deterministico o algoritmico (caso particolare)	grammatiche regolari	automi a stati finiti	BINARIA
deterministico e algoritmico	algoritmi	complessità computazionale	DISCRETA

La definizione di complessità ha dato il via ad una serie di studi teorici che hanno portato ad una nuova definizione, detta "algoritmica", del concetto di casualità (6)

Una sequenza binaria è casuale se il più piccolo programma p che la computa in base all'algoritmo M (detto "universale" da Solomonov e "asintoticamente ottimale" da Kolmogorov) è lungo come la sequenza stessa, cioè quando la sua complessità (in assoluto) coincide con la sua lunghezza:

Per apprezzare i pregi delle definizioni di complessità e casualità appena date bastano poche considerazioni, tratte da un articolo di G.S.Chaitin (7).

"Si considerino le due successioni binarie

01010101010101010101
01101100110111100010

La prima è chiaramente costruita secondo una regola semplice da individuare: basta ripetere il numero 01 per 10 volte ... L'esame della seconda successione non permette di individuare uno schema generale di questo tipo. Non vi è nessuna regola che apparentemente presieda alla formazione del numero ...

La seconda successione di cifre binarie è stata generata lanciando una moneta per 20 volte e scrivendo 1 se risultava testa, 0 se risultava croce ... Il lancio di una moneta ripetuto per 20 volte può dare come risultato una qualsiasi di 2^{20} serie binarie equiprobabili. Quindi non dovrebbe essere più sorprendente ottenere con questo metodo la serie che mostra una chiara struttura invece di quella che sembra casuale; ognuna è un evento con probabilità 2^{-20} . Se l'unico indizio per stabilire la casualità di un evento probabilistico fosse la sua origine, allora entrambe le successioni dovrebbero considerarsi casuali e quindi anche tutte le altre subirebbero la stessa sorte, dato che la medesima procedura può generare ogni possibile successione. In conclusione, questo metodo non si dimostra fecondo allo scopo di distinguere il casuale dall'ordinato.

E' chiaro che occorre una definizione di casualità più raffinata, che non contraddica il concetto intuitivo di numero "senza schema". Una definizione di questo tipo è stata messa a punto solo negli ultimi 10 anni. In essa non si fa alcun riferimento all'origine del numero, ma ci si fonda solamente sulle caratteristiche della successione di cifre. La nuova definizione ci permette di descrivere le proprietà di un numero casuale con precisione superiore rispetto al passato e stabilisce una gerarchia di gradi di casualità."

Da queste considerazioni di Chaitin appaiono chiaramente quei pregi della definizione algoritmica di casualità che la rendono particolarmente indicata per le applicazioni musicali, primo fra tutti il fatto che si parli di maggiore o minore casualità delle frequenze, intendendo presenza, in maggiore o minor misura, di un determinato schema logico, schema che deve essere riconoscibili dall'uomo.

Il metodo algoritmico in generale, poi, presenta un altro grande vantaggio, quello di non richiedere come presupposto per la composizione o per l'interesse nell'ascolto un'competenza costituita dalle matrici delle probabilità condizionali, con i conseguenti limiti

pratici e concettuali che abbiamo visto. Infatti, se nel metodo probabilistico, la previsione dell'evento sonoro successivo che, secondo Meyer, (2) è una condizione necessaria per l'esistenza di significati musicali, si fonda, attraverso le probabilità condizionali, sulla conoscenza statistica di un gran numero di casi simili a quello del quale si cerca di prevedere la risoluzione, nel metodo deterministico (o algoritmico) la previsione dell'evento successivo si fonda esclusivamente sulla riconoscibilità di uno schema (e bastano pochissime note, come nell'esempio DO RE MI FA ...) e sulla inferenza di una possibile legge (o algoritmo) che lo genera.

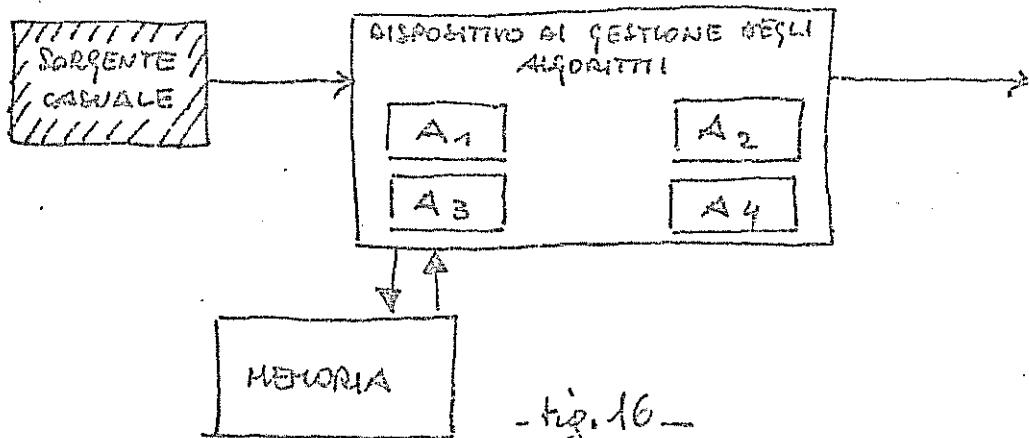
II approssimazione

Per complicare un po' il funzionamento del dispositivo di gestione degli algoritmi della sorgente D, basta attivare lo scambio di dati con la memoria.

In questo caso è possibile progettare, tra le tante possibilità che nascono, quella di particolari algoritmi che invece di creare nuove strutture, si limitino a operare semplici trasformazioni di strutture del passato, da prelevarsi dai registri di memoria già introdotti. Indicazioni di grande importanza a questo proposito sono fornite da un articolo di A. Bertoni, G. Haus, G. Mauri, M. Torelli (8). In esso il problema viene affrontato dal punto di vista puramente analitico, ma come prospettato dagli stessi autori, gli strumenti di analisi proposti potrebbero venire facilmente convertiti in strumenti di sintesi.

III approssimazione

Il funzionamento del dispositivo di gestione degli algoritmi si complica ulteriormente se l'algoritmo non è unico. Il modo più semplice di utilizzare diversi algoritmi è quello sequenziale:



- fig. 16 -

dove A_1, A_2, A_3, A_4 sono algoritmi prefissati.

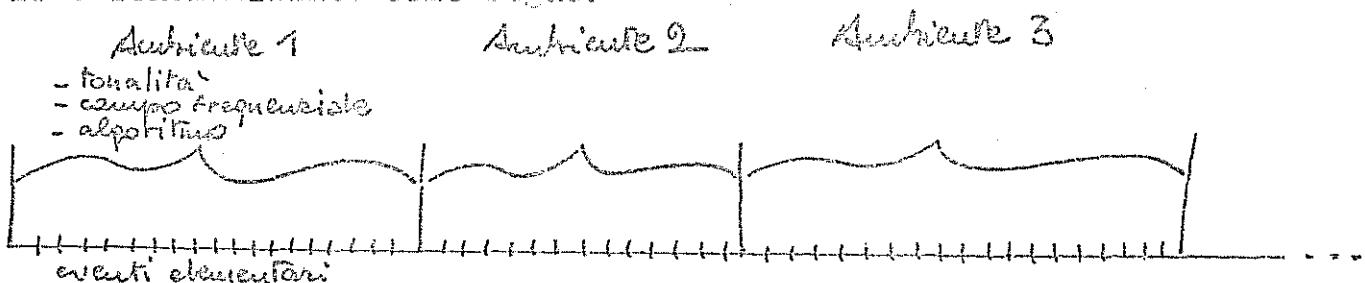
La sorgente casuale decide casualmente quale algoritmo usare provvisoriamente e per quanto tempo e fornisce ad esso i dati (o il programma) (fig. 15).

Le sequenze ottenute presenteranno diversi tipi di struttura in sequenza corrispondenti ai diversi algoritmi utilizzati e la sensazione dell'ascoltatore sarà quella di attraversare diversi ambienti.

Questo modo di procedere è simile a quello dei campi frequenziali provvisori, tipico dei programmi di composizione presentati (al posto dei campi abbiano gli algoritmi) d'altra parte non sarebbe difficile

individuare anche nella musica classica processi passibili di tale descrizione modellistica: si pensi ad es. alla successione di ambienti data dai cambi di tonalità.

Ciò è schematizzato come segue:



- fig 12 -

IV approssimazione

Non tutti gli algoritmi generano sequenze di eventi elementari, cioè ambienti; alcuni generano sequenze di ambienti, cioè ambienti di un livello gerarchico superiore.

Abbiamo discusso alcuni problemi relativi a due dispositivi (memoria e gestione degli algoritmi) interni della sorgente D di fig.13.

Rimane allora :

Sorgente casuale

Della sorgente casuale sappiamo tutto.

La sua presenza all'interno della sorgente D non deve sembrare in contrasto con l'aver definito "deterministico" (o algoritmico) il processo compositivo di cui l'intera sorgente D è modello. Il termine "deterministico" si riferisce infatti al comportamento degli algoritmi; viceversa i dati (o programmi) in entrata agli algoritmi sono casuali (con riferimento alla definizione algoritmica di casualità), e se non lo sono, possono essere ricondotti a programmi più brevi e completamente casuali.

Da qui la necessità della sorgente casuale.

Da quanto si è visto poi nella discussione sul dispositivo di gestione degli algoritmi risulta che la sorgente casuale può avere compiti differenti, come decidere casualmente la successione e la durata di utilizzo degli algoritmi, o fornire i programmi casuali per gli algoritmi preposti a tali decisioni.

Bibliografia

- 1) P.Grossi - G.Sonni : "Studi musicali: DGP- versione per il sistema 360/67". Pubbl. N°53 del CNUCE(6/1974)
- 2) Leonard B.Meyer :"Significato in musica e teoria dell'informazione" B.A.A.C. (giugno 1957), oppure su "Estetica e teoria dell'informazione" di aa.vv., Satelliti Bompiani (1972)
- 3) E.Coons e D.Kraehenbuehl:"l'informazione come misura di struttura in musica" su "Estetica e teoria dell'informazione" (vedi 2))
- 4) M.Aiello, A.Albano, G.Attardi,U.Montanari: "Teoria delle computabilità, logica, teoria dei linguaggi formali". Edit. TeS.Scien. Pisa (1976)
- 5) A.N.Kolmogorov: "Logical Basis for Information and Probability Theory" IEEE Trans ON Information Theory, IT14,N.5,1968
- 6) P.Martin Lof:"the definition of Random Sequences" Information and Control, 9, 1966
- 7) G.j. Chaitin: "Casualità e Dimostrazione Matematica" Le Scienze, N°65, (Settembre 1975)
- 8) A.Bertoni, G.Haus,G.Mauri,M.Torelli : "Compattazione di strutture informative nella descrizione di processi musicali". Congresso A.I.C.A. 1976, Milano

FAUSTO RA 221

"RELAZIONE COSTANTE TRA 'PRODUTTORI'
(municipi e riunisanti) e pubblico"

In questo intervento, che va considerato come l'apporto di un compositore e valutato quindi secondo l'angolazione che tale posizione comporta, vorrei anzitutto porre l'accento su di un fatto: nel rapporto musica/scienza - e quindi nell'interazione tra musicista e scienziato - una ricerca che non si ponga in relazione costante con coloro che dovranno confrontarsi con i risultati di tale lavoro può rischiare di trasformarsi in un'operazione totalmente inutile. Occorre quindi tener presente che i problemi non riguardano solamente la ricerca in sé o l'approntamento, pur necessario, di sedi atte alla ricerca ed all'ascolto, ma anche il continuo confronto tra "produttori" e pubblico: e sono dunque problemi di diffusione e sviluppo della cultura musicale, senza la quale è del tutto impensabile qualsiasi dialettica e la ricerca tende a perdgersi nella speculazione astratta, affermando se stessa col negare la realtà che la circonda.

L'idea che, trattando di musica, ci si ponga automaticamente in un ordine mentale diverso da quello di "scienza" nasce probabilmente dall'opinione molto diffusa secondo la quale il fatto musicale discende esclusivamente da un insieme di fattori irrazionali (l'ispirazione) dal quale si può far derivare a sua volta l'idea, anch'essa notevolmente diffusa, di una funzione "liberatoria" del la musica: ed è ovvio, invece, che tale mentalità va ricordotta ad una fondamentale disinformazione sulla musica, appunto, e sui suoi processi creativi. Ora, tale disinformazione è tanto più grave in quanto si riscontra anche presso molti musicisti: i quali magari ritengono possibile il binomio musica/scienza, ma lo intendono in un senso che, privilegiando il secondo termine, comporta automaticamente una connotazione limitativa del primo: e quindi una valutazione negativa del risultato.

Un atteggiamento di questo tipo esiste nei riguardi della musica prodotta mediante apparecchiature elettroacustiche, ed in particolare con l'aiuto di elaboratori. Le ragioni sono di ordine sia ideologico che pratico: l'idea di un lavoro con l'elaboratore è spesso associata a quella di un ritorno a concezioni filosofiche di tipo pitagorico, quasi che il compositore non possa sottrarsi ad una sorta di "misticismo del numero": o, in ogni caso, è arbitriamente ritenuta inscindibile da una astrazione a priori che impedirebbe programmaticamente il contatto con l'aspetto "materico" del suono e la possibilità di lavorare su di esso: escludendo quindi quel meccanismo di interazione tra compositore e materia sonora che è giustamente ritenuto uno degli elementi propulsivi dell'atto compositivo.

In tutte le considerazioni che concorrono alla formazione di un tale giudizio fondamentalmente negativo occorre però notare: 1) si contesta in tal modo al musicista una necessità che sarebbe assurda non ritenere fondamentale per chiunque, ossia la conoscenza del materiale: quasi che il comporre segnali sonori con una coscienza di ordine anche fisico/acustico possa precludere il momento irrazionale; 2) la valutazione sulla validità della computmusica è basata su quanto in esemplare è stato finora realizzato

in questo campo; ma se il livello è in generale abbastanza modesto dal punto di vista musicale, le cause riguardano piuttosto la mentalità ed i modi di produzione che non i mezzi; 3) l'atteggiamento dei musicisti deriva spesso dalla sensazione - più o meno inconfessata - che accettare questo tipo di tecnica comporterebbe un cambiamento di posizione da parte del compositore: in sostanza, ciò cui ci si oppone è il salto di mentalità, che renderebbe ovviamente necessario l'abbandono di tutto un modo di considerare la musica, ivi comprese le acquisizioni tecniche derivanti dagli studi e dall'esperienza; ma a quest'ultima obiezione in particolare si oppone l'esigenza manifesta di tutta la nostra musica dal 1900 in poi: e basterebbe citare al riguardo le posizioni illuminanti di Busoni.

Ci sono dei grossi ostacoli: il più evidente è rappresentato dall'estrema difficoltà nel poter disporre di laboratori elettroacustici per una ricerca ed una produzione musicale. La situazione italiana è caratterizzata dall'esistenza di un solo studio pubblico - ossia quello di Fonologia della RAI a Milano -, ove peraltro si accede con difficoltà, per tempi di lavoro generalmente del tutto insufficienti, e in ogni caso solo per un'attività di produzione, coi attrezzi di tipo tradizionale.

Occorre quindi pensare a strutture specifiche - ovviamente senza che ciò significhi istituire Centri di dimensioni paragonabili a quelle dell'IRCAM a Parigi - le quali, utilizzando le possibilità di sviluppo di strutture già esistenti, possano costituire un momento di aggregazione per la creazione di nuove competenze.

Mi sembra però fondamentale ribadire che il nodo centrale per la soluzione della questione è comunque la corretta impostazione del rapporto tra gli operatori interessati alla ricerca: non è sufficiente infatti disporre di strutture adeguate, com'è dimostrato appunto dai risultati finora ottenuti nel campo della computer music, di scarso interesse da un punto di vista musicale poiché probabilmente riconducibili ad una eccessiva presenza dei tecnici nei confronti dei compositori.

Naturalmente va subito detto che il modo di operare, diverso, - e già configurabile come lavoro di ampi gruppi, al di là della pura produzione musicale, per le implicazioni che comporta e per le connessioni con altri campi di ricerca, quali ad esempio la psicoacustica o la diffusione a livello generalizzato - dovrà senza dubbio comportare una diversa posizione mentale da parte del musicista: anche se non nel senso che il compositore debba acquistare il grado di preparazione specifica che è proprio del fisico. (Ciò, almeno, per quanto riguarda un periodo iniziale che non potrà certamente essere breve).

Un altro concetto egualmente importante mi sembra sia quello del rapporto tra compositore e "strumento": lungi dal risolversi da parte dell'uomo in una passiva accettazione (o al massimo in una scarsa condizionata) di soluzioni derivate da processi puramente automatici, dovrà essere inteso secondo una posizione storicamente più corretta: ossia nel senso di un suo offerto dalla macchina per estendere la ricerca oltre i limiti degli strumenti tradizionali, ma senza che tale intervento sancisca l'abdicazione dell'uomo,

il quale rinunzierebbe a trasmettere ad altri il suo pensiero. Ed è forse il caso di aggiungere che la possibilità di lavorare con programmi estremamente flessibili rimetterà in gioco quel meccanismo di interazione tra compositore e materia sonora che il tipo stesso di operazione sembrava in un primo tempo aver escluso.

Voglio ancora accennare a due questioni: la prima è quella connessa con l'idea che le nuove apparecchiature offrano "un campo infinito di possibilità": è vero infatti che siamo ora in grado di superare tutti i limiti che finora condizionavano il nostro concetto di "musica": per esempio l'altezza del suono (non più vincolata agli strumenti tradizionali) o la durata dei fenomeni (del tutto indipendente dalle capacità fisiologiche degli esecutori); e che inoltre da tempo ormai - Stravinskij Bartók Varèse - è stata abbattuta la barriera che separando il "suono" dal "rumore", precludeva a quest'ultimo il diritto di esistenza in musica: ciò significa però che le strutture sonore - che erano costituite da materiali organizzati secondo una convenzione linguistica pluriscolare (il sistema armonico/tonale, o almeno ciò che ne restava, la scala temperata) - sono state organizzate, da un certo momento in poi, con l'impiego di un materiale riferito ad una qualsiasi convenzione [arbitraria] e addirittura usato abolendo apparentemente del tutto ogni forma di convenzione.

A queste condizioni la trasmettibilità del messaggio è divenuta assai problematica, per l'opinabilità o per l'assenza di un codice di riferimento.

Si potrebbe a questo punto ipotizzare un'organizzazione del materiale basata sulla diversità di situazioni registrata dal cervello all'aspetto comparativo di fenomeni che presentano un diverso grado di complessità: è ovvio che tali fenomeni sarebbero in gran parte sconosciuti, in quanto - da un certo grado di complessità in poi - la non pertinenza con i codici di riferimento finora utilizzati ne renderebbe difficile la riconoscibilità. Ma se consideriamo che il nostro grado di sensibilità percettiva è presumibilmente in rapporto diretto con il mondo sonoro con il quale siamo stati finora in contatto (in altre parole, noi sentiamo solamente ciò che siamo abituati a sentire, rifiutando tutto quanto oltrepassa quello che riteniamo rappresentanti il normale limite di utilizzazone), non possiamo non tener conto di due fatti: 1) l'affinamento di sensibilità si è sinora sempre verificato solo all'interno dei limiti propri degli strumenti tradizionali; 2) qualità degli strumenti di cui possiamo ora disporre ci permette invece di valicare tali limiti in termini di estrema precisione, sì che il superamento non si identifichi con un'incurseione in un infinito caos. Ma allora, per ottenere un affinamento di sensibilità al di fuori dei limiti tradizionali, si può ipotizzare un capovolgimento del rapporto uomo/macchina: non nel senso di una prevaricazione da parte dello strumento ma affidando ad esso il compito di "guidare" l'uomo nel riconoscimento di valori che egli all'inizio può stabilire - da un certo grado di complessità in poi - solo teoricamente: una serie di valori non basata su convenzioni arbitrarie ma corrispondente a differenze, ordinatamente crescenti, di situazione fisica e quindi, in seguito, di effetti.

Tutto ciò, ovviamente, non potrà non richiedere che tempi di sen-

sibilizzazione molto ampi ("coscienziosi e lunghi esperimenti e una continua educazione dell'orecchio", secondo l'espressione di Busoni)

La seconda questione riguarda i modi di inserimento della voce umana (intendendo voce "dal vivo" e non voce registrata, manipolata o sintetizzata). Ritengo assurdo pensare che essa abbia esaurito la sua funzione: e non tanto da un punto di vista puramente tecnico - ossia per le caratteristiche di strumento più di ogni altro flessibile e ricco di possibilità - quanto invece considerando l'improprietà dell'esclusione di un mezzo di espressione che è interno all'uomo e il cui impiego nasce dalla necessità stessa di esprimersi: che potrebbe dunque rappresentare nel rapporto con i nuovi strumenti l'unica forma di mediazione per un'operazione (la "musica") la quale - pur nei casi di una sua dimensione più astratta - ha sempre richiesto, a diversi livelli e in diversi modi, la partecipazione viva dell'uomo.

Fausto Razzi

GIOVANNI RAVASI - VITO OLZOLA

Hardware e software per la composizione
musicale automatica in tempo reale.

Milano, 14 dicembre 1977

1. Hardware. Purtroppo è solo allo stato di progetto, ma questo progetto è rifinito in ogni particolare e attende per l'inizio della realizzazione solo i fondi a ciò eventualmente destinati dal C.N.R.

Il problema della generazione dei suoni è stato risolto con l'impiego di un organo elettronico con le caratteristiche seguenti: due manuali di 4 ottave ciascuno, pedaliera di una ottava, 20 registri reali, 10 registri d'effetto, 7 comandi di accoppiamento (IIM-IIM, IIM-P, IIM-P, e altri), espressione (=volume).

Il canale diretto del computer chiude tutti e soli i circuiti dei tasti dei manuali e della pedaliera. I registri (reali, d'effetto, d'accoppiamento) e l'espressione sono comandati manualmente nel corso dell'esecuzione, eventualmente su indicazione di messaggi provenienti dal computer correlati agli stati relativi al processo compositivo.

Il computer è l'Olivetti P6060 già disponibile e funzionante, con le seguenti caratteristiche: 16 K di memoria + 32 K da acquistare, stampante-plotter, introduzione istruzioni e dati a tastiera con controllo visivo, memoria ausiliaria a dischi.

L'accoppiamento computer-organo è progettato dalla filiale bresciana della Olivetti e realizzato in collaborazione con ditte specializzate in problemi di intervento computerizzato nel governo di macchine utensili e di impianti analoghi anche di maggior dimensione.

2. Scopo

Lo scopo del progetto è la composizione in tempo reale. Con questo termine si intende quanto segue.

Il tempo fisico viene suddiviso in intervalli di ampiezza costante, per esempio 4 secondi, ma con possibilità di variare questo parametro anche fino a 64 secondi. (L'intervallo di 4 secondi corrisponde ad una battuta di quattro quarti a metronomo $\text{♩} = 60$). Il passaggio da un intervallo all'altro viene segnalato all'operatore con messaggi output con funzione di metronomo.

In ogni intervallo di tempo I_r l'esecutore introduce un messaggio M_r che viene elaborato nel tempo I_{r+1} . I dati output, corrispondenti a 4 secondi o più di esecuzione musicale a 4 voci, vengono registrati in memoria centrale (o su disco) secondo una codifica prefissata, in seguito chiamata Codass. Il tempo di elaborazione deve essere inferiore all'ampiezza di ogni intervallo I_r . Al tempo I_{r+2} la codifica della stringa composta al tempo I_{r+1} è registrato in memoria alla fine di tale tempo viene decodificata da una subroutine su disco e tradotta in segnali azionanti l'organo, al tempo I_{r+2} stesso. È evidente la necessità di disporre di due banchi di memoria e di due zone disco, ed anche la necessità di collegare i banchi di memoria, con le zone disco incrociando i collegamenti a tempi I_r , I_{r+1} alterni. Al tempo I_1 hanno luogo solo ingressi di messaggi nel computer, al tempo I_2 hanno luogo solo ingressi di messaggi ed elaborazione complessiva. Dal tempo I_3 in poi il funzionamento entra in regime. Analogico processo, ma in senso inverso, ha luogo alla fine dell'esecuzione. È previsto il caso in cui l'operatore sbagli nell'impartire istruzioni, o si riservi tempo per "cambiare rotta". In questo caso l'elaborazione proseguirà in base a messaggi standard memorizzati preventivamente e interpretati dal computer in modi variati ma stilisticamente e strutturalmente conformi alla precedente composizione. Questa possibilità risulta da quanto detto nel successivo § 5.

3. Scrittura dei risultati

Per il controllo critico approfondito dei risultati è previsto che la codifica Codass dei dati output possa essere interpretata ed espressa con la usuale notazione su pentagramma. La relativa subroutine è, anzi, già preparata e collaudata.

Essa permette di scrivere direttamente i dati output su quattro righe musicali sovrapposti di tipo tradizionale. La scrittura avviene con l'impiego dei simboli # e ↩ usuali riportati (solo gli uni e solo gli altri) in corrispondenza ad ogni nota alterata.

Alla notazione con accidenti in chiave provvede il trascrittore a mano, che non deve fare altro che ricalcare a penna piccoli segni fatti sul rigo dalla stampante, completandoli con i grafemi di durata. Il computer esprime le durate solo distanziando più o meno, secondo regole, fissate, i segni di nota, che così possono essere incolonnati correttamente nei quattro righi.

E' prevista la scrittura fino in quattro chiavi (violino, alto, tenore basso) e la possibilità di trasporti in tutte le tonalità per l'eventuale esecuzione con strumenti traspositori.

La tecnica di scrittura consente l'aumento teoricamente indefinito del numero delle parti scrivibili, senza aggravio di impiego di memoria: infatti si possono comporre separatamente le voci a quattro e quattro e riportarle poi coerentemente in una stessa partitura. Quest'ultima possibilità è tuttavia subordinata al possesso della tecnica compositiva di cui si parla nel § 5.

4. Aspetti pratici delle esecuzioni

La struttura generale del piano indica la possibilità di lavorare nel l'immediato al fine di costruire un vero e proprio strumento musicale di nuova specie: questo non produce suoni su comando delle dita o delle labbra dell'esecutore, ma produce suoni su comandi espressi da messaggi simbolici, corrispondenti a sintagmi musicali di contenuto semanticamente anche molto elevato, come si dice nel § 5.

Nel caso degli strumenti tradizionali il pensiero musicale risiede tutto nella mente dell'esecutore, anzi risiede quasi tutto nella mente del compositore, mentre l'esecutore lo ricostruisce e lo interpreta senza incrementi sintatticamente significativi.

Invece nel caso del "sintagmofono", se così lo si può chiamare, il pensiero musicale viene trasferito allo strumento in fasi successive per modo che al momento dell'esecuzione si può dire alla lettera che lo strumento possiede una parte non trascurabile del pensiero musicale che il compositore gli ha preventivamente somministrato.

Lo strumento possiede questo pensiero sottoforma simbolica di subroutine

ne, memorizzate e pronte a interferire in tempo reale coi messaggi in arrivo per l'esecuzione vera e propria.

Quanto alla successione di questi messaggi essa può essere il risultato di precedenti elaborazioni (vedi § 5), può essere improvvisata da un esecutore virtuoso che possieda perfettamente la tecnica compositiva, o può essere il frutto combinato di precedenti elaborazioni e di scelte estemporanee, come nella pratica musicale del jazz tradizionale o delle cadenze nei concerti solistici del periodo pre-romantico.

La nascita di uno strumento provoca di solito la comparsa dei relativi virtuosi. Ma questi diverranno tali con uno studio di specie diversa da quella tradizionale: dovranno impadronirsi della conoscenza di elevate strutture informatiche, come appare dal successivo § 5. L'Istituto Matematico di Brescia si impegna a rendere noti i risultati degli esperimenti, anche se negativi, in ogni caso parziali, entro un anno dalla data di acquisto delle apparecchiature mancanti. Si precisa che il computer è già disponibile e funzionante, e che le subroutine che riguardano le parti del piano impegnanti il solo computer sono in via di avanzata realizzazione.

Tuttavia un giudizio completo sul valore informatico e musicale del piano non può essere dato fino a che non si potrà ascoltare il prodotto finale in forma sonora.

Il poco peso ed ingombro delle apparecchiature rende del tutto praticabile l'idea di esecuzioni pubbliche, con o senza aggiunta di strumentisti "diretti" dal computer stesso, o in tempo reale per via di messaggi output, o per mezzo di parti scritte in precedenza.

5. La tecnica compositiva

La parte concettuale più delicata del piano è quella che riguarda la tecnica compositiva da impiegare.

Va anzitutto precisato che qualunque progetto compositivo avente i dati output codificati (p. es. su disco) secondo Codass può essere tradotto in grafia o in fonie con l'hardware e il software progettati. In tal modo sono da prevedere in pochissimo tempo le realizza-

zioni di progetti compositivi già completati per la parte concettuale da alcuni anni (§ 6) ma non rifiniti dal punto di vista editoriale a causa di difficoltà nell'uso di generatori di suoni progettati in precedenza e realizzati solo in maniera pionieristica.

La flessibilità e la potenza delle apparecchiature in progetto rendono del tutto praticabile l'idea di affrontare gli aspetti informatici e linguistici della composizione musicale in una maniera radicalmente diversa sia dalla pura e semplice invenzione fantastica di nuove tecniche sia dalla pura analisi strutturale con metodi statistici del linguaggio musicale tradizionale.

La prima maniera è indiscutibilmente valida sul piano della creazione artistica ma è poco credibile sul piano del rigore scientifico, la seconda maniera, reciprocamente, è carica di credenziali scientifiche ma dà luogo a gravissime difficoltà nella costruzione di valori musicali essenzialmente nuovi.

La via che si propone è alquanto diversa, e per nulla intermedia, rispetto alle due dette qui sopra.

Secondo il progetto che qui si propone, e che costituisce il nucleo centrale della presente nota, si immagina che ogni breve cella musicale I_r sia dominata dalla presenza contemporanea di un numero finito di eventi.

Il Computer è fornito di un algoritmo per la traduzione della presenza degli eventi E_1, E_2, \dots, E_N in una stringa concreta a quattro veci, di breve durata (una battuta, quattro battute, una serie, un minuto di tempo, a seconda della struttura compositiva prescelta).

Gli eventi da tradurre in musica possono essere di svariata natura e devono essere scelti in un insieme di eventi astratti che la teoria e la pratica musicale devono indicare come significativi. A titolo di esempio si possono indicare i seguenti insiemi di eventi. Questi insiemi sono gli insiemi dei valori assumibili dalle variabili E_R al tempo t ($R = 1, 2, \dots, N$).

- a) - Per la musica di struttura tradizionale: scala, tonalità, grado fondamentale per ogni voce, schema di ritmo, schema di volta e di imitazione, ...
- b) - Per la musica seriale: serie prescelta in una rosa finita, sue trasformate per inversione e retrogradazione, schemi di ritmo, spezzamenti e ricostruzioni di serie.
- c) - Per la musica di tipo "postweberniano" l'elenco si restringe e nello stesso tempo si allarga: i "punti" e le loro variazioni sono un materiale più facilmente governabile con il computer che con la mente.

Un campo sterminato sembra essere quello della musica multitorale, atonale, dodecafonica e anche della musica antica dominata dai modi e dalle scale anzichè dalla tonalità (musica medievale, grégoriana, ellenica, di tradizione rinascimentale o fiamminga ...).

L'algoritmo per la traduzione di un vettore di eventi astratti in una stringa concreta non è che il primo passo di un procedimento ricorsivo a rovescio molto complesso e già studiato con discreta profondità. Infatti la ricomparsa di un evento o del legame di un evento con un altro è un altro evento di livello più alto. Detti eventi di livello zero quelli codificati in Codass, si possono dire di livello uno quelli codificati nella successione di vettori (catena)

$$(1) \quad E_{11}, E_{12}, \dots E_{1R_1}$$

e allora la struttura di questa successione può essere descritta con la successione (di solito più breve) di eventi di livello due

$$(2) \quad E_{21}, E_{22}, \dots E_{2R_2},$$

accompagnata da un algoritmo per tradurla nella (1). Il processo si riapplica alla (2) fino a pervenire ad una successione di eventi di livello massimo (sei, sette, otto, finora) costituente per così dire la più penetrante "radiografia" ^{alla catena} del pezzo composto.

Da un certo punto di vista può essere significativo che gli algoritmi per passare da una catena più bassa siano fra loro omogenei per struttura e addirittura che siano solo casi particolari di

un unico algoritmo

ritmo (algoritmo delle catene, con spiccate somiglianze con le ordinarie algebre), ma sembra bene non procedere con troppa fiducia nelle complicazioni formali puramente matematiche.

Il valore concettuale dei risultati è sperimentale e non formale. Perciò si ritiene prudente non descrivere più in dettaglio gli algo ritmi finora immaginati per costruire una catena di livello uno e così tradurla in un pezzo musicale. Gli esperimenti non dicono ancora fino a che livello arrivano le catene che possono essere elaborate entro il computer quando si opera in tempo reale (§ 2), ma si può prevedere che i messaggi in entrata in tempo reale possano essere quelli relativi alle catene di livello tre o forse quattro. Le catene di livello due, uno, zero sono calcolate deterministicamente dalla macchina. Quelle di livello più alto sono o risultato di precedenti elaborazioni automatiche o sono elaborate mentalmente dal compositore-esecutore.

Il loro possesso equivale alla dottrina e alla cultura musicale da lui posseduta.

Le catene di livello più alto sono infatti il riscontro schematico delle massime libertà creative del compositore stesso. Il metodo della catena sembra correlato in modo profondo a certi risultati precedenti (Selzi) che sembrano applicabili addirittura alla cibernetica delle reti nervose e in particolare dell'area corticale uditiva.

6. Precedenti attività degli autori in computer music

Il presente progetto è nato abbastanza spontaneamente come evoluzione delle prime composizioni musicali automatiche, che hanno fruttato agli autori di questa nota, in collaborazione con R.Lanza, il secondo posto nella graduatoria vincente del concorso internazionale di composizione musicale bandito dalla Società Italiana di Musica contemporanea nel 1971. In quell'occasione l'hardware usato fu quello del Centro di Calcolo dell'Università di Milano, con generatore di suoni costruito direttamente da M. Lunelli. Alcuni risultati raggiunti da allora ma non pubblicati saranno riedicati fonicamente appena realizzato il nuovo hardware.

MARTISSA ENZO

"ANALISI E SINTESI DI PROCESSI
PSEUDO-MUSICALI"

Estratto dalla Tesi di Laurea:

"Analisi e sintesi di processi pseudo-musicali",
Enzo Martisso -1977- Istituto di Cibernetica, Milano.

Nell'ambito della musica contemporanea riveste una grande importanza la musica seriale, i cui principi innovativi sono stati introdotti da Arnold Schoenberg , (1874-1951) e dalla scuola dodecafonica nel terzo decennio del secolo, ripresi e generalizzati dalle scuole successive (postweberiane) con la loro applicazione a tutti i parametri sonori: altezza, durata, intensità, timbro, fino ad interessare la struttura complessiva dell'opera (P. Boulez).

La cellula base di una composizione dodecafonica è la serie dodecafonica. Con questo termine si intende una successione delle dodici note della scala temperata, che presenta la massima varietà possibile nelle altezze (frequenze) all'interno di un'ottava .

La serie dodecafonica è quindi una permutazione di dodici elementi distinti.

Dal concetto musicale di serie si passa a quello di serie equilibrata (+) imponendo la massima varietà di intervalli oltre che di altezza.

Si definisce questa serie come una particolare permutazione nell'insieme degli interi dotato della proprietà di congruenza.

(+) Secondo la terminologia adottata da A. Riotte

Serie equilibrate. Definizioni.

Def. 2.1. Si definisce scarto (mod n) tra due interi a, b il numero $s \equiv b - a \pmod{n}$ con $0 \leq s < n$.

Ad ogni successione di m interi a_1, a_2, \dots, a_m è possibile quindi associare una successione di $m-1$ scarti s_1, s_2, \dots, s_{m-1} dove

$$s_k \equiv a_{k+1} - a_k \pmod{n} \quad \text{con } k=1, 2, \dots, m-1$$

in altri termini:

$$a_2 \equiv a_1 + s_1 \pmod{n}$$

$$a_3 \equiv a_2 + s_2 \equiv a_1 + s_1 + s_2 \pmod{n}$$

$$\dots \dots \dots \\ a_{k+1} \equiv a_1 + \sum_{i=1}^k s_i \pmod{n} \quad k=1, 2, \dots, m-1$$

Def. 2.2. Una successione a_1, a_2, \dots, a_n che formi un sistema completo modulo n si definisce serie equilibrata di ordine n se l'insieme s_1, s_2, \dots, s_{n-1} degli scarti unito all'elemento $s_n = 0$ è un sistema completo modulo n.

In tal caso la successione degli scarti s_1, s_2, \dots, s_{n-1} si definisce serie di scarti equilibrata di ordine n.

Indicheremo nel seguito con \underline{a} una serie equilibrata e con \underline{s} la serie di scarti relativa ad \underline{a} .

Identificando le n classi di resti modulo n con gli interi $1, 2, \dots, n$ si può descrivere una serie equilibrata come una permutazione a_1, a_2, \dots, a_n dei primi interi positivi, in cui la serie di scarti relativa sia anch'essa una permutazione dei primi $n-1$ interi positivi:

Teorema 2.1. Una successione a_1, a_2, \dots, a_n di interi positivi minori od eguali a n è una serie equilibrata se e soltanto se, indicata con s_1, s_2, \dots, s_{n-1} la successione degli scarti, valgono le due proprietà:

- $a_j = a_k$ sse $j = k$ $j, k = 1, 2, \dots, n$
- $s_j = s_k$ sse $j = k$ $j, k = 1, 2, \dots, n-1$.

Una serie di scarti s che sia una permutazione dei primi $n-1$ interi positivi è una serie di scarti equilibrata se nella serie a relativa ad s la differenza modulare tra due elementi qualsiasi è sempre diversa da zero. Un corollario del teorema 2.1 è allora il seguente:

Teorema 2.2. Sia s_i , $i=1, 2, \dots, n-1$ una permutazione dei primi $n-1$ interi, allora s_i è una serie di scarti equilibrata se e solo se valgono le relazioni:

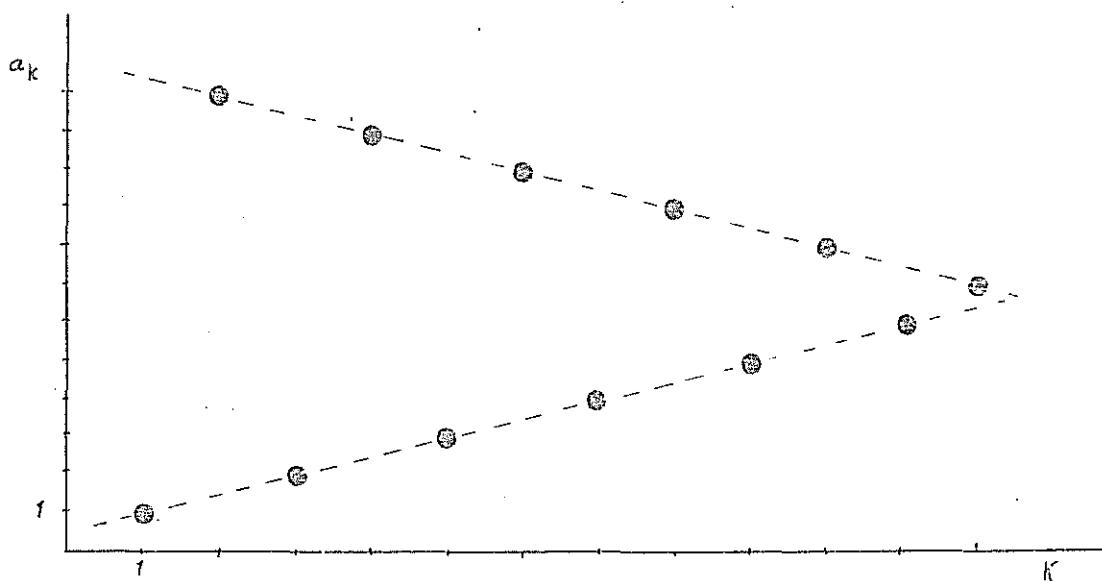
$$1) \sum_{i=1}^j s_i \not\equiv 0 \pmod{n} \quad j=1, 2, \dots, n-1$$

$$2) \sum_{i=1}^j s_i - \sum_{i=1}^k s_i \equiv 0 \pmod{n} \text{ sse } j=k \\ j, k = 1, 2, \dots, n-1.$$

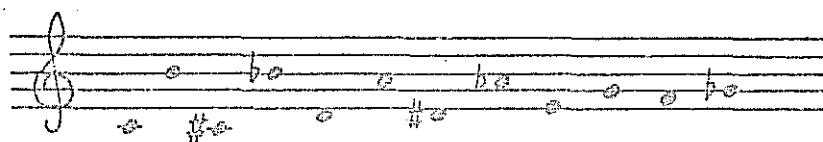
Si osservi come una serie di tipo $s_k = (-)^k k$, ad esempio, per $n=12$, la

$$\begin{array}{cccccccccc} s & = & 11 & 2 & 9 & 4 & 7 & 6 & 5 & 8 & 3 & 10 & 1 \\ a & = & 1 & 12 & 2 & 11 & 3 & 10 & 4 & 9 & 5 & 8 & 6 & 7 \end{array}$$

raffigurata in un diagramma a_k / k , presenta una particolare struttura grafica che viene percepita visivamente come un insieme di due linee convergenti:



La struttura di questa serie viene riconosciuta in modo analogo anche all'ascolto, e si presenta come l'alternanza di due scale per semitonni ben distinte, una ascendente e l'altra discendente:



Operatori sulle serie. Definizioni.

Sia $\underline{a} = a_1, a_2, \dots, a_m$ una successione di m interi qualunque,

Def. 3.1. Si indica ${}^n T_k$ l'operatore di trasposizione k modulare ad n (in termini matematici si tratta di una traslazione modulare), così definito:

$$({}^n T_k(\underline{a}))_i \equiv k + a_i \pmod{n}.$$

Ad esempio, per $m=n=8$

$$\begin{aligned} \underline{a} &= 1 & 2 & 4 & 7 & 3 & 8 & 6 & 5 \\ {}^8 T_1(\underline{a}) &= 2 & 3 & 5 & 8 & 4 & 1 & 7 & 6 \end{aligned}$$

Def. 3.2. Si indica con R l'operatore di retrogradozione (trasposizione) che opera nel modo seguente:

$$(R(\underline{a}))_i = a_{m+1-i}.$$

Ese., $m=8$,

$$\begin{aligned} \underline{a} &= 1 & 2 & 4 & 7 & 3 & 8 & 6 & 5 \\ R(\underline{a}) &= 5 & 6 & 8 & 3 & 7 & 4 & 2 & 1 \end{aligned}$$

Vale chiaramente

$$RR(\underline{a}) = \underline{a}$$

quindi

$$R^{-1} = R.$$

Def. 3.3. Si indica con ${}^n I_k$ l'operatore di inversione speculare rispetto a k , modulo n , definito nel modo seguente:

$$({}^n I_k(\underline{a}))_j \equiv_{k-} a_j \pmod{n}$$

Ese. $m=8$:

$$\begin{array}{ccccccccc} \underline{a} & = & 1 & 2 & 4 & 7 & 3 & 8 & 6 & 5 \\ {}^8 I_8(\underline{a}) & = & 7 & 6 & 4 & 1 & 5 & 8 & 2 & 3 \end{array}$$

Dalla definizione si ottiene $I_k^{-1} = I_k$.

Def. 3.4. Si indica con ${}^n M_k$ l'operatore di moltiplicazione modulare ad n così definito:

$$({}^n M_k(\underline{a}))_j \equiv_k a_j \pmod{n}$$

Ese.:

$$\begin{array}{cccccccccc} \underline{a} & = & 1 & 2 & 4 & 3 & 8 & 5 & 9 & 7 & 10 & 6 \\ M_7(\underline{a}) & = & 7 & 4 & 8 & 1 & 6 & 5 & 3 & 9 & 10 & 2 \end{array}$$

Si osservi come questi operatori, con l'esclusione di T_k , siano commutabili, si ha infatti:

$$T_k I_n = I_n T_{n-k} = I_k$$

Si verifica quindi che le differenti combinazioni degli operatori T_k , R e I_k , i tradizionali operatori di trasposizione, retrogradazione, inversione, di origine musicale, sono $4 \cdot n$.

Serie centrate e simmetriche.

Def. 3.6. Si definisce serie centrata, e la indicheremo con \underline{c} , una serie equilibrata con la proprietà di avere l'elemento centrale della serie di scarti uguale a $n/2$:

$$s_{n/2} = n/2.$$

Il sottoinsieme delle serie centrate contiene a sua volta due sottoinsiemi con particolari caratteristiche di simmetria, legate alle proprietà degli operatori R e X rispettivamente:

Def. 3.7. Si definisce serie (di scarti) R -simmetrica (9) \underline{c}_R , una serie (di scarti) equilibrata che goda della seguente proprietà:

$$s_i + s_{n-i} = n \quad \text{con } i=1, 2, \dots, n/2$$

In particolare si ha

$$s_{n/2} + s_{n/2} = n$$

quindi la serie è anche centrata.

Es.

$$\underline{s} = 1 \ 2 \ 4 \ 7 \ \underline{5} \ 3 \ 6 \ 8 \ 9$$

è una serie di scarti R -simmetrica.

Teorema 3.1. Sia \underline{s} una serie di scarti R -simmetrica.

Vale allora

$$R(\underline{s}) = I_n(\underline{s}).$$

Si ha infatti:

$$(R(\underline{s}))_i = s_{n-i} = n - s_i$$

$$(I_n(\underline{s}))_i = n - s_i$$

Pertanto una serie R-simmetrica è invariante per l'applicazione della coppia di operatori $R I_n$.

Def. 3.8. Si definisce serie (di scarti) X-simmetrica \underline{s}_x , una serie (di scarti) equilibrata con la proprietà:

$$s_i + s_{n/2+i} = n \quad i=1, 2, \dots, n/2$$

Ricordando quanto detto sopra, la condizione

$$s_{n/2} + s_n = n$$

equivale a $s_{n/2} = n/2$.

Es. $n=10$:

$$\begin{array}{cccccccccc} \underline{s} & = & 1 & 2 & 6 & 3 & 5 & 9 & 8 & 4 & 7 \\ \underline{a} & = & 1 & 2 & 4 & 10 & 3 & 8 & 7 & 5 & 9 & 6 \end{array}$$

Teorema 3.2. Sia \underline{s} una serie di scarti X-simmetrica.

Allora vale:

$$X(\underline{s}) = I_n(\underline{s}).$$

Infatti

$$(X(\underline{s}))_i = s_{j+i} = s_{n/2+i} = n - s_i$$

$$(I_n(\underline{s}))_i = n - s_i$$

Ad esempio, una serie che non possiede proprietà di simmetria individua una classe di equivalenza cui appartengono $4 \cdot \varphi(n)$ elementi distinti. Es. $n=10$ ($\varphi(n)=4$):

$$\underline{s} = 1 \ 2 \ 9 \ 5 \ 7 \ 4 \ 8 \ 3 \ 6$$

$$X(\underline{s}) = 7 \ 4 \ 8 \ 3 \ 6 \ 5 \ 1 \ 2 \ 9$$

$$R(\underline{s}) = 6 \ 3 \ 8 \ 4 \ 7 \ 5 \ 9 \ 2 \ 1$$

$$M_{n-1}(\underline{s}) = 9 \ 8 \ 1 \ 5 \ 3 \ 6 \ 2 \ 7 \ 4$$

$$M_3(\underline{s}) = 3 \ 6 \ 7 \ 5 \ 1 \ 2 \ 4 \ 9 \ 8$$

$$M_7(\underline{s}) = 7 \ 4 \ 3 \ 5 \ 9 \ 8 \ 6 \ 1 \ 2$$

$$RM_{n-1}(\underline{s}) = 4 \ 7 \ 2 \ 6 \ 3 \ 5 \ 1 \ 8 \ 9$$

$$XM_{n-1}(\underline{s}) = 3 \ 6 \ 2 \ 7 \ 4 \ 5 \ 9 \ 8 \ 1$$

$$RM_3(\underline{s}) = 8 \ 9 \ 4 \ 2 \ 1 \ 5 \ 7 \ 6 \ 3$$

$$XM_3(\underline{s}) = 1 \ 2 \ 4 \ 9 \ 8 \ 5 \ 3 \ 6 \ 7$$

$$RM_7(\underline{s}) = 2 \ 1 \ 6 \ 8 \ 9 \ 5 \ 3 \ 4 \ 7$$

$$XM_7(\underline{s}) = 9 \ 8 \ 6 \ 1 \ 2 \ 5 \ 7 \ 4 \ 3$$

$$XR(\underline{s}) = 9 \ 2 \ 1 \ 5 \ 6 \ 3 \ 8 \ 4 \ 7$$

$$M_{n-1}XR(\underline{s}) = 1 \ 8 \ 9 \ 5 \ 4 \ 7 \ 2 \ 6 \ 3$$

$$M_3XR(\underline{s}) = 7 \ 6 \ 3 \ 5 \ 8 \ 9 \ 4 \ 2 \ 1$$

$$M_7XR(\underline{s}) = 3 \ 4 \ 7 \ 5 \ 2 \ 1 \ 6 \ 8 \ 9$$

Osserviamo come la composizione degli operatori R ed X sia equivalente alla retrogradazione delle due sottostringhe della serie di scarti che precedono e seguono lo scarto $n/2$.

Sempre per $n = 10$, da una serie R-simmetrica ($M_9(\underline{s}) = R(\underline{s})$) si verifica che la classe relativa contiene solo $2\varphi(n) = 8$ elementi (ricordiamo che, per $n = 10$, $M_3 M_9 = M_7$, $M_7 M_9 = M_3$):

$$\underline{s} = R M_9(\underline{s}) = \begin{array}{cccccccccc} 1 & 2 & 4 & 7 & 5 & 3 & 6 & 8 & 9 \end{array}$$

$$X(\underline{s}) = X R M_9(\underline{s}) = \begin{array}{cccccccccc} 3 & 6 & 8 & 9 & 5 & 1 & 2 & 4 & 7 \end{array}$$

$$R(\underline{s}) = M_9(\underline{s}) = \begin{array}{cccccccccc} 9 & 8 & 6 & 3 & 5 & 7 & 4 & 2 & 1 \end{array}$$

$$M_3(\underline{s}) = R M_7(\underline{s}) = \begin{array}{cccccccccc} 3 & 6 & 2 & 1 & 5 & 9 & 8 & 4 & 7 \end{array}$$

$$M_7(\underline{s}) = R M_3(\underline{s}) = \begin{array}{cccccccccc} 7 & 4 & 8 & 9 & 5 & 1 & 2 & 6 & 3 \end{array}$$

$$XR(\underline{s}) = X M_9(\underline{s}) = \begin{array}{cccccccccc} 7 & 4 & 2 & 1 & 5 & 9 & 8 & 6 & 3 \end{array}$$

$$X M_3(\underline{s}) = X R M_7(\underline{s}) = \begin{array}{cccccccccc} 9 & 8 & 4 & 7 & 5 & 3 & 6 & 2 & 1 \end{array}$$

$$X M_7(\underline{s}) = X R M_3(\underline{s}) = \begin{array}{cccccccccc} 1 & 2 & 6 & 3 & 5 & 7 & 4 & 8 & 9 \end{array}$$

Analogamente per una serie X-simmetrica ($M_9(\underline{s}) = X(\underline{s})$):

$$\underline{s} = X M_9(\underline{s}) = \begin{array}{cccccccccc} 1 & 2 & 6 & 3 & 5 & 9 & 8 & 4 & 7 \end{array}$$

$$X(\underline{s}) = M_9(\underline{s}) = \begin{array}{cccccccccc} 9 & 8 & 4 & 7 & 5 & 1 & 2 & 6 & 3 \end{array}$$

$$R(\underline{s}) = R X M_9(\underline{s}) = \begin{array}{cccccccccc} 7 & 4 & 8 & 9 & 5 & 3 & 6 & 2 & 1 \end{array}$$

$$M_3(\underline{s}) = X M_7(\underline{s}) = \begin{array}{cccccccccc} 3 & 6 & 8 & 9 & 5 & 7 & 4 & 2 & 1 \end{array}$$

$$M_7(\underline{s}) = X M_3(\underline{s}) = \begin{array}{cccccccccc} 7 & 4 & 2 & 1 & 5 & 3 & 6 & 8 & 9 \end{array}$$

$$RX(\underline{s}) = R M_9(\underline{s}) = \begin{array}{cccccccccc} 3 & 6 & 2 & 1 & 5 & 7 & 4 & 8 & 9 \end{array}$$

$$RM_3(\underline{s}) = RX M_7(\underline{s}) = \begin{array}{cccccccccc} 1 & 2 & 4 & 7 & 5 & 9 & 8 & 6 & 3 \end{array}$$

$$RM_7(\underline{s}) = RX M_3(\underline{s}) = \begin{array}{cccccccccc} 9 & 8 & 6 & 3 & 5 & 1 & 2 & 4 & 7 \end{array}$$

Le classi R-simmetriche e X-simmetriche non sono però le uniche a contenere solo $2 \varphi(n)$ elementi. Anche escludendo il caso $R(\underline{s}) = M_k(\underline{s})$, o $X(\underline{s}) = M_k(\underline{s})$, con $(k, n) = 1$ e $k \neq n-1$, esiste ancora la possibilità che sia $R X(\underline{s}) = M_k(\underline{s})$, per qualche \underline{s} .

Esempio: $n=12$ (per cui vale $M_5 M_7 = M_{11}$, $M_5 M_{11} = M_7$, $M_7 M_{11} = M_5$):

$$\underline{s} = R X M_5(\underline{s}) = 2 \ 11 \ 3 \ 7 \ 10 \ 6 \ 4 \ 1 \ 9 \ 5 \ 8$$

$$R(\underline{s}) = X M_5(\underline{s}) = 8 \ 5 \ 9 \ 1 \ 4 \ 6 \ 10 \ 7 \ 3 \ 11 \ 2$$

$$X(\underline{s}) = R M_5(\underline{s}) = 4 \ 1 \ 9 \ 5 \ 8 \ 6 \ 2 \ 11 \ 3 \ 7 \ 10$$

$$M_5(\underline{s}) = R X(\underline{s}) = 10 \ 7 \ 3 \ 11 \ 2 \ 6 \ 8 \ 5 \ 9 \ 1 \ 4$$

$$M_7(\underline{s}) = RX M_{11}(\underline{s}) = 2 \ 5 \ 9 \ 1 \ 10 \ 6 \ 4 \ 7 \ 3 \ 11 \ 8$$

$$M_{11}(\underline{s}) = RX M_7(\underline{s}) = 10 \ 1 \ 9 \ 5 \ 2 \ 6 \ 8 \ 11 \ 3 \ 7 \ 4$$

$$X M_7(\underline{s}) = R M_{11}(\underline{s}) = 4 \ 7 \ 3 \ 11 \ 8 \ 6 \ 2 \ 5 \ 9 \ 1 \ 10$$

$$X M_{11}(\underline{s}) = R M_7(\underline{s}) = 8 \ 11 \ 3 \ 7 \ 4 \ 6 \ 10 \ 1 \ 9 \ 5 \ 2$$

Nell'esempio precedente la serie era centrata, ma le serie con questa proprietà non necessariamente appartengono al sottoinsieme $\{\underline{c}\}$. Si osservi ad esempio la seguente:

$$\underline{s} = R X M_5(\underline{s}) = 1 \ 10 \ 8 \ 7 \ 3 \ 11 \ 4 \ 2 \ 5 \ 6 \ 9$$

$$M_5(\underline{s}) = R X(\underline{s}) = 5 \ 2 \ 4 \ 11 \ 3 \ 7 \ 8 \ 10 \ 1 \ 6 \ 9$$

Chiameremo serie RX-simmetriche le serie per cui valga $RX(\underline{s}) = M_k(\underline{s})$, per qualche operatore M_k .

Numero delle serie di scarti equilibrate

n	s	c	c_p	c_x
2	1	1	1	1
4	2	2	2	2
6	4	4	4	0
8	24	8	8	0
10	288	64	48	16
12	3,856	576	176	120
14	89,328		576	0

BIBLIOGRAFIA

- 1) Haus, G.: "Descrizione formale di processi musicali"
1976 - Tesi di Laurea - Università di Milano.
- 2) Milani, M.: "N-Tone System: Symmetrical Series".
1976 - CNUCE-CNR - Pisa.
- 3) Niven, I.-Zuckerman, H.S. : "An Introduction to the
Theory of numbers" - 1960 - New York.
- 4) Riotte, A.: "Generation des Series Equilibrees"
1963 relazione interna Euratom - Ispra.

GRUPPO DOC (RAI - MILANO)
"Computer music e produzione di
programmi radioTeleritivi"

Computer-music e produzione di programmi radiotelevisivi

L'approccio al computer con l'intenzione di far musica non è solo una stimolante novità, peraltro ricercata nei secoli, ma soprattutto un nuovo modo di produzione che si affianca e non sostituisce il modello tradizionale.

La nostra presenza qui non può né vuole essere un contributo teorico ai vari gruppi che studiano questo problema nel campo dell'informatica. Noi ci presentiamo con un prodotto finito, che andrà in onda sui canali ufficiali; ci presentiamo come degli utenti e dei promotori dei vostri studi, ma non crediamo nemmeno più alla figura del compositore nei termini in cui la tradizione lo vuole.

Da una parte sentiamo l'esigenza di conoscere il mezzo per poterne trarre e rendere operative tutte le possibilità che una manualità limitata non fa venire a galla, e dall'altra ci siamo resi conto che la poetica stessa non può non subire la variazione determinata dalla stessa logica di strutturazione della macchina. La macchina non è un essere pensante, ma è un essere che manifesta, per la sua stessa logica di classificazione, il pensiero dei tanti che hanno collaborato alla sua costruzione e programmazione; questa logica della macchina muta la logica della composizione.

Il dato è fin quasi banale nella sua semplicità, ma assume una sua validità se si pensa all'esigenza che ogni volta si presenta di ingabbiare il processo creativo in regole che siano diverse da quelle classiche della armonia o comunque di un sistema organizzato; è curiosa e affascinante la disponibilità a costruire su leggi prese a prestito da un'altra disciplina, sia essa la matematica o la pittura. Il computer permette questa traslazione del senso logico, anche se non supportato da alcun senso significante. In questa ottica noi ci siamo prefissi alcuni obiettivi estremamente limitati e semplici: da una parte funzionalizzare un rapporto già aperto con un ente operante nel campo dell'informatica, e dall'altra aprire dei rapporti, anche se frammentari, con chiunque sia interessato a questa colla-

borazione. Ogni conoscenza, ogni nuovo approccio deve comunque sortire un nuovo programma, vuoi radiofonico, vuoi televisivo.

Noi supponiamo, anche se l'ipotesi non è per ora suffragata da nessuna considerazione logica, che sia possibile arricchire la possibilità produttiva mixando le esperienze di tutti.

Nell'ambito dello spirito della riforma radiotelevisiva sentiamo come dovere la collaborazione con forze esterne qualificate in questa ottica. Lo spazio esiste: infatti sia con la seconda rete televisiva, sia con la terza rete radiofonica, sia con la direzione degli sperimentali, sia con il Premio Italia, si sono avviate concrete possibilità di collaborazione, non più in maniera saltuaria e frammentaria. Nel contempo è in progetto una istituzionalizzazione dei rapporti con il CNUCE ed un ampliamento delle possibilità operative del nostro studio di fonologia musicale, che sarà in grado di produrre anche computer-music.

Tutto ciò non è che l'avvio di un processo che non può non svilupparsi, e che è partito grazie al supporto di programmi come questo.

Soltanto in questa luce noi siamo oggi qui presenti, fidando in una collaborazione che possa dare a noi nuovi strumenti di produzione, ed a voi una tribuna dalla quale far conoscere, in tutto o in parte, i risultati delle vostre ricerche.

Gruppo DOC

Centro di Produzione RAI - Milano

Giuliano TISATO

"Un sistema interattivo di software per
le ricerche del ruono in Tempo reale"

AN INTERACTIVE SOFTWARE SYSTEM FOR REAL-TIME SOUND SYNTHESIS

Graziano G.Tisato

Centro di Calcolo - Università di Padova

Abstract

A real-time all software system for computer generated sound synthesis is described.

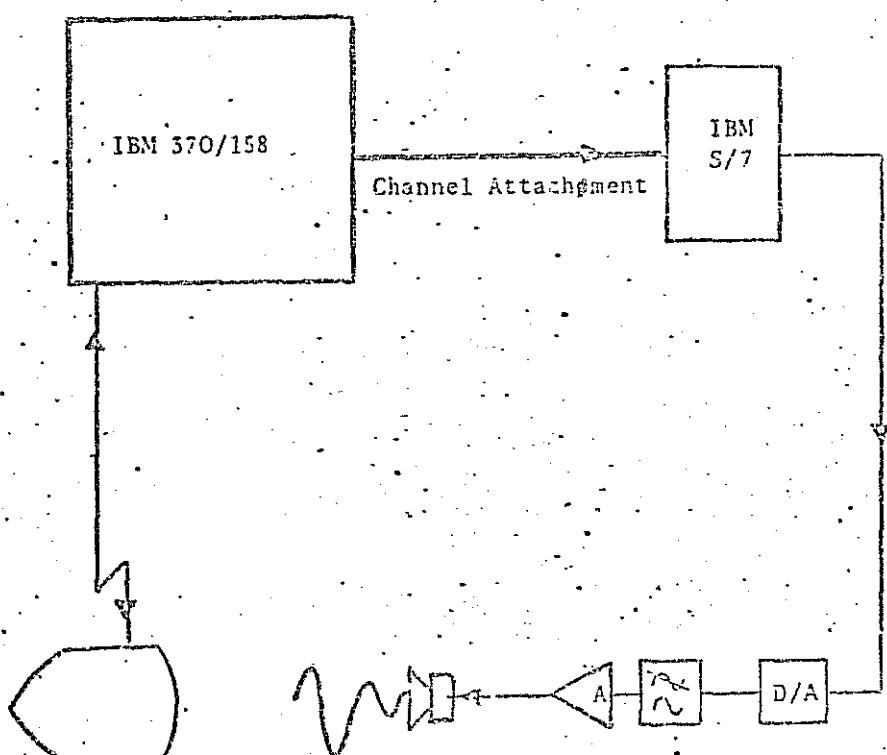
The system, designed for the IBM 360/370 interfaced with a D/A converter, works in a multiprogramming environment and permits the real-time input and processing of the various sound parameters by means of a keyboard or video terminal.

The purpose of the system is to provide for both the researcher in acoustical theory and the composer of computer-realized electronic music a working tool appreciably more simple and direct than traditional computer-synthesis techniques.

Most important especially for timbre research, the system provides an instant feedback in response to any parameter change, with a substantial reduction in both actual work-time and CPU-time.

Introduction: The researcher who is concerned with digital sound synthesis, encounters some difficulties due both to the considerable complexity of the languages so far available for this purpose, and to the excessive time-delay, caused by batch processing, between conception and the actual audio results. These inconveniences cause an uneconomical use of the computer and discourage people potentially interested in such a promising field for research and creative activity.

The 'ICMS' project (Interactive Computer Music System) proposes to facilitate the computer approach to researchers and composers as well as to provide didactic help to music students. (A version of this program is already being utilized by the students of Accademies of music "C.Pollini" of Padua and "B.Marcello" of Venice).



Video Terminal 3270
or
Communications Terminal 2741

Fig.1 Overall system.

The system:

Figure 1 shows the general characteristics of this system as it has been implemented at the 'Centro di Calcolo' of the University of Padua.

The input consists of a 2741 Communications Terminal or a 3270 Video Terminal remote, connected to an IBM System 370/158. This is a 2 Mbytes real storage computer and works under VS2 operating system.

The 'ICMS' system does not require that the S/370 be dedicated, thus permitting the computer to be run simultaneously for other applications both batch and teleprocessing (for example:CICS, TSO, etc.).

The S/370 is interfaced with a D/A through an IBM S/7 mini-computer, which works only as a transit buffer.

The S/370 and S/7 connection is carried out through Channel Attachment and performs a high speed memory to memory data transfer (abt. 250 Kbytes/sec).

The software:

The use of a large size computer grants considerable software advantages both with respect to programming and to memory use.

The program requires a 250 Kbytes region and it is conceived according to modular principles to enable an easy replacement and enlargement of its functions.

S/370

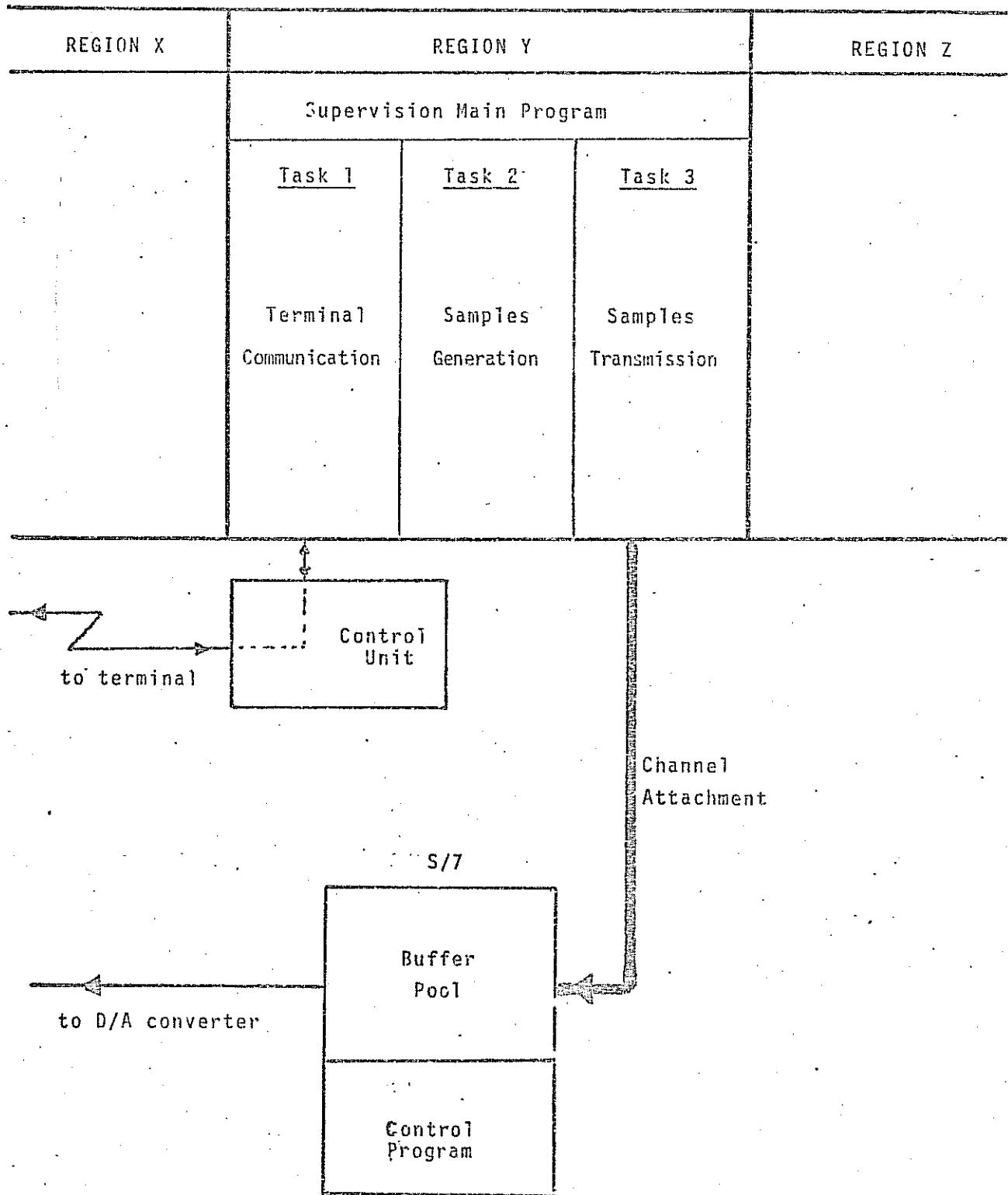


Fig. 2 Functional blocks

Within its region the program works in multitasking, superimposing the samples calculation on the I/O operations.

Figure 2 illustrates the functions carried out by the various tasks.

The supervision functions and the synchronization operations are executed by a main PL/I program.

Task 1 actuates the terminal-S/370 connection. Two modules have been written in Assembler according to the BTAM access method, one for supporting the 2741 and one for the 3270.

The function of task 2 is to transmit to the S/7 the samples in blocks of atb. 15.000 at a time.

The channel operations have been expressly defined using EXCP (Execute Channel Program)macro.

Task 3, written in assembler, calculates the samples with a cycle of approximately 40 usec/ sample, and operates mainly by register operations.

The current synthesis algorhithm (which can be replaced by any other one based on a different synthesis technique provided that it is quick enough) makes use of frequency modulation in order to obtain a time-variant sound spectrum according to Chowning's proposals.

The function table for the sinusoidal oscillator consists of 8.192 elements in order to keep the truncation error small enough, according to criteria suggested by D.Freedman.

The sizes of the other tables are, however, variable depending on the specific contextual requirements.

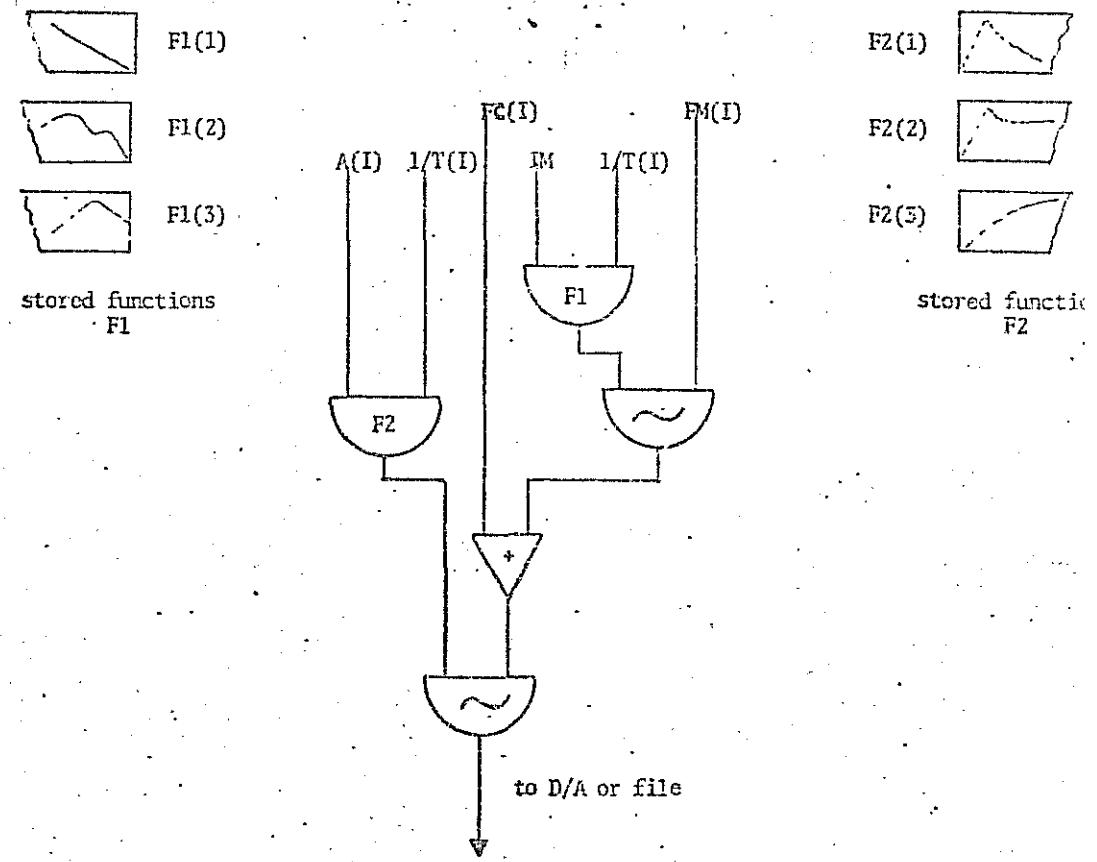


Fig.3 Frequency Modulation circuit, and parameters over which we can obtain initial or interrupt control:
amplitude sound envelope A and F_2 ,
sound duration T ,
carrier frequency FC ,
index modulation IM and F_1 ,
modulating frequency FM .
So we can define an appropriate association between interrupt key and single parameter or set of parameters and obtain dynamic sound variations at synthesis time.

The synthesis of the samples proceeds according to parameters previously fixed or supplied in real-time by the terminal: figure 3 shows the FM circuit and variable parameters.

After the identification, the interrupt is handled by changing the pointers on the tables or replacing the tables themselves with others. This occurs within a few μ sec.

Finally the assembler module running on S/7, simulates a peripheral device connected directly by a Selector Channel to S/370 and acts as an interface towards the D/A converter. The Selector Channel use is important since I/O operations are fully executed before control is given to other operations of the same type. The 16 kwords of the S/7 memory are used nearly exclusively to form two buffers of at least 5000 samples each, which are filled by the channel and emptied alternatively into the D/A converter. Thus there is no time limitation whatsoever for any particular run.

The Language:

The language is the most interesting aspect of the system and permits a wide variety of developments, since the use of a large computer enables links with other applications as well as enlargement of the options.

The principles we have followed are:

- ease of data input
- control of all the sound parameters during both initialization and synthesis
- possibility of changing the parameters separately or in block.

The user has thus both a considerable flexibility at his disposal and a ease of operation which permit him to devote most of his efforts to the creative and musical aspect of his work.

The interventions on an interrupt basis allow the user to freely experiment with dynamic variations or combinations of the parameters whose results cannot always be foreseen.

The possible options are of 3 types:

initialization, operative and control options.

Through initialization it is possible to construct the tables of the variables such as the modulation index, the amplitude envelope, and tempo, and to establish the correspondence of these tables with the interrupt keys.

The functions are defined as points to be interpolated or as continuous functions, and it is possible to add these functions together. During initialization the sampling rate is also established the upper limit of which depends, of course, on the current machine load. For synthesis and variation of parameters it is possible to obtain up to 20 KHz in real time.

SPECTRUM FOR INDEX MODULATION 1:15.50 AND RATIO OF THE CARRIER TO THE MODULATING FREQUENCY Ratio: 43

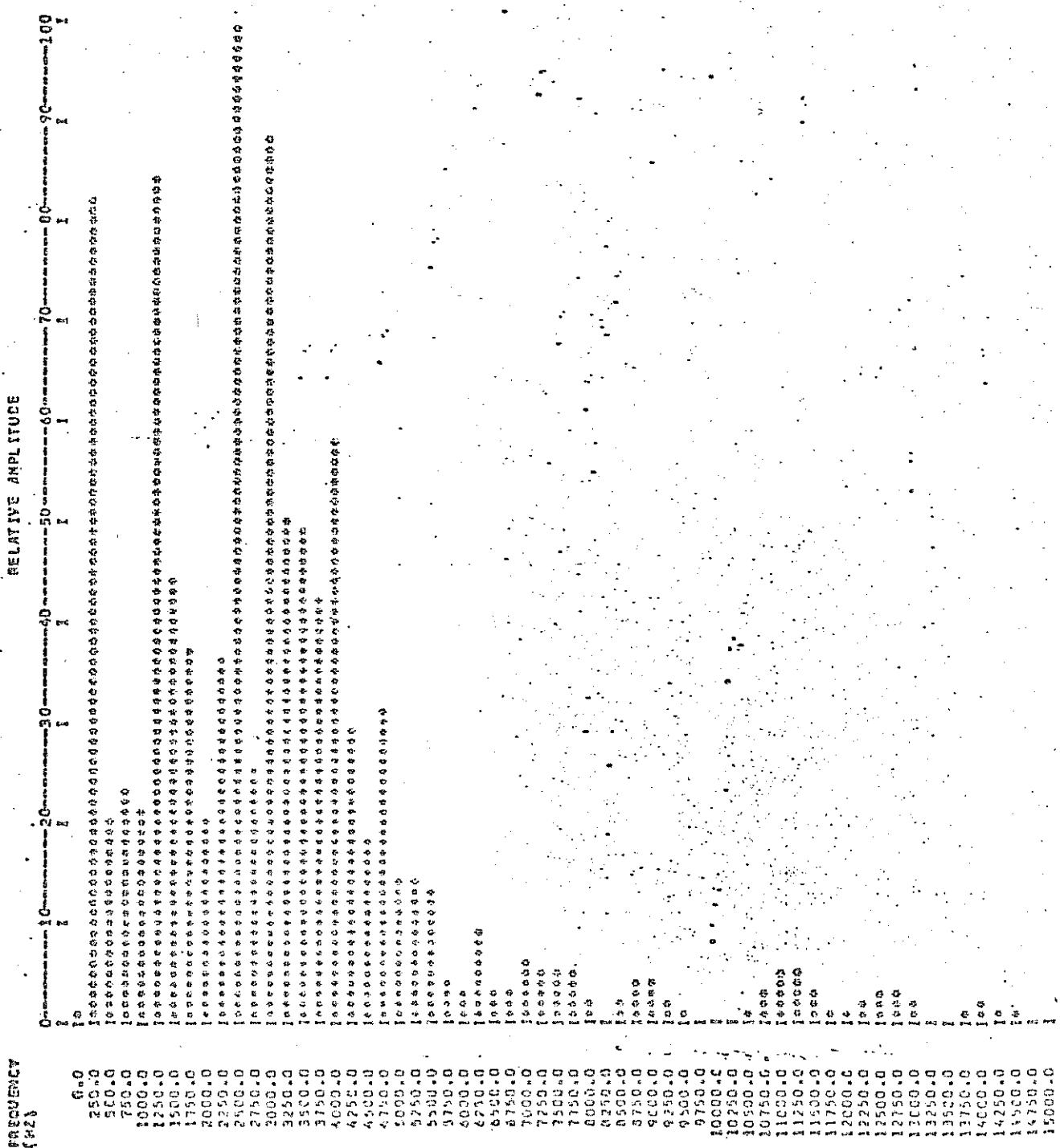


Fig. 4 ICMS video or printer output

Then a higher sampling rate or stereophonic output is required, it is possible to synthesize the sound and store it on Disk for immediate listening.

In fact the limit is not so much imposed by the transmission channel, which permits a notably high frequency, as by the calculation time per sample.

The fundamental frequencies (variable A of figure 3) and durations T are obtained very simply by associating a frequency and a pre-established duration to every key of the keyboard.

The correspondence between frequencies and keys is, by default, determined by the pitches of the tempered scale (86 pitches are available, 43 for lower-case characters and 43 for the upper-case).

This correspondence can be modified in any moment by any other set of frequency relationships.

The fundamental frequencies and durations are stored in vectors and eventually saved on disk from which they may be subsequently recalled.

To realize traditional scores as if in a performance-interpretation situation, the output of the musical code of the 'MUSICA' program may be fed in to the system.

The operative options enable the user to hear the sound and to vary the parameters on interrupt basis according to the pre-established table-correspondences.

Furthermore, they permit storing the digital signal on disk and performing on it simple operations such as mixing additional sounds, rescaling any or all parameters and setting-up for a stereophonic output.

The control options allow

- to store into records the necessary parameters, so that it is possible to reproduce the desired sound at any time.

	SPECTRUM (KHZ)	TIME (20 ms)
0	3	1
1	6	2
2	9	3
3	12	4
4	15	5
5		6
6		7
7		8
8		9
9		10
10		11
11		12
12		13
13		14
14		15
15		16
16		17
17		18
18		19
19		20
20		21
21		22
22		23
23		24
24		25
25		26
26		27
27		28
28		29
29		30
30		31
31		32
32		33
33		34
34		35
35		36
36		37
37		38
38		39
39		40
40		41
41		42
42		43
43		44
44		45
45		46
46		47
47		48
48		49
49		50
50		51
51		52
52		53
53		54
54		55
55		56
56		57
57		58
58		59
59		60

Fig.5 ICMS printout with spectral components evolution
for index modulation ranging from 0 to 10 and
ratio of the modulating frequency to the carrier $r=1.43$

- to display on video or on printer spectral components of signals obtained by FFT, see fig. 4.
- to print total harmonic spectral evolution during a fixed time interval for a fixed signal, see fig. 5
- to call for a complete printout of the command flow and parameter variation so that it is possible to have a description of any situation which might have occurred.

Possible error conditions are shown and handled so that the user may provide the appropriate corrections.

Conclusions:

The results so far achieved are encouraging and suggest new applications both of implementation of other synthesis techniques and of extended man-machine interactions.

REFERENCES

Max V. Mathews, The technology of computer music.

The MIT press, Boston, 1969

Jhon M. Chowning, The synthesis of complex audio spectra
by means of frequency modulation.

JASA, Vol.21, N.7, September 1973, p.526 - 534

David M. Freedman, A digital computer for the electronic
music studio.

JASA, Vol.15, N.1, January 1967, p.43 - 50

Herbert S. Howe, Electronic music synthesis.

Dent & Sons, London, 1975

Jean-Claude Risset, An introductory catalog of computer
synthesized sounds.

Bell Telephone Laboratories, N.J., 1970

Jon H. Appleton, The development and practice of
electronic music

PRENTICE-HALL, N.J., 1975

G. De Poli-G. Debiasi, Un linguaggio per la trascrizione di
testi musicali per elaboratori elettronici.

ZANIBON, Padova, 1974

FUGAZZA - MENCHI NELLI

"DISPOSITIVO PER LA GENERAZIONE DI SUONI
MEDIANTE SINTESI ADDITIVA ARMONICA
OPERANTE PER MEZZO DI PERIFERICHE
VELOCI CONTROLLATE DA UN MICROCALCOLATORE"

R E L A Z I O N E

Relatore: E.M. - Elettronica Musicale - V. S. Isaia - 49
BOLOGNA -

Nelle persone di:

Prof. G.F. Fugazza
Ing. S. Menchinelli

Argomento:

descrizione ed indirizzi di utilizzazione di:

DISPOSITIVO PER LA GENERAZIONE DI SUONI MEDIANTE
SINTESI ADDITIVA ARMONICA COSTITUITO DA PERIFERICHE
VELOCI CONTROLLATE DA UN MICROCALCOLATORE.

PREMESSA.

Desiderio sempre vivo e non ancora totalmente appagato di ogni musicista resta ancora quello di poter esercitare un diretto controllo sui parametri che caratterizzano l'evento sonoro. Tali parametri, come la teoria e la pratica ci hanno ormai incontestabilmente dimostrato, sono frequenza, intensità e timbro, comprendendo in questa voce anche i transitori. Mentre altezza ed intensità risultano ormai agevolmente controllabili, lo stesso non accade per il timbro che si identifica nel rapporto di ampiezza tra le armoniche presenti e, in condizioni nonstatiche, anche con il rapporto di fase tra esse.

Gli apparecchi a disposizione dei musicisti nella generalità dei casi consentono la variazione di spettri armonici mediante sintesi sottrattiva, (filtri) ciò che impedisce all'operatore il controllo diretto su ciascuna armonica e, conseguentemente, gli impedisce di rilevare sperimentalmente lo apporto di ciascuna armonica alla costituzione di un 'colore' musicale. E' tuttavia possibile realizzare tali controlli, facendo però ricorso a sistemi molto complessi (computers con terminali speciali) che impongono alta specializzazione, elevati costi di esercizio, collocandosi fuori dall'area di accessibilità di gran parte dei musicisti.

La disponibilità di microprocessori sufficientemente versatili a prezzi sempre più contenuti ci hanno indotto ad esaminare ogni loro possibile utilizzazione nel settore specifico dell'elettronica musicale; e da questo esame abbiamo tratta la convinzione che da un loro corretto impiego potrebbe venire una concreta prospettiva per il superamento delle difficoltà prima dette. Forti di una tale convinzione abbiamo elaborato il progetto per una prima apparecchiatura sperimentale dalle molteplici prospettive di utilizzo; prima fra tutte quella di consentire al musicista: una quantizzazione dei parametri caratterizzanti l'evento sonoro in termini numerici e quindi ripetibili senza errori significanti, un intervento continuo su di essi anche in tempo reale; un rapporto uomo/

macchina che non imponga la mediazione di specialisti e, ultimo solo in ordine di esposizione, il vantaggio di un contenuto costo delle apparecchiature.

I microprocessori di cui s'è detto, pur essendo sufficientemente versatili, possono risultare piuttosto lenti, nella nostra proposta la difficoltà è stata superata adottando una periferica veloce che verrà descritta successivamente.

E' noto che per quanto riguarda la generazione di forme di onda in tempo reale anche i grandi calcolatori dell'attuale generazione si trovano in seria difficoltà in quanto tale funzione impegnava la quasi totalità della loro capacità operativa, lasciando troppo poco spazio alle funzioni di strutturazione e controllo della composizione, funzioni che sono, per altro, più congeniali alla architettura del calcolatore. Da ciò derivano del resto le limitazioni incontrate nell'utilizzo di grandi calcolatori per "generare" musica in tempo reale! Anche in questo settore noi riteniamo che la strutturazione di un sistema utilizzante le apparecchiature da noi proposte possa rappresentare UN BUON PASSO AVANTI.

STRUTTURA DEL SISTEMA.

Si presenta in due configurazioni:

'Sistema minimo' - costituito da un microcalcolatore e da una periferica veloce.

'Sistema espanso' - costituito da un certo numero di sistemi minimi - anche relativamente elevato - gestiti da un calcolatore centrale di capacità adeguata.

Le prestazioni del 'Sistema minimo' sono illustrate più oltre. Relativamente al 'sistema espanso' possiamo dire subito che il calcolatore centrale, sgravato dei compiti connessi alla generazione degli eventi sonori e dei loro inviluppi perchè assolti dai singoli sistemi minimi, è in grado di dedicare l'intera sua capacità di calcolo alla elaborazione di macrostrutture inherenti l'architettura formale della composizione.

LA PERIFERICA VELOCE.

E' realizzata con circuiti logici MSI in Schottky TTL per ottimizzare il rapporto velocità/costo. La periferica è in grado di generare forme d'onda mediante 128 campionamenti/periodo, ogni campione è la somma algebrica di 16 valori determinati dalla ampiezza e dalla fase delle prime 16 armoniche della serie di Fourier, per una frequenza fondamentale fino a 2.000 Hz. La frequenza fondamentale è determinata da un clock controllabile digitalmente. Il segnale audio in uscita è fornito da un convertitore DAC a 12 bit seguito da un tracking filter destinato a sopprimere il rumore di dismetizzazione.

I valori della funzione 'seno' sono permanentemente memorizzati (solo il primo quadrante) in una RAM: La tabella contenente tali valori è ripetuta 32 volte, per i 32 valori di ampiezza previsti per ciascuna delle 16 armoniche. La ripetizione della tabella è stata scelta, in alternativa all'adozione di moltiplicatori digitali, per i seguenti motivi: I° - maggiore velocità di calcolo, II° - minor costo dell'occorrente spazio di ROM rispetto al prezzo dei moltiplicatori attualmente disponibili.

Come si è visto l'ampiezza di ogni singola armonica viene discretizzata mediante 32 livelli (parole di 5 bit binari) corrispondenti ad I_{db} per livello. L'ultimo però dei 32 livelli si fa però corrispondere allo zero di ampiezza. (armonica totalmente assente) La fase delle armoniche è discretizzata in 128 valori (parola di 7 bit binari) che determinano uno sfasamento elementare di 2.81 gradi.

L'insieme delle informazioni di ampiezza e di fase costituisce quindi una parola di 12 bit che caratterizza ogni singola armonica; una tabella di 16 parole da 12 bit caratterizza conseguentemente una struttura armonica.

Parte integrante della periferica è una copia di RAM veloci da 16x12 bit. Durante il funzionamento la periferica preleva le informazioni da una delle due RAM mentre l'altra resta disponibile per memorizzare una nuova tabella di dati forniti dal microcalcolatore. Al termine di ogni ciclo (periodo della forma d'onda) la periferica provvede ad invertire il ruolo delle due RAM se una nuova tabella è stata nel frattempo caricata sulla RAM rimasta a disposizione del microprocessore. Questa operazione fornisce anche un segnale di 'ready' che viene avviato al microcalcolatore che si asterrà dal caricare nuove tabelle sulla RAM fino alla comparsa del segnale di 'ready'; ciò ad evitare che il microcalcolatore carichi nuove tabelle sovrapposte alle precedenti non ancora utilizzate dalla periferica.

Il dispositivo come descritto consente, nonostante la relativa lentezza dei microcalcolatori attualmente disponibili, di modificare la forma d'onda ciclo per ciclo fino ad una frequenza fondamentale di 1000 Hz. Macchine con grado di parallelismo superiore agli 8 BIT (tipici dei microcalcolatori) consentirebbero di rinnovare le tabelle a velocità superiori ad 1 KHz, ma questo non sembra essere determinante per l'attuazione di transitori timbrici.

MICROCALCOLATORI

Il microcalcolatore originariamente scelto per questa applicazione è l'8048 della INTEL, dispositivo avente un rapporto prestazioni/prezzo molto interessante, un ampio repertorio d'istruzioni ed una sufficiente capacità di calcolo. Tuttavia l'estrema versatilità della periferica è tale da consentire l'impiego di microcalcolatori anche più lenti e con repertorio di istruzioni meno esteso.

Nel sistema minimo il più semplice dispositivo di 'ingresso dati' potrà essere costituito da una tastiera numerica che permette di programmare manualmente ampiezza e fase di ogni armonica. La forma d'onda risultante potrà essere eseguita alle varie frequenze per mezzo di una tastiera standard che controlla il clock per mezzo di un dispositivo digitale, realizzando la scala temperata o qualsiasi altra serie intervallare. Il microcalcolatore potrà essere corredato di opportune sub-routines che manipolano la struttura armonica in base alla frequenza della fondamentale per adeguare il timbro alle regioni formantiche via via interessate dalla tessitura dell'esecuzione. (Cio permette di realizzare timbri varianti a seconda del 'registro' come avviene per gli strumenti tradizionali.)

Altre sub-routines, costituenti il 'soft' di base del microcalcolatore, riguarderanno l'esecuzione automatica di inviluppi timbri-

ci in cui ogni armonica avrà un suo profilo temporale preprogrammato.

Tutte le sub-routines, sia quelle riguardanti la sintesi formantica sia quelle riguardanti l'inviluppo timbrico, potranno essere modificate nei loro valori frequenziali (formanti) e temporali (ADSR) mediante interventi in tempo reale da effettuarsi su comandi appositamente predisposti. (interfaccia uomo/macchina)

Altre sub-routines ancora potranno riguardare la modulazione di fase per ottenere vibrati o 'trasnitori' realizzati mediante inizio dell'evento con tutte le armoniche 'in fase' e successivo sfasamento casuale.

Una speciale sub-routine potrebbe anche contenere un programma per la modulazione in fase di una o più armoniche al fine di simulare eventi sonori costituiti anche da frequenze tra loro in rapporto non armonico.

va qui sottolineato che qualsiasi modulazione può essere ottenuta imprimendo ad ogni parametro un profilo modulante proprio, sfuggendo dall'andamento monotonico imposto dalla sintesi sottrattiva. E l'area di sperimentazione che così si pone a disposizione del musicista è di una vastità e di un interesse difficilmente quantizzabili.

Nel 'sistema espanso' ciascun dei microcalcolatori riceverà dal calcolatore centrale le istruzioni per la gestione delle sub-routines constituenti il suo repertorio ed il calcolatore centrale risulterà sgravato dei compiti di 'tempo reale' di primo e secondo ordine. (periodo della frequenza in gioco=Ims. - periodo dell'inviluppo=Isec.) Il risultato sarà la possibile esecuzione di brani polifonici e politimbrici anche di notevole densità.

Rispetto ad altri sistemi fino ad ora realizzati riteniamo che il 'sistema espanso' da noi proposto presenta alcuni importanti vantaggi:

I - potendo asservire al calcolatore centrale un numero relativamente elevato di 'sistemi minimi' si otterrà un 'sistema espanso' in grado di controllare separatamente i parametri di un elevato numero di eventi sonori anche contemporanei. (polifonico e politimbrico) II - esecuzione di brani molto complessi, con una elevata velocità di mutamento simultaneo dei parametri caratterizzanti ciascun evento. III - La assoluta indipendenza di ciascun 'sistema minimo' quale generatore di suoni consente di istruire i segnali provenienti da ciascuno verso 'punti suono' altrettanto indipendenti (distribuzione del suono nello spazio) o di mixare tali segnali nella più ampia libertà. IV - Il rapporto prestazioni/prezzo rimane ottimale ad ogni grado di espansione.

UTILIZZAZIONI.

STRUMENTO MUSICALE.

Provisto di tastiera standard e dotato di un certo numero di sub-routines il 'sistema minimo' si presenta come un normale strumento musicale per esecuzioni dal vivo.

Esso potrà generare eventi interamente presettati, oppure alcuni parametri potranno essere lasciati all'intervento dello esecutore, od ancora: su programmi prefissati potranno essere lasciate alcune aree di intervento da parte dell'esecutore.

Qualsiasi tipo di trasduttore ed ogni forma di comandi potrà essere adottato per facilitare l'intervento dell'esecutore sulla macchina. La scelta è legata al numero ed alla contemporaneità dei parametri da controllare ed alle capacità manuali dello esecutore.

SUUSIDIO DIDATTICO.

Provisto di opportuno display l'apparecchio nella versione 'minima' può servire per la generazione e l'ascolto di spettri di cui si conosce la struttura. L'operazione, correlando immagine, suono e dati matematici, rappresenta un training di eccezionale valore per musicisti ed operatori.

APPARATO DA LABORATORIO.

Le sue applicazioni in questa direzione sono numerosissime.

Simulatore di filtro. Fornendo i dati relativi a: forma d'onda in ingresso, frequenza, funzione di trasferimento del filtro, si otterrà il risultato -acusticamente valutabile-. Sarà così possibile, anche in via sperimentale, selezionare tramite ascolto il timbro voluto prima di scegliere la forma d'onda ed il tipo di filtro opportuni al raggiungimento di un certo 'colore' sonoro.

Formanti. Volendo ottenere un certo risultato timbrico lungo una 'estensione data' mediante zone formantiche fisse, si potrà simulare anche questo procedimento, per valutarne il risultato acustico prima di operare una scelta.

APPARECCHIO AUDIOMETRICO.

Stante la grande versatilità nella generazione di eventi sonori molto diversificati - estante la ripetibilità dei medesimi - l'apparecchio può generare le strutture più adatte ad un esame audiometrico.

Si può ipotizzare la preparazione di profili audiometrici individuali di soggetti sano formulare profili audiometrici caratterizzanti particolari disfunzioni dell'apparato uditivo.

PERIFERICA DI UN SISTEMA ESPANSO.

Tale applicazione è stata sufficientemente trattata nel testo.

CONCLUSIONE.

Il dispositivo da noi proposto è chiaramente in fase sperimentale. Alcune scelte di base, quali il numero delle armoniche, la loro ampiezza e discretizzazione della medesima, la discretizzazione della fase, l'estensione in frequenza, sono state operate sulla base di due considerazioni determinanti: a) disponibilità di componenti con un soddisfacente rapporto prestazioni/prezzo, b) dimensione minima utile a verificare la validità del sistema.

L'apparato è in corso di realizzazione presso una Ditta italiana, la CRB Elettronica S.p.A. e sarà disponibile per un test completo prevedibilmente entro la metà del prossimo anno.

E' nei nostri programmi realizzare sperimentazioni estese il più possibile ed in funzione di questo invitiamo gli interessati a segnalarci la loro disponibilità per una collaborazione i cui termini potranno essere concordati in seguito.

Da queste sperimentazioni intendiamo ricavare indicazioni sufficienti alla definizione del progetto definitivo.

SERGIO CAVALLIERE - GIUSEPPE DI GIVENDO
VITO PEDULLO - INMACOLATA ORTOSECCO
GRUPPO DI ACUSTICA ELETTRONICA
ISTITUTO DI FISICA - NAPOLI

"UN CALCOLATORE VELOCE ORIENTATO
PER LA SINTESI DI SEGNALI ACUSTICI
IN TEMPO REALE"

Sergio Cavalieri - Giuseppe Di Giugno - Vito Fedullo - Francesco Orlando
Gruppo di Acustica Elettronica
Istituto di Fisica sperimentale via Tanassi 6 20134
Università di Napoli

UN CALCOLATORE VELOCE ORIENTATO PER LA SINTESI DI SEGNALI
ACUSTICI IN TEMPO REALE

S O M M A R I O

Abbiamo costruito nel laboratorio di acustica elettronica un calcolatore veloce orientato capace di realizzare 64 oscillatori e 64 modulatori di ampiezza in tempo reale e con la possibilità di controllare, sempre in tempo reale, fino a 16 parametri del segnale acustico così sintetizzato.

Questo sistema è usato per applicazioni musicali, in particolare per la sintesi additiva di segnali acustici.

Il sistema attualmente è controllato da un PDP15 della Digital ma l'interfaccia è facilmente adattabile ad un più piccolo minicomputer oppure ad un microprocessore. La stabilità in frequenza dipendente da un cristallo di quarzo è di 10^{-6} . La frequenza di ciascun oscillatore può essere variata da zero a 32 KHz con passi di 4×10^{-3} Hz e l'ampiezza da zero al valore massimo di 2048 passi. Gli oscillatori possono realizzare forme d'onda arbitrarie.

INTRODUZIONE.-

'approccio usato fino ad ora per la sintesi di segnali acustici complessi è stato quello di utilizzare oscillatori analogici controllati da computer tramite convertitori digitali-analogici. Questo metodo è affetto dalle note limitazioni dei sistemi analogici: variazioni nelle caratteristiche del segnale causate da variazioni di temperatura, invecchiamento dei componenti, variazioni della tensione di alimentazione, accuratezza della conversione digitale-analogica ecc. Queste limitazioni diventano intollerabili se si desiderano ad esempio battimenti lenti per sintesi di segnali acustici che differiscono di frazioni di Hz. È noto inoltre che i sistemi inte-

ramente digitali non sono soggetti alle precedenti limitazioni e che le uniche limitazioni consistono negli errori di quantizzazione che sono intrinseci a questi sistemi.

Sono stati fatti alcuni tentativi in questa direzione nella Stanford University, presso la Bell Telephon ed altrove ma le soluzioni adottate sono soggette a grosse limitazioni.

Alcuni di questi sistemi generano solo onde quadre, altri varie forme d'onda che comunque possono essere modificati solo con cambiamenti dell'hardware della macchina. In ogni caso nessuno di questi sistemi può sintetizzare simultaneamente ad un tempo reale più di un solo oscillatore.

DESCRIZIONE DEL SISTEMA.-

Il sistema rappresentato in Fig. 1 è costituito da sette blocchi fondamentali:

- a)-Oscillatore
- b)-Tabella della forma d'onda
- c)-Modulatore
- d)-Logica di smistamento
- e)-Unità di controllo
- f)-Interfaccia con il computer.

L'oscillatore (Fig.2) è una trasposizione in hardware dell'oscillatore simulato dal programma Music V, con una velocità di campionamento di 64 KHz, ed una risoluzione in frequenza di 4×10^{-3} Hz.

L'oscillatore utilizza una memoria per la forma d'onda di 4 K x 12 bit. I campioni della forma d'onda sono scritti in memoria con il PDP15 e quindi si può usare una forma d'onda arbitraria; con queste dimensioni della memoria della forma

d'onda il rapporto segnale-rumore è di 72 dB.

La memoria della forma d'onda può essere utilizzata in tre modi differenti: a) per una forma d'onda di 4 K x 12 bit; b)-per due forme d'onda di 2 K x 12 bit; c)-per quattro forme d'onda di 1 K x 12 bit.

Il caricamento tramite il PDP15 di una forma d'onda di 4 K x 12 bit richiede circa 40 ms.

L'oscillazione viene poi modulata in ampiezza secondo un inviluppo prefissato in software. I 64 oscillatori sono poi smistati su 4 convertitori d/a a 12 bit ; si ottengono così quattro canali audio costituiti ciascuno da 16 oscillatori.

L'attuale interfacciamen^tto con il PDP15 permette una velocità di trasferimento di circa 10^5 parole al secondo. Per un solo oscillatore la frequenza e l'ampiezza possono essere aggiornati ogni $10\mu s$. Questo significa che se vogliamo modificare 64 ampiezze contemporaneamente, ciascuna ampiezza può essere modificata in non meno di $640\mu s$.

Questo implica una limitazione nella frequenza massima di modulazione ed un certo rumore nel caso di variazioni rapide dell'ampiezza. Per superare queste limitazioni stiamo realizzando un generatore di inviluppi in hardware con una frequenza di campionamento di 16 KHz per ciascun oscillatore.

Il sistema funziona con un clock di 2^{22} Hz(4.194304 MHz) ed è realizzato con 112 circuiti integrati TTL. Il sistema è interamente organizzato in pipeline e consente in ogni ciclo di 238.14 ns di eseguire in parallelo tutte le operazioni necessarie per ottenere un nuovo campione ed,inoltre, di scrivere un dato in una delle memorie dall'esterno rendendo così il sistema costantemente "trasparente" al calcolatore.

CONTROLLO DEL SISTEMA.-

Il controllo del sistema viene fatto con una libreria di programmi Macro ed adatti programmi Fortran.

Inoltre é possibile il controllo in tempo reale di sedi ci qualsiasi parametri del suono per mezzo di una console avente 16 ingressi analogici e 16 switch digitali. L'acqui sizione in tempo reale dei dati analogici é realizzata per mezzo di un Datel del tipo MDAS-16 che consente il trasferimento di dati alla velocità di 50 K byte/sec e la risoluzio ne di 12 bit.

La funzione di questi controlli é programmabile ed in più é possibile memorizzare i parametri summenzionati alla velo cità di circa 100 Hz. Di conseguenza possiamo manipolare ed elaborare strutture musicali complesse sempre digitalmen te ed anche in fasi successive senza aver bisogno di costosi registratori acustici multitracce.

APPLICAZIONI.-

In figura 3 e 4 sono indicate due delle varie interconnes sioni che si possono realizzare con questo sistema.

Con le connessioni di Fig. 3, determinando gli inviluppi del suono degli strumenti musicali tradizionali ed i timbri per mezzo dell'analisi armonica, si possono sintetizzare mol ti strumenti.

Lo schema di fig. 4 realizza la sintesi di segnali modulati in frequenza con la tecnica di Chwning; il blocco 1 é il sintetizzatore dogotale hardware. Con il blocco 2 si realizza il controllo dell'inviluppo della frequenza e della frequenza & profondità da modulazione. Il blocco 3 serve ad arricchire i timbri, alterando in modo random le frequenze dei vari oscil latori intorno al valore stabilito dal blocco 2 ottenendo così effetti corali. I blocchi N 2 e 3 sono realizzati in software.

Lo strumento realizzato con questa connessione consente la produzione ed il controllo in tempo reale di timbri particolarmente complessi.

CONCLUSIONE.-

Il sistema realizzato presenta ottime prestazioni per quanto riguarda molti aspetti: l'alta velocità di funzionamento, l'elevato numero di oscillatori sintetizzati simultaneamente, l'elevata frequenza di sintesi, l'alta risoluzione in frequenza. Particolare importanza ha poi la possibilità di controllare in tempo reale, simultaneamente molti parametri del segnale acustico sintetizzato e di registrare in modo digitale le variazioni realizzate.

Grazie a queste caratteristiche il sistema permette, oltre alla realizzazione delle usuali strutture sonore proprie delle apparecchiature analogiche, risultati nuovi per quanto riguarda la produzione di timbri complessi, la realizzazione di strutture musicali particolarmente complesse formate da molti voci contemporanee che evolvono separatamente nel tempo secondo leggi comunque assegnate, grazie anche alla semplicità di programmazione.

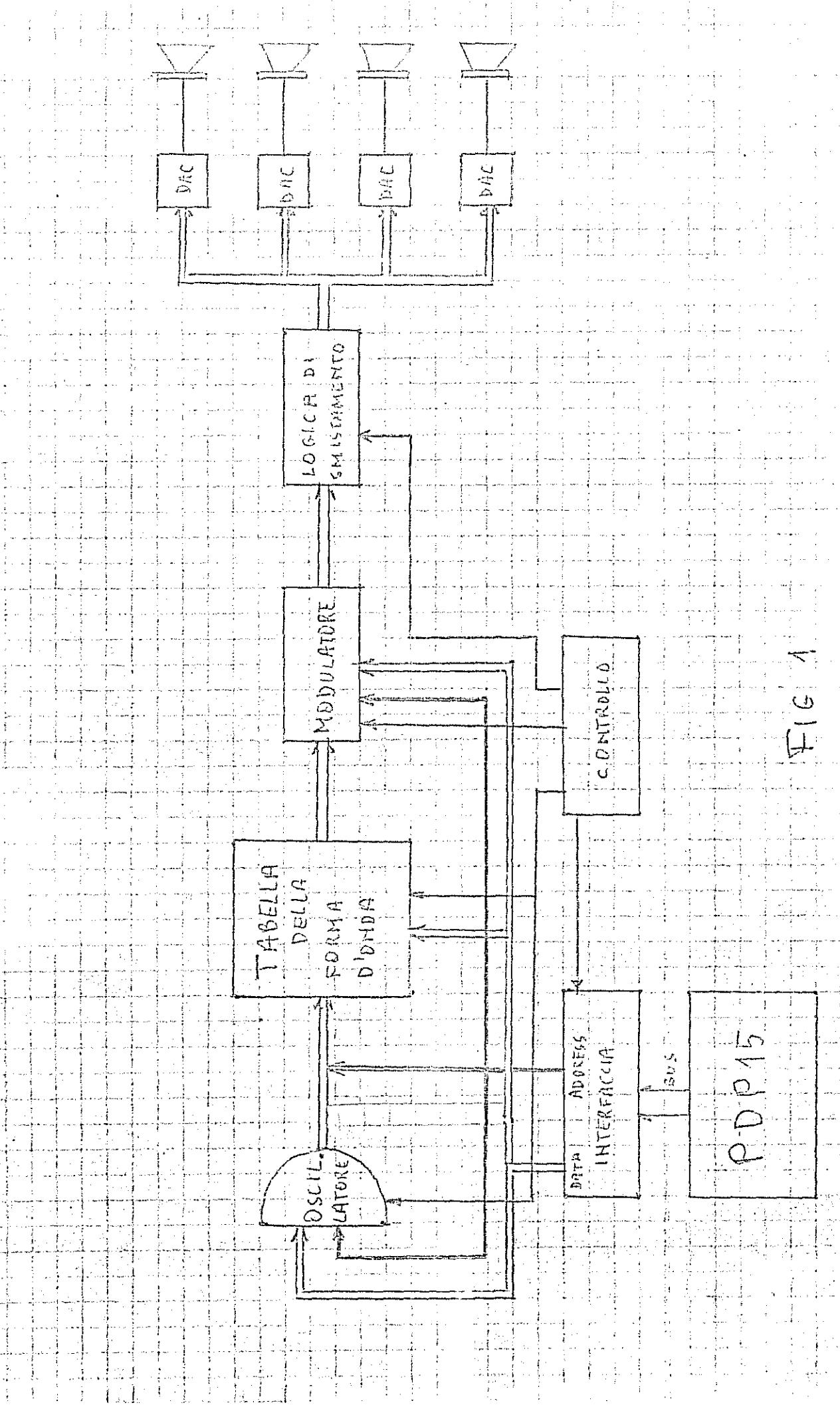


FIG 1

