

**VII COLLOQUIO
DI INFORMATICA MUSICALE**

ATTI

**Roma 23-26 marzo 1988
Pallazzo Taverna, IN/ARCH**

© Copyright 1988 by Associazione Musica Verticale - Roma

Finito di stampare
nel mese di marzo 1988
dalle Edizioni Arti Grafiche Ambrosini
Via Annia, 54 - Roma

Fotocomposizione Studio AR

A cura di Diego Fabrizio Tomassini

Tutti i diritti sono riservati. Nessuna parte di questa pubblicazione può essere riprodotta in qualsiasi forma elettronica o meccanica, comprese fotocopie, senza l'autorizzazione da parte dell'Editore.

VII COLLOQUIO
DI INFORMATICA MUSICALE

A T T I

Roma 23 - 26 Marzo 1988
Palazzo Taverna, IN / ARCH

ENTI PROMOTORI

Ministero del Turismo e dello Spettacolo
Assessorato alla Cultura del Comune di Roma
AIMI / Associazione di Informatica Musicale Italiana
Associazione Musica Verticale
CIDIM / Comitato Nazionale Italiano Musica (CIM / Unesco)
Facoltà di Ingegneria / Dipartimento INFOCOM - Università di Roma
SIM Srl / Società di Informatica Musicale di Roma
Texas Instruments Italia Spa
IN / ARCH Istituto Nazionale di Architettura
Istituto di Acustica "O. M. Corbino" / C.N.R. di Roma

Con il Patrocinio della Regione Lazio

COMITATO PROMOTORE

Alvise Vidolin	AIMI / Associazione di Informatica Musicale Italiana
Paolo Emilio Giua	Istituto di Acustica "O. M. Corbino" del C.N.R.
Francesco Agnello	CIDIM / Comitato Nazionale Italiano Musica (CIM / Unesco)
Nicola Sani	SIM / Società di Informatica Musicale
Giuseppe Di Giugno	IRCAM / IRIS
Mario Corti Colleoni	Studio M.K.
Luigi Pestalozza	Musica / Realtà
Gianni Orlandi	Dip. INFOCOM / Facoltà di Ingegneria - Università di Roma
Luciano Berio	Istituto "Tempo Reale"

COMITATO ORGANIZZATORE

Nicola Sani
Serena Tamburini
Diego Fabrizio Tomassini

COMITATO SCIENTIFICO

Vito Asta	Ist. di Acustica "O. M. Corbino" - Roma
Giovanni De Poli	Università di Padova
Lindoro Massimo Del Duca	SIM Srl - Roma
Eugenio Guarino	SIM Srl - Roma
Goffredo Haus	Università di Milano
Gianni Orlandi	Università di Roma e Ancona
Aldo Piccialli	Università di Napoli

COMITATO MUSICALE

Mauro Bagella	Associazione Musica Verticale
Mario Baroni	Università di Bologna
Nicola Bernardini	Conservatorio di Cagliari
James Dashow	Compositore
Michelangelo Lupone	SIM Srl - Roma
Fausto Razzi	Compositore
Marco Stroppa	IRCAM - Parigi
Alvise Vidolin	AIMI / LIMB - Venezia

UFFICIO STAMPA

Diego Fabrizio Tomassini
Maria Alessandra Casadio

SEGRETERIA

Lucilla Telloni
Beatrice Cristini

Atti a cura di: Diego Fabrizio Tomassini

INTRODUZIONE AL VOLUME

Questo volume, che vede la luce grazie al prezioso contributo del CIDIM / Comitato Nazionale Italiano Musica (CIM / Unesco), raccoglie gli atti del VII Colloquio di Informatica Musicale che si tiene a Roma, Palazzo Taverna, sede dell'IN / ARCH Istituto Nazionale di Architettura, dal 23 al 26 marzo 1988.

Le circa quaranta * relazioni, tra cui quelle di ricercatori stranieri, sono state raccolte in nove sessioni:

- Hardware
- Software
- Elaborazione Numerica del Suono
- Applicazioni alla Musicologia
- Composizione Musicale
- Rapporti di Attività
- Dimostrazioni
- Posters
- di Ascolto

Un incontro / Dibattito sul tema "Ricerca, produzione e diffusione: quale futuro per l'informatica musicale", che vede la partecipazione di eminenti personaggi del mondo della cultura e dell'industria, e due concerti di musica informatica dedicati ad autori italiani e stranieri, completano il programma del Colloquio.

La ricerca e l'utilizzazione di nuove tecnologie sono oggi un elemento imprescindibile per chi opera in campo musicale. Non solo per la produzione di opere, ma anche per tutte le altre fasi di "lavorazione" della musica, fino alla sua diffusione e riproduzione. Con l'avvento del compact disc, degli strumenti musicali digitali e del personal computer, l'informatica musicale è uscita dai laboratori di ricerca per entrare a far parte del lavoro quotidiano di molti musicisti e tecnici, e questo anche a livello amatoriale. La visione che abbiamo oggi di questo mondo musicale è quindi mutata radicalmente in pochi anni. L'informatica musicale non è più un tema per pochi addetti ai lavori, ma un'area dove confluiscono interessi diversi, che spaziano dalla ricerca scientifica alla didattica, dal prodotto industriale al mondo dello spettacolo.

Il significato di questo VII Colloquio di Informatica Musicale è dunque quello di essere un attento testimone di una evoluzione culturale che ha portato il nostro Paese ad un livello di produzione pari a quello "storico" della Francia e degli Stati Uniti. Sono nate infatti nuove realtà imprenditoriali e culturali, come testimoniano anche l'interesse di aziende come la Texas Instruments, l'impegno di Enti come il CIDIM e il Ministero del Turismo e dello Spettacolo, la formazione di nuove strutture come la SIM di Roma, l'Istituto "Tempo Reale" di Luciano Berio a Firenze e il nuovo Consorzio IRIS.

Un'edizione, questa, volta non solo a presentare i risultati più avanzati della ricerca scientifica nel settore, ma anche a riflettere sul futuro dell'informatica musicale in Italia, intesa come sintesi di cultura ed innovazione tecnologica.

Diego Fabrizio Tomassini

* Alcune relazioni non sono presenti in questa pubblicazione perché non pervenute in redazione in tempo per la stampa.

INDICE

Sessione Hardware

Collegamento a fibra ottica per trasmissione di segnali audio digitalizzati .	pag.	1
Interfaccia programmabile con buffer FIFO per sistema di analisi e sintesi di segnali audio digitalizzati	»	6
Tecnica di interpolazione multipla	»	11
Ambiente di sviluppo multiutente per il processore 4X	»	14
Controllo MIDI non convenzionale per il sistema 4X	»	23
Campionamento del segnale con il sistema Soft Machine	»	33
A.S.F. an Audio Synthesis Family of VLSI chips	»	37

Sessione Software

Interazioni Musica-Movimento: un sistema per la generazione di partiture da animazioni 3D	»	42
Rappresentazione della conoscenza musicale: sistemi analogici e sistemi logico- simbolici	»	48
Definizione di specifiche per il collegamento fra tastiera musicale e sistemi di sintesi digitale del suono via Standard MIDI: un esempio applicativo ...	»	55
Linguaggio compositivo in ambiente MIDI	»	64
La Sound Processing Unit 03 (SPU 03): un piccolo sistema flessibile per uso prevalentemente didattico	»	68

Sessione Elaborazione Numerica del Suono

Forme d'onda per la sintesi granulare sincrona	»	70
Un convertitore Pitch-to-MIDI Stand-alone	»	76
Applicazioni della tecnica di distorsione non lineare a funzioni di ingresso in regime non sinusoidale	»	82
Sintesi per formanti: controllo mediante modulazione di fase	»	87

Sessione Applicazioni alla Musicologia

Performance rules for computer music	»	92
Un software per la generazione automatica e lo studio della musica jazz ...	»	97
Formalizzazione di un processo compositivo: lo "Studie 1" di Karlheinz Stock- hausen	»	102
Informatica e musicologia: un sistema per il recupero di "fondi" musicali	»	106

Sessione Composizione Musicale

Importanza della didattica per l'evoluzione dell'informatica musicale	»	113
Un sistema di aiuto alla composizione: Music Assistant	»	119
Analisi della composizione "Aquam flare in media labia tua"	»	125
Processi cognitivi come parametri compositivi	»	129

Sessione Rapporti di Attività

Il laboratorio di trattamento del segnale audio dell'Istituto di Acustica "O.M. Corbino" del C.N.R.	pag.	140
Recent work in A.I. & music with lines for the future	»	143
TELETAU: un software package per l'informatica musicale	»	147
Elaborazione dei suoni naturali e composizione con il sistema 4I	»	150
Considerazioni sui primi quattro anni di attività della SIM nell'informatica musicale	»	157

Sessione Dimostrazioni

Composer 1: un progetto di ambiente compositivo per una mini "stazione di lavoro" musicale	»	159
Elaborazione interattiva del segnale digitale	»	161
CRONOS: nuovo sistema a microprocessore per la conduzione gestuale in tempo reale di elaboratori musicali, sequencer e drum-machine	»	162
PRO.FUMO	»	167

Sessione Posters

La Tartamusa: ovvero l'oggettivazione dell'astratto musicale	»	169
Wavetable come automa cellulare: una nuova tecnica di sintesi	»	174
Collegamento MIDI per computers DEC PDP11 e VAX	»	178
Sistemi a basso costo per l'elaborazione del suono	»	179
Composizione tramite automi cellulari	»	181

Sessioni di Ascolto

L. Todoroff: Juego de velocidades	»	183
N. Sani: Tendenze	»	184
G. Tedde: Archi elettrici	»	185
F. Galante: Tra i suoni estremi	»	186
R. Doati: Deve essere tenuto lontano da fonti di luce	»	187
K. F. Gerber: Ting	»	189
M. Pedrazzi: RAN	»	190
B. Fagarazzi: Birdwatching	»	191
S. Petrarca: Campana del tramonto	»	192
M. Querzola: Villotta	»	193
A. Di Scipio: Hy-Lur	»	194
A. Caprioli: Per lo dolce silentio de la notte	»	195
M. Biasutti: Ritmi biologici	»	198
S. Rüzicka: Rota	»	199
K. Saariaho: Jardin Secret 1	»	200

SESSIONE HARDWARE

COLLEGAMENTO A FIBRA OTTICA PER TRASMISSIONE DI SEGNALI AUDIO DIGITALIZZATI

P. Massaruti, M. Rubbazer, G.B. Debiasi
C.S.C. - Università di Padova - Via Gradenigo 6/A - PD

Questo lavoro è stato sviluppato nell'ambito di un programma di analisi e sintesi di segnali vocali e musicali intrapreso dal Centro di Sonologia Computazionale (C.S.C.) dell'Università di Padova.

Tale programma prevede l'acquisizione, ad alta velocità di campionamento e ad alta risoluzione, di un massimo di quattro canali indipendenti, con successiva analisi nel dominio del tempo e della frequenza, e la generazione di segnali audio su un massimo di quattro canali a partire dai dati elaborati da un calcolatore, con le medesime specifiche di frequenza e risoluzione.

Il sistema di Acquisizione - Generazione si compone dei seguenti blocchi funzionali (vedi Fig. 1):

- Unità di conversione Analogico - Digitale e Digitale - Analogica (A/A e D/A).
- Interfaccia Programmabile per l'accumulo temporaneo dei dati in transito tra l'elaboratore e le unità A/D e D/A.
- Canale di trasmissione a fibra ottica.
- Sistema per l'elaborazione dei dati acquisiti e per la generazione dei campioni relativi alla sintesi.

Scopo del presente lavoro è il progetto e la realizzazione del canale di trasmissione a fibra ottica tra l'Interfaccia Programmabile ed il calcolatore; tale canale deve soddisfare alle seguenti specifiche:

- a) totale trasparenza del canale riguardo alle preesistenti modalità operative per la programmazione e lo scambio di dati tra il calcolatore e l'Interfaccia Programmabile;
- b) velocità di trasmissione compatibile sia con l'impiego di unità di trasduzione di tipo correntemente disponibile, che con l'utilizzo di circuiti logici con livelli TTL (escludendo pertanto i circuiti ECL).

Esaminando ora più da vicino le caratteristiche dei blocchi funzionali che compongono il sistema, si nota che le unità di conversione rendono possibile l'acquisizione e la generazione di segnali audio su 1, 2 o 4 canali indipendenti (mono-, stereo-, e quadrifonia); la frequenza massima di campionamento è di 50 kHz e la digitalizzazione ha luogo su 16 bit.

L'interfaccia programmabile svolge la funzione di coda FIFO (First-In-First-Out) bidirezionale, con una capacità di 64kWord: è pertanto in grado di accumulare i dati provenienti dai convertitori A/D (in fase di analisi) ovvero quelli destinati ai convertitori D/A (in fase di sintesi) sollevando il calcolatore dalla necessità di gestire in continuità il trasferimento di dati tra la memoria e le unità di conversione.

Una seconda funzione dell'Interfaccia Programmabile è quella di fornire ai convertitori la cadenza di campionamento/generazione in base ai valori impostati in un apposito registro;

tale registro è accessibile dal calcolatore tramite una linea di qualificazione, che indica all'Interfaccia se i dati inviati dal calcolatore devono venire immessi nella coda FIFO ovvero sono destinati al registro di controllo.

Da quanto esposto risulta chiaro che un utilizzo efficiente della capacità di accumulo dei dati nella coda FIFO è strettamente legato alla massima velocità di trasferimento tra l'Interfaccia Programmabile e l'elaboratore: ciò ha condotto, durante la prima fase dello sviluppo del sistema, alla scelta di un canale di trasmissione di tipo parallelo - asincrono con 43 segnali (2 linee dati unidirezionali da 16 bit ciascuna, 4 segnali di handshake, 6 segnali di interruzione ed una linea di controllo).

La scelta del protocollo asincrono consente la trasmissione alla massima velocità permessa dalle unità adibite alla transazione (Interfaccia Programmabile e canale DMA del computer); la scelta di un collegamento di tipo parallelo non impone d'altro canto vincoli di serializzazione e deserializzazione ai segnali da trasmettere.

Un evidente svantaggio legato a quest'ultima scelta è tuttavia l'impossibilità di effettuare collegamenti di lunghezza superiore a qualche metro senza incorrere in una degradazione delle caratteristiche di velocità e di immunità ai disturbi.

Queste considerazioni, unitamente alla necessità di ottenere un isolamento galvanico tra i circuiti del sistema di elaborazione e quelli del sistema di acquisizione, hanno condotto allo sviluppo di un canale di trasmissione su fibra ottica tra l'Interfaccia Programmabile ed il computer.

Dalle caratteristiche citate dei convertitori A/D e D/A si determina un flusso di dati variabile tra 0.8 Mbit/s (monofonia) e 3.2 Mbit/s (quadrifonia): ciò ha portato alla scelta di due collegamenti unidirezionali da 10 Mbaud NRZ su fibra in silice di tipo "graded index" per quanto concerne il supporto fisico della trasmissione.

La specifica di trasparenza del canale rispetto alle preesistenti modalità operative ha determinato la conservazione del protocollo asincrono con segnali di handshake (dato disponibile - dato accettato); considerazioni relative all'immunità dal rumore della unità di serializzazione (UDS) e di deserializzazione (UDD) hanno portato alla fine all'adozione di un formato con 1 bit di start e 2 bit di stop.

La struttura di ogni pacchetto di dati è risultata pertanto la seguente (v. Fig. 2 - particolare in alto a destra):

- n. 1 bit di start a livello alto;
- n. 1 bit di qualificazione dati (per pacchetti dal calcolatore all'Interfaccia Programmabile) oppure di segnalazione di "interrupt" (per i pacchetti nell'altra direzione);
- n. 2 bit relativi ai segnali di handshake;
- n. 16 bit di dati;
- n. 1 bit di stop a livello alto;
- n. 1 bit di stop a livello basso;

e pertanto ogni pacchetto conta 22 bit, dei quali 16 sono relativi ai dati, 3 ai segnali di controllo e 3 alla delimitazione del pacchetto stesso.

La realizzazione dell'UDS non ha comportato particolari soluzioni circuitabili: come illustrato in Fig. 2, il processo di serializzazione ha inizio quando uno dei segnali di interscambio (IAVL o OACC) oppure il segnale di qualificazione D/CTRL effettuano una transizione nel

verso indicato dentro al blocco relativo al "rilevatore di transizione".

Il processo quindi prosegue, caricando i dati da trasmettere nel registro a scorrimento da 22 bit e trasferendoli attraverso l'uscita seriale al convertitore elettro-ottico; un apposito contatore determina la fine della serializzazione dopo la trasmissione del secondo bit di stop, dopodiché l'UDS viene riportata nello stato iniziale in attesa di una nuova transizione.

Per quanto concerne l'UDD, il problema più impegnativo è risultato essere la progettazione di un circuito in grado di campionare la sequenza di bit, proveniente dal convertitore ottico-elettrico, entro una finestra temporale di ampiezza assegnata e centrata sulla mezzieria dell'intervallo occupato da ciascun bit.

Il problema è stato risolto con una Macchina Sequenziale composta da 4 Flip - Flop ("Selettore Fase del Clock di Campionamento"), la quale ha il compito di discriminare la fase del segnale ricevuto nei confronti della fase del clock locale.

Il circuito adottato ha il pregio di funzionare con una frequenza di clock pari a 2 fc, dove con fc si è indicata la frequenza di cifra dei bit trasmessi dalla UDS; l'impiego di circuiti a campionamento multiplo avrebbe richiesto una frequenza di clock pari ad almeno 4 fc, rendendo più complessa la temporizzazione dei vari segnali ed aumentando i problemi legati ai ritardi di propagazione.

Il registro a scorrimento dell'UDD restituisce i dati in formato parallelo: essi vengono trasferiti ad un insieme costituito da un registro di uscita, un multivibratore monostabile e due Flip - Flop. Essi hanno il compito di fornire in uscita dei segnali che rispettino le relazioni temporali presenti nei segnali in ingresso all'UDS e che erano andate perdute nel processo di serializzazione.

Come si può notare dalla Fig. 2, il registro a scorrimento fornisce anche un segnale di "sequenza corretta" qualora i bit di start e di stop siano presenti con le polarità e le relazioni temporali che competono loro: tale segnale ha lo scopo di abilitare o meno il trasferimento dei dati presenti nel registro a scorrimento verso il registro di uscita ed annessi Flip - Flop.

Si può infine osservare che la lunghezza sia del registro a scorrimento che del contatore che determina la fine del processo di deserializzazione è di 21 bit, mentre il pacchetto trasmesso è di 22 bit. La differenza di lunghezza è giustificata dal fatto che l'intervallo temporale relativo al primo bit ricevuto (start bit) viene impiegato dal circuito selettore della fase di clock di campionamento per portare a termine l'operazione di discriminazione: solo al termine di quest'ultima il clock potrà giungere al registro a scorrimento ed al contatore, i quali vedranno pertanto soltanto 21 dei 22 bit che compongono il pacchetto.

Le prove effettuate sul canale di collegamento a fibra ottica hanno fornito risultati eccellenti: il tasso di errore è dell'ordine di 1 su 10^9 , ed il flusso di dati scambiabile è di fatto limitato solo dalla necessità di includere il segnale di avvenuta ricezione in un pacchetto di 22 bit che non reca alcuna ulteriore informazione.

È pertanto emerso che il mantenimento del protocollo di trasmissione con segnali di interscambio non consente di andare oltre l'acquisizione o generazione su due canali a 50 KHz, nell'ipotesi di limitare il flusso di dati sulla fibra a 10 Mbit/s.

Utilizzando invece una fibra da 20 Mbit/s o più, è possibile sfruttare appieno le possibilità offerte dall'Interfaccia Programmabile, col funzionamento su quattro canali di conversione.

Si è infine esaminata la possibilità di utilizzare un protocollo di trasmissione "ad anello

aperto”, il quale impiega dei segnali di interscambio fittizi generati dall’UDS al termine di ciascuna serializzazione; dalle misure effettuate è risultato che un tale protocollo, impiegato con un canale da 10 Mbit/s, è perfettamente compatibile con l’acquisizione o generazione di segnali audio su 4 canali indipendenti ad una frequenza di campionamento di 50 kHz. Per di più, questo artificio consente anche di prescindere dalla lunghezza del collegamento a fibra ottica, e dell’associato ritardo di propagazione.

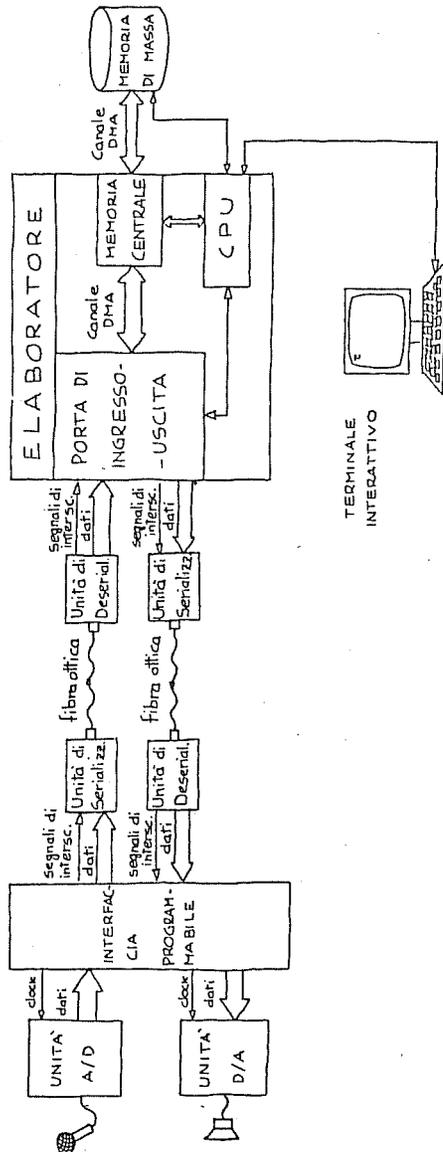


Fig. 1. - Schema a blocchi del sistema di Acquisizione - Generazione.

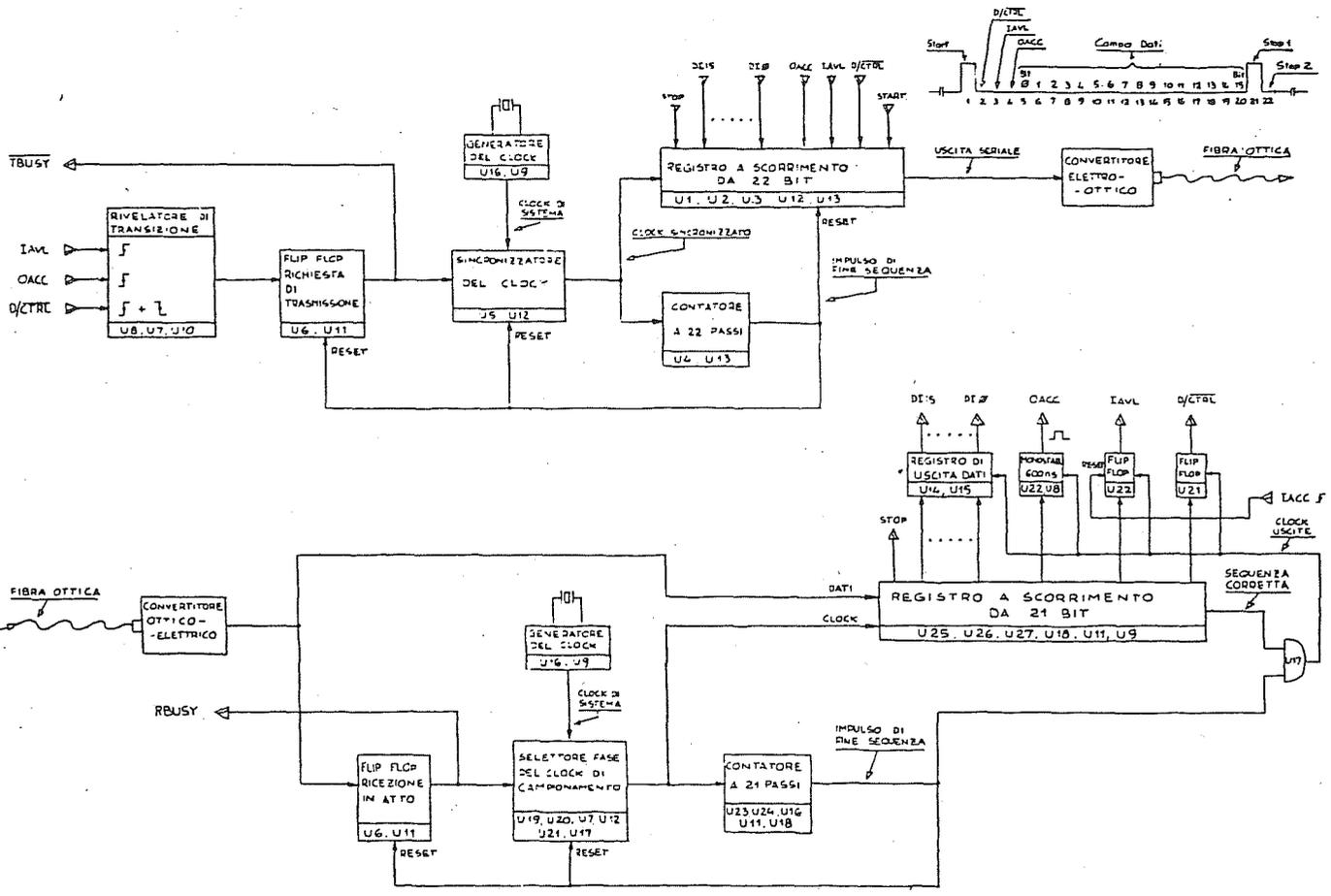


Fig. 2. - Schema a blocchi delle unità di serializzazione e deserializzazione.

INTERFACCIA PROGRAMMABILE CON BUFFER FIFO PER SISTEMA DI ANALISI E SINTESI DI SEGNALI AUDIO DIGITALIZZATI

G. Capuzzo, M. Rubbazer, G.B. Debiassi
C.S.C. - Università di Padova - Via Gradenigo 6/A - PD

L'interfaccia che sarà descritta qui di seguito è stata progettata per essere integrata in un sistema di analisi e sintesi sonora realizzato presso il Laboratorio di Sonologia Computazionale dell'Università di Padova.

Il sistema completo prevede, oltre all'interfaccia in oggetto, alcune unità di conversione A/D e D/A a 16 bit, un computer dotato di hard disk, ed, eventualmente, una unità di collegamento a fibra ottica per permettere il posizionamento del sistema di analisi e sintesi ad una certa distanza dal computer di controllo.

Le principali caratteristiche che contraddistinguono questa interfaccia sono:

- possibilità di collegare fino a 4 unità di conversione A/D e 4 unità di conversione D/A operanti ad una frequenza di campionamento massima di 50 KHz ciascuna;
- presenza di un buffer di memoria di tipo First-In First-Out (FIFO) da 64 KWord, organizzato a "coda circolare", per facilitare il trasferimento di lunghe sequenze di dati tra il computer e le unità di conversione;
- elevata velocità di trasferimento dei dati tra l'interfaccia ed il computer: 1.25 MWord/S massimo;
- possibilità di riconversione D/A immediata dei dati durante una acquisizione, per verificare il corretto svolgimento dell'acquisizione stessa (ECHO);
- possibilità di verificare il corretto funzionamento dell'interfaccia, sia da software, leggendo un opportuno "registro di stato", sia attraverso alcuni diodi LED posti sul pannello frontale dell'interfaccia;
- presenza di un generatore di periodo di campionamento, controllato a quarzo, programmabile da 16.0 μ s a 999.9 μ s, a passi di 100 nS;
- possibilità di selezionare tutte le varie opzioni, sia da software, scrivendo in opportuni "registri di controllo", che manualmente, tramite deviatori posti sul pannello frontale dell'interfaccia;
- possibilità di generare richieste di interrupt verso il computer, al verificarsi di eventi particolari (buffer quasi pieno, pieno, quasi vuoto, vuoto, errore, ecc.);
- presenza di alcune linee di I/O digitale, sotto controllo software, a disposizione dell'utente per applicazioni generiche.

Lo schema a blocchi dell'interfaccia programmabile è riportato in figura 1.

L'impostazione del progetto ha privilegiato soprattutto la versatilità d'uso e la facilità di collegamento con computers di tipo diverso. L'interconnessione fra interfaccia e computer viene effettuata attraverso due porte parallele unidirezionali a 16 bit più alcune linee di servizio. Attraverso la porta di uscita del computer vengono inviati all'interfaccia i dati destinati alle unità di conversione D/A, ma anche le "parole di controllo" necessarie alla programma-

zione delle modalità di funzionamento dell'interfaccia stessa. Attraverso la porta di ingresso, invece, il computer riceve dall'interfaccia i dati prodotti dalle unità di conversione A/D ed anche il contenuto del "registro di stato" contenente alcune informazioni sul funzionamento dell'interfaccia.

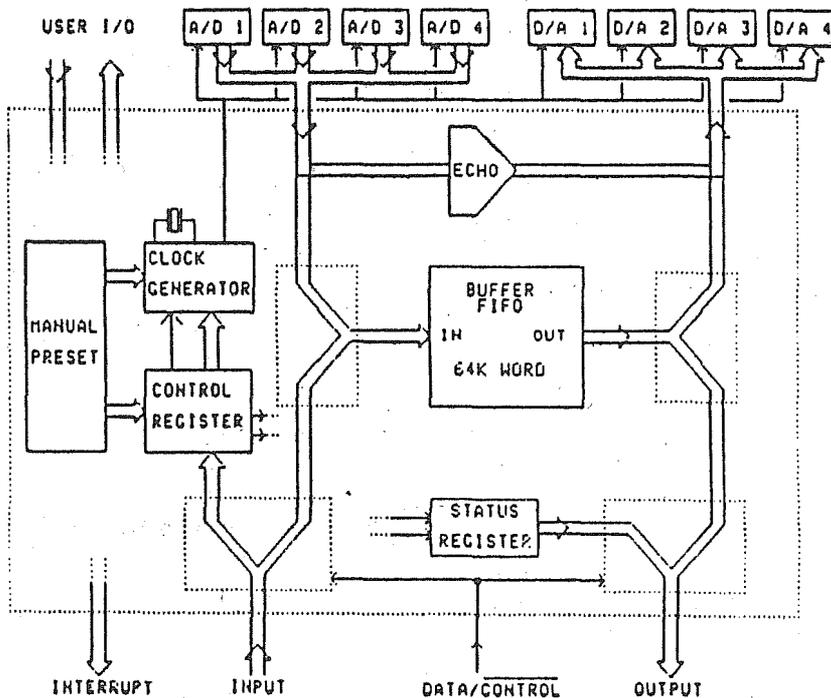


Fig. 1 - Schema a blocchi dell'interfaccia programmabile.

Per completare il collegamento con il computer è richiesta una ulteriore linea, in uscita da quest'ultimo, che qualifichi l'informazione in transito sulle porte parallele, come dato o come parola di stato/controllo, e uno o più ingressi per le richieste di interrupt.

Il collegamento tra l'interfaccia e le unità di conversione avviene attraverso due BUS DATI unidirezionali a 16 bit più un terzo bus unidirezionale per i segnali di controllo. Il bus dati di uscita unisce l'interfaccia alle 4 unità di conversione D/A, collegate in parallelo; analogamente il bus dati di ingresso unisce le 4 unità di conversione A/D, in parallelo, all'interfaccia. Il bus di controllo, infine, collega l'interfaccia con tutte le unità A/D e D/A in parallelo, e porta a quest'ultime tutti i segnali richiesti per un loro regolare funzionamento (clock di campionamento, selezione dei filtri antialiasing, alimentazioni, ecc.).

È possibile utilizzare un numero di unità inferiore al massimo consentito dall'interfaccia senza necessità di modifiche all'interfaccia medesima.

Entro i limiti imposti dal numero di unità di conversione installate, è possibile scegliere, via software, quante e quali unità utilizzare, tramite la scrittura di un'opportuna parola di controllo nell'apposito registro. Ad esempio, disponendo di 3 unità di conversione A/D, e volendo fare una acquisizione su due canali, è possibile scegliere tra le unità 1-2, 1-3 o 2-3.

Per ottimizzare la velocità di trasferimento dei dati tra il computer e le unità di conversione, lungo il percorso dei dati, all'interno dell'interfaccia, è stato inserito un buffer di tipo FIFO in grado di contenere fino a 64Kword da 16 bit.

Va ricordato, infatti, che mentre le unità di conversione richiedono un trasferimento di dati con cadenza costante (ad esempio un dato ogni 25 μ s, per frequenza di campionamento di 40 KHz), il computer effettua i trasferimenti molto velocemente (tipicamente un dato ogni μ s), ma dopo un certo numero di trasferimenti necessita di pause relativamente lunghe (qualche ms), per scambiare i dati con la memoria di massa (dischi magnetici, ecc.). Interponendo una memoria di tipo FIFO si ottiene una indipendenza tra le velocità di trasferimento dei dati tra il computer e il buffer FIFO e tra il buffer FIFO e le unità di conversione, ottenendo quindi uno sfruttamento ottimale delle risorse del computer.

Il buffer FIFO utilizzato in questa interfaccia è strutturato come una "coda circolare" di dati (figura 2).

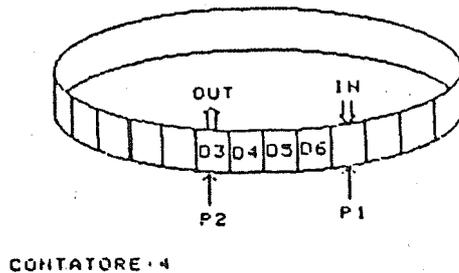


Fig. 2 - Buffer FIFO a "coda circolare".

Vi sono due puntatori P1 e P2 che indirizzano, rispettivamente, la prima locazione libera e l'ultima locazione occupata. Un dato in ingresso al buffer viene depositato nella locazione puntata da P1; P1 viene poi incrementato di uno, cioè viene spostato alla locazione successiva. Un dato in uscita dal buffer viene prelevato dalla locazione puntata da P2; P2 viene poi incrementato di uno. Un contatore up-down, incrementato in corrispondenza di ogni dato in ingresso e decrementato in corrispondenza di ogni dato in uscita, tiene traccia del livello di riempimento del buffer.

A buffer vuoto vengono impediti le operazioni di uscita, mentre a buffer pieno vengono bloccate quelle di ingresso.

Il computer può conoscere il livello di riempimento del buffer FIFO, eseguendo una lettura del registro di stato dell'interfaccia; quest'ultima può, inoltre, attivare una richiesta di interrupt verso il computer in corrispondenza del raggiungimento di un determinato livello di riempimento.

Un'eventuale tentativo di scrittura con buffer FIFO pieno provoca un errore di overflow; analogamente nel caso venga tentata una operazione di lettura con buffer vuoto, si ha un errore di underflow. Entrambe le situazioni di errore sono rilevate e memorizzate nel registro di stato; inoltre l'interfaccia provvede ad attivare una richiesta di interrupt verso il computer, affinché sia possibile intervenire opportunamente.

Il clock di campionamento necessario alle unità di conversione A/D e D/A, viene generato all'interno dell'interfaccia da un apposito circuito. Questo è formato da un divisore di frequenza, programmabile, a 4 decadi, che riceve in ingresso un segnale a 10 MHz, generato da un oscillatore controllato a quarzo. Il fattore di divisione utilizzabile varia da 160 a 9999. Il periodo di campionamento ottenibile varia perciò da 16.0 μ s a 999.9 μ s, a passi di 100 ns.

La scelta del periodo di campionamento viene fatta, da software, inviando le 4 cifre rappresentanti il numero di us codificate in BCD, nei registri di controllo dell'interfaccia, oppure manualmente, attraverso 4 commutatori rotativi posti sul pannello frontale dell'interfaccia.

È possibile anche utilizzare come ingresso per il divisore programmabile un segnale di clock proveniente dall'esterno. Questa opzione si può selezionare inviando un'opportuna parola di controllo al registro relativo. Attraverso questo stesso registro, inoltre può essere definito il livello logico di 4 linee di uscita digitale dell'interfaccia, a disposizione dell'utente per applicazioni particolari.

Come già accennato, il trasferimento di dati tra il computer di controllo e l'interfaccia, e viceversa, avviene attraverso due porte parallele a 16 bit unidirezionali ed è controllato da due linee di handshake per ogni porta (figura 3)

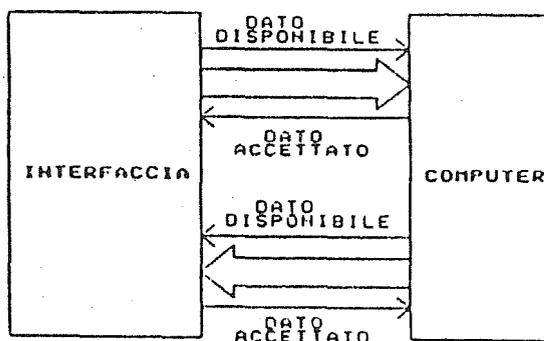


Fig. 3 - Collegamento tra computer e interfaccia con handshake.

Questi segnali sono del tipo "Dato Disponibile" - "Dato Accettato" e permettono un controllo completo del trasferimento. Infatti quest'ultimo viene abilitato solo fintantoché il buffer FIFO è in grado di fornire (o ricevere) dati validi, ovvero fino a quando non è vuoto (o completamente pieno).

Per quanto riguarda le modalità di trasferimento di blocchi di dati (cioè in pratica, il programma di gestione dell'interfaccia), queste sono diverse a seconda che il computer usi un I/O controllato direttamente da programma, o un trasferimento con DMA (Direct Memory Access).

Nel primo caso è la CPU che effettua i trasferimenti tra le porte di I/O del computer e la memoria centrale, per mezzo di istruzioni di I/O.

Per realizzare un trasferimento di questo tipo in maniera efficiente, è necessario utilizzare i due flag dell'interfaccia che indicano quando il buffer FIFO è "Quasi-Pieno" e "Quasi-Vuoto". Durante una generazione sonora, cioè trasferendo i dati dal computer verso l'inter-

faccia, la linea associata al flag "Quasi-Vuoto" viene collegata ad un ingresso di richiesta di interrupt del computer.

Il trasferimento inizia con l'invio al buffer FIFO di un "pacchetto" di dati di dimensione prefissata. Al termine di questo primo trasferimento la CPU può essere impiegata in altri task (per esempio la lettura di nuovi dati dalla memoria di massa), mentre il buffer FIFO si sta svuotando, fornendo i dati alle unità di conversione D/A.

Quando il numero dei dati ancora presenti nel buffer FIFO diventa inferiore ad una certa soglia (prefissata al momento dell'installazione), il flag "Quasi-Vuoto" si attiva, generando una richiesta di interrupt nel computer. La routine di servizio dell'interrupt deve semplicemente trasferire al buffer FIFO un altro "pacchetto" di dati (sempre di dimensioni fisse) e ripristinare i dati in memoria centrale, prelevandoli dalla memoria di massa. Il trasferimento prosegue in questo modo fino al suo completamento.

In maniera del tutto analoga (utilizzando, però, il flag "Quasi-Pieno", come richiesta di interrupt), si svolge il trasferimento di dati nel verso opposto, durante una acquisizione.

Nei trasferimenti con DMA, invece, i dati sono trasferiti direttamente dalle porte di I/O alla memoria centrale, e viceversa, da un apposito dispositivo del calcolatore (DMA controller), senza coinvolgere la CPU. Per questo motivo la velocità di trasferimento dei dati è nettamente superiore rispetto ad un I/O controllato da programma.

Utilizzando la tecnica DMA, il trasferimento dei dati può essere lasciato sotto il controllo delle sole linee di handshake, senza utilizzare le linee di richiesta di interrupt disponibili nell'interfaccia. La CPU ha, in questo caso, solo il compito di riprogrammare correttamente i registri del DMA controller ogni volta che quest'ultimo ha completato il trasferimento di un blocco di dati; nell'intervallo di tempo tra due successive riprogrammazioni dei registri del DMA controller, operazione per altro molto rapida, la CPU è libera di svolgere altri task.

L'interfaccia è stata progettata interamente in logica sparsa, utilizzando circuiti integrati della famiglia HCMOS che offre alta velocità unita a bassi consumi.

Un prototipo è stato realizzato e collegato sperimentalmente ad un personal computer APPLE II PLUS e ad un IBM PC AT. È stata inoltre verificata la possibilità di collegare l'interfaccia ad un computer VAX DIGITAL, attraverso una scheda di I/O parallela DRE11C.

TECNICA DI INTERPOLAZIONE MULTIPLA

E. Guarino

SIM - Via Riccardo Grazioli Lante, 30 - Roma

L'interpolazione è un semplice algoritmo il cui uso è invalso sempre di più negli ultimi anni in architetture che prevedono l'uso di un moltiplicatore, sia perché assicura un significativo miglioramento delle prestazioni di un sistema, sia perché consente di sfruttare al massimo le potenzialità offerte dal moltiplicatore giustificandone pienamente il "costo", ove per costo si può intendere il prezzo, nel caso di un sistema ad integrati discreti, ovvero l'area di silicio in un sistema integrato.

L'interpolazione consente una drastica riduzione della lunghezza di tabella di una forma d'onda allorché sia effettuata tra campioni consecutivi; inoltre permette di miscelare in percentuali prestabilite ed eventualmente variabili nel tempo, campioni provenienti da diverse tabelle realizzando così un'interessante dinamica timbrica, anche a livelli multipli.

Consideriamo, ad esempio, un sistema con una frequenza di campionamento di 40 KHz: per riprodurre una sinusoide di 20 Hz, sarà necessaria una tabella di circa 2K campioni.

Ma per la stessa sinusoide a 200 Hz ne verrebbero utilizzati soltanto 200, e a 2 KHz, appena 20.

Se invece adottiamo una lunghezza di tabella di 256 campioni interpolati con soli 4 bit di parte frazionaria della fase, la definizione della stessa viene aumentata producendo una tabella virtuale di 4K campioni ($256 * 2^{**4}$): si è ottenuto un risparmio in parole di memoria pari ad 1/8.

Inoltre, la moltiplicazione necessaria per l'interpolazione fornisce un certo numero di bit addizionali alla lunghezza di parola e consente pertanto di limitare tale lunghezza in memoria (ad esempio a 12 bit) restituendo un virtuale aumento della definizione in ampiezza (a 16 bit nell'esempio citato).

Queste considerazioni possono sembrare irriskorie in tempi di crollo dei prezzi delle memorie; si noti allora che per uno strumento con tre livelli di interpolazione ed appena 32 timbri, sono sufficienti 64K di memoria contro i 512K necessari con tabelle da 2K.

La soluzione interpolativa, d'altro canto, comporta una limitazione di banda relativa pari a 128 armoniche della sinusoide riprodotta: per una fondamentale di 20 Hz, ad esempio, la banda massima è 2.56 KHz.

Poiché ogni operazione di interpolazione richiede almeno una moltiplicazione, 2 addizioni e 2 accessi in memoria, è chiaro che un sistema interpolativo multiplo impone un'adeguata organizzazione del flusso di dati in ingresso e in uscita al moltiplicatore che ne costituisce il "cuore". Spesso risulta indispensabile un'architettura di tipo "pipeline" con un register-file (o una RAM) di transito per i risultati parziali; ma anche quando questo non fosse necessario, si avrebbero di fronte strutture abbastanza complesse da meritare alcune considerazioni.

Cominciamo col considerare una generica operazione di interpolazione, cioè una media pesata tra due valori di una tabella:

$$[1] \quad V(i) = A(i) * X(i) + B(i) * X(j)$$

in cui $X(i)$ ed $X(j)$ rappresentano i valori della tabella, "i" e "j" ne sono gli indici legati da una qualche relazione (spesso $i = j + 1$), $A(i)$ e $B(i)$ i coefficienti o pesi che possono essere normalizzati:

$$[2] A(i) + B(i) = 1$$

In tal caso la [1] diventa:

$$[1a] V(i) = A(i) * X(i) + (1 - A(i)) * X(j)$$

ovvero

$$[1b] V(i) = A(i) * (X(i) - X(j)) + X(j).$$

Per allacciarsi ad un caso particolare, consideriamo l'interpolazione tra campioni estratti da una tabella contenente una forma d'onda cui si accede usando la fase come indirizzo (Fig. 1); la fase è calcolata in base alla consueta formula:

$$[3] f(i) = f(i-1) + q$$

ove "q" è una costante proporzionale alla frequenza e la fase è composta dalle parti intera e frazionaria:

$$[3a] f(i) = If(i) + Ff(i)$$

Vale allora la relazione:

$$[4] (W*(i) - W(i))/Ff(i) = (W+(i) - W(i))$$

da cui si ricava:

$$[4a] W*(i) = W(i) + Ff(i) * (W+(i) - W(i))$$

ovvero

$$[4b] W*(i) = W(i) * (1 - Ff(i)) + W+(i) * Ff(i)$$

che esprimono due possibili implementazioni dell'interpolazione (Fig. 2); la scelta tra di esse verrà compiuta tenendo conto delle esigenze architettoniche del sistema.

Le definizioni

$$W(i) = W(If(i))$$

$$W(i+1) = W(If(i+1))$$

$$W+(i) = W(If(i)+1)$$

dovrebbero chiarire la differenza tra $W+(i)$ e $W(i+1)$: da esse si deduce che $W+(i)$ rappresenta il campione successivo a $W(i)$ nella tabella, mentre $W(i+1)$ è il campione indirizzato in corrispondenza della fase $f(i+1)$; è importante sottolineare che, in generale, sarà

$$W+(i) \langle \rangle W(i+1)$$

Si noti, inoltre, l'analogia tra le [1a], [1b] e le [4a], [4b].

Per semplicità di notazione, conveniamo di indicare il campione interpolato $W*(i)$ con una notazione più sintetica:

$$[5] Wf* = [Wf(i) , Wf+(i)]$$

avendo designato con [,] l'operatore di interpolazione con tutte le operazioni elementari che esso comporta.

Indicheremo pertanto l'interpolazione monodimensionale rispetto a variabili diverse dalla fase quali la frequenza, l'ampiezza, il timbro, con espressioni del tipo:

$$[5a] Wq* = [Wq(j) , Wq+(j)]$$

$$[5b] Wt* = [Wt(k) , Wt+(k)]$$

L'interpolazione composta tra due o più di tali variabili comporta il raddoppio delle operazioni elementari per ogni variabile, come mostra la formula per due variabili (le variabili

sono identificate dal rispettivo indice per non fissarne il tipo):

$$[6] \quad W^*(i,j) = [[W(i,j), W(i+,j)] , [W(i,j+), W(i+,j+)]]$$

mentre in generale si avrà una struttura ad albero (Fig. 3) la cui implementazione impone la ricerca del miglior compromesso tra l'architettura a totale parallelismo (massimo dispendio di H/W) e quella a totale divisione di tempo (massimo tempo di calcolo).

La strategia di interpolazione è stata prescelta nel progetto di uno dei VLSI (tecnologia CMOS - 2 micron) forniti dalla SIM alla ELKA: si tratta dell'AMX, un integrato capace tra le altre funzionalità di implementare interpolazioni multiple con interconnessioni microprogrammabili. Esso è dotato di due moltiplicatori ad operandi di tipo misto (l'uno "signed", l'altro "unsigned") basati sull'algoritmo di Booth generalizzato; questo algoritmo riduce ad $n/2$ ("n" è il numero di bit del moltiplicatore) gli addizionatori necessari.

La soluzione architetturale è di tipo misto: infatti un moltiplicatore 12×5 è dedicato esclusivamente all'interpolazione rispetto alla fase, mentre un 16×16 esegue tutti gli altri passi richiesti.

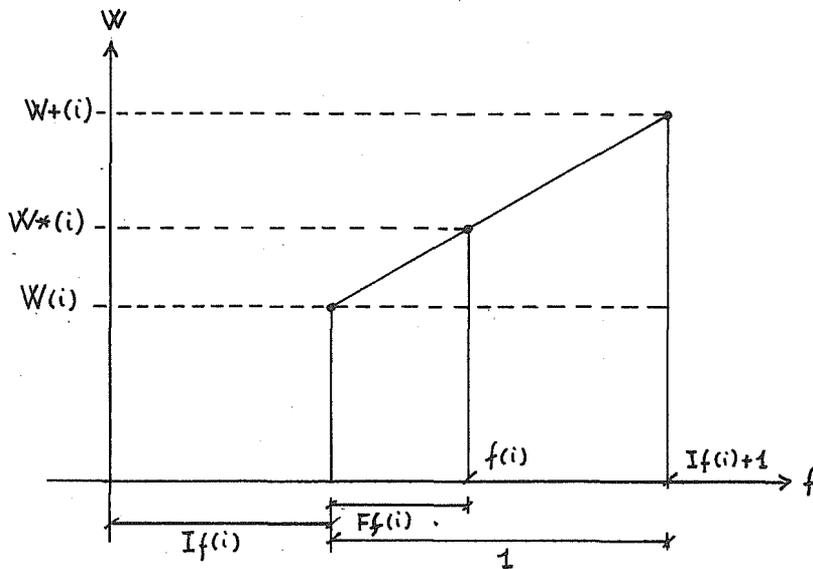


Fig. 1. - Interpolazione tra campioni consecutivi di una tabella

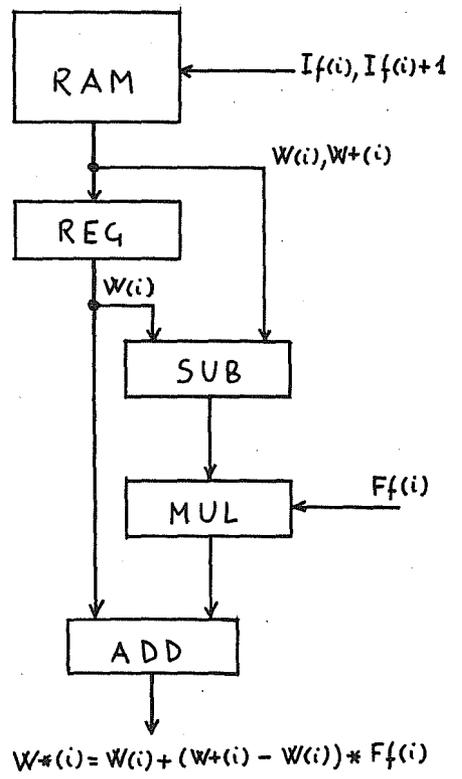
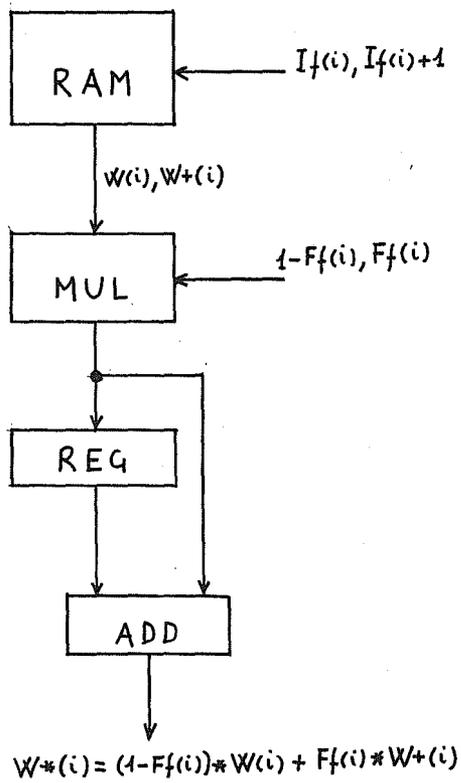


Fig. 2. - Schemi di implementazione delle [4a] e [4b]

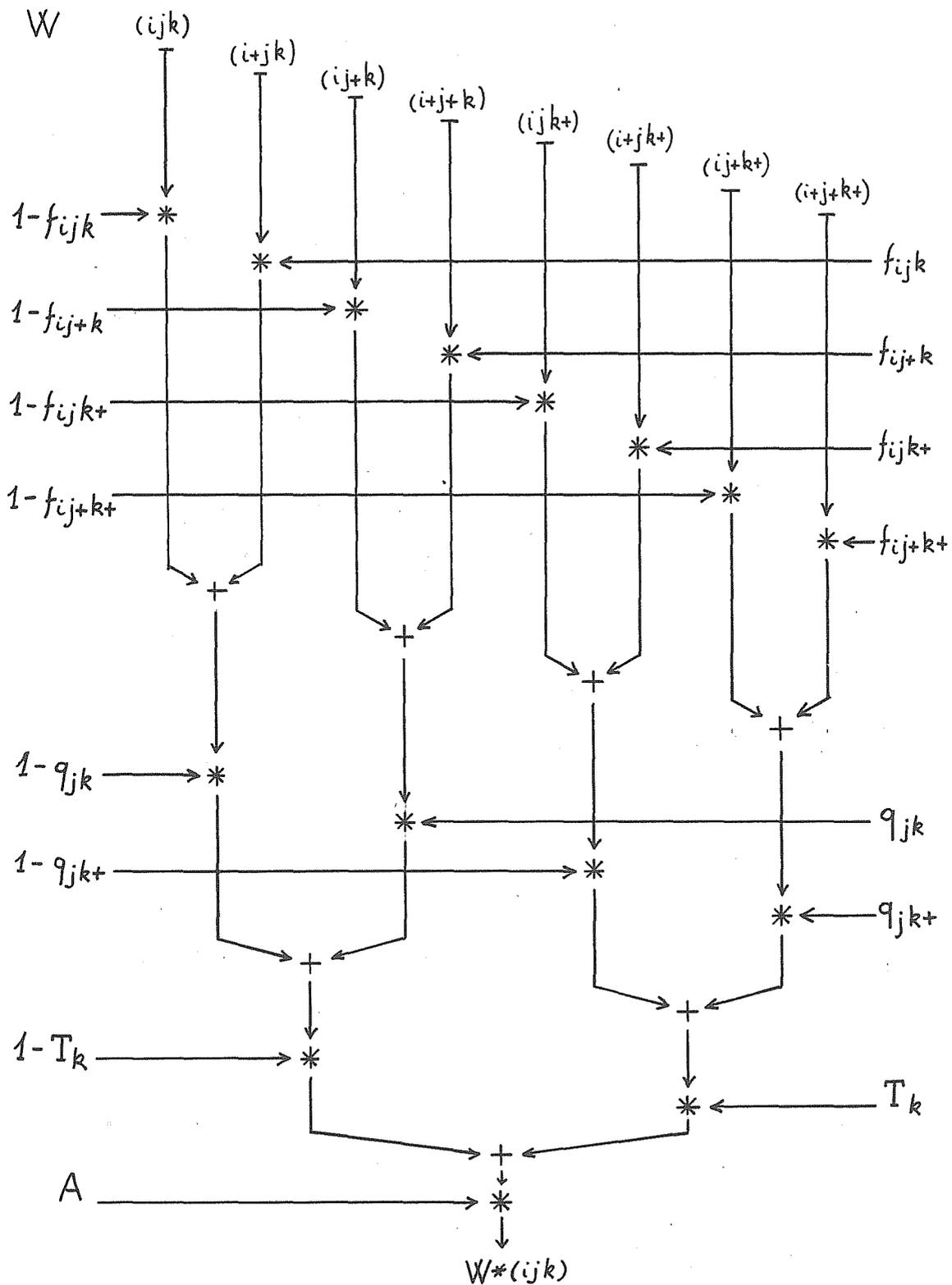


Fig. 3. - Flusso ad albero di un'interpolazione multipla

AMBIENTE DI SVILUPPO MULTIUTENTE PER IL PROCESSORE 4X

M. Rizzo, S. Sapir

C.S.C. - Università di Padova - Via Gradenigo 6/A - PD

Introduzione

L'uso dell'elaboratore in informatica musicale riguarda moltissimi aspetti della composizione, dall'elaborazione della struttura formale fino all'esecuzione del materiale sonoro.

Il tempo reale può essere utile in più situazioni: ad esempio nella fase di messa a punto degli algoritmi di elaborazione sonora, per tarare i valori di differenti parametri, o acquisire dei dati per mezzo di dispositivi gestuali durante l'esecuzione, fase in cui interagiscono interpreti e sistemi numerici.

Distingueremo due tipi di ambiente di lavoro musicale da realizzare attorno ai sistemi audio-numeric: un'ambiente non necessariamente operante in tempo reale per lo sviluppo e la definizione del materiale musicale operativo, ed un'ambiente per l'esecuzione di tale materiale. Verranno descritte di seguito le caratteristiche di un'ambiente di sviluppo realizzato per il sistema 4X, disponibile presso il Centro di Sonologia Computazionale (C.S.C.) dell'Università di Padova.

Uso attuale del processore 4X

Il Sistema 4X del CSC è composto dal primo prototipo del processore 4X, realizzato all'IRCAM da G. Di Giugno (5), con i suoi 16 canali DAC e 4 canali ADC, e dal suo controllore: il PDP 11/34 della Digital con il sistema operativo RT11SJ.

Il processore 4X è costituito, come indicato dalla Figura 1, da 9 schede comunicanti fra loro tramite un bus interno (P-BUS) diviso in tre parti. Queste schede possono essere divise in tre gruppi distinti: tre di queste servono al controllo dell'assieme (interfaccia con il PDP-11, controllo del bus interno e delle altre schede); le altre sei schede sono disponibili per l'elaborazione numerica del suono. Queste ultime schede, denominate "4U", sono identiche dal punto di vista del bus interno, e possono assumere configurazioni differenti in funzione di un microprogramma che le viene assegnato.

Una scheda 4U è composta da una unità funzionale (FU) e da tre blocchi di memoria: la memoria dei dati (DM), la memoria degli indirizzi (AM) e la memoria delle funzioni (WM), come si può vedere in figura 2. I dati prelevati dalla DM sono elaborati nella FU a seconda delle modalità fissate nel microprogramma associato, e reinserite dopo l'elaborazione nella DM; esistono inoltre due registri (IN-REG ed OUT-REG) che consentono lo scambio dei dati fra 4U e le operazioni di I/O con i DAC o gli ADC della 4X. Per informazioni più dettagliate vedere (1, 2, 8).

All'IRCAM e anche nei primi tempi al C.S.C., il PDP-11 veniva usato come unico controllore del processore. I compiti d'elaborazione dell'informazione musicale erano divisi fra 4X e PDP nel modo seguente: il processore 4X era interamente dedicato all'elaborazione in tempo reale del segnale sonoro, mentre il PDP elaborava l'informazione ad un più alto livello (controllo gestuale, partiture, schermate etc.) Il PDP-11 è stato quindi il protagonista principale per lo sviluppo di programmi per l'uso del processore 4X, e più in particolare il suo sistema operativo RT11SJ.

Il software è stato realizzato in gran parte in FORTRAN e MACRO-ASSEMBLER, per varie ragioni legate alla storia del sistema.

Esistono due programmi principali che consentono lo sviluppo di algoritmi di sintesi e l'esecuzione di partiture tipo MUSIC V: l'interprete 4R ed il programma PAR4X.

L'interprete 4R consente all'utente di definire in modo interattivo le varie connessioni fra moduli presenti nelle unità configurate secondo il microprogramma "4R"; questo interprete è lo strumento base per l'elaborazione dell'"orchestra" di una composizione.

Il programma PAR4X consente l'esecuzione di partiture scritte in un linguaggio simile al MUSIC V: il linguaggio NOT4X. I motivi che hanno portato alla realizzazione di tale programma provengono dall'esperienza già acquisita con il processore 4I [4], [6] e sono stati esplicitati in [2]. In particolare in questo ambiente di lavoro si è constatata l'insufficienza di potenza del PDP-11 rispetto a quella del processore 4X. Questo è dovuto in parte alla scarsa memoria disponibile (20 Kwords utili), e all'inadeguato tipo di comunicazione 4X-PDP per il trasferimento dei dati (dati di 24 bits trasmessi in due volte attraverso l'UNIBUS). Il PDP-11 è una macchina poco adatta al controllo del 4X, ma essendo l'unica fisicamente connessa al processore di suoni (il prototipo è stato pensato in funzione dell'unibus) si cercherà di scaricare al massimo i suoi compiti, inserendolo in una rete locale di elaboratori che consentono di definire una stazione musicale più potente e flessibile.

Analisi dell'uso del sistema 4X.

Per programmare il processore 4X è necessario eseguire tre tipi di operazioni:

- realizzare gli algoritmi di sintesi o di elaborazione del suono tramite interconnessioni dei vari moduli predisposti dalle 4U;
- creare la partitura operativa (programma FORTRAN o partitura NOT4X);
- eseguire la partitura: fornire tramite procedure gestite dal PDP i parametri in ingresso degli algoritmi di sintesi.

Il processore 4X è realmente necessario nel primo e nell'ultimo dei tre punti di cui sopra. Entrambi rappresentano nel migliore dei casi il 10% dell'utilizzo effettivo del sistema 4X. Per il resto, il sistema viene utilizzato come "macchina da scrivere" per l'editing dei file sorgenti (strumenti e partiture) e per la compilazione e l'elaborazione dei file eseguibili.

Queste ultime operazioni potrebbero essere realizzate senza l'impiego della 4X, eccezione fatta forse per la messa a punto degli algoritmi di sintesi, che richiedono alcune prove e verifiche. Si è quindi pensato di dividere in due tipi di ambienti la realizzazione del software per il processore 4X: un ambiente dedicato allo sviluppo del progetto musicale da realizzarsi all'esterno del sistema PDP-4X, e un ambiente dedicato all'esecuzione del pezzo che necessita la presenza del 4X.

Il PDP è stato quindi inserito in una rete locale di personal computer che durante l'esecuzione sono incaricati di gestire partiture e dati gestuali. Il PDP riceve dei dati operativi che manda al processore 4X sincronizzandosi con una mappa interna di eventi previsti e con i timers del processore; per ulteriori informazioni rimandiamo all'articolo [martin, sapir 1988] e a [2].

Con ambiente di sviluppo si intendono tutti quei supporti per la preparazione della sezione di esecuzione: elaborazione di files eseguibili e prove elementari per assicurare il buon funzionamento degli algoritmi realizzati. Per poter poi effettivamente provare ed usare il software

realizzato esternamente al sistema 4X-PDP, è stato necessario realizzare una simulazione del processore 4X, che ci consenta di verificare come funzionano gli algoritmi programmati.

Il sistema 4X in una rete di computer.

Una rete di computer è una struttura che rende disponibili all'utente dati e funzioni di calcolo remote. I motivi che rendono vantaggioso il collegamento di computer ed altri dispositivi in una rete sono di diverso tipo.

La condivisione delle risorse resa possibile dalle reti locali produce un forte risparmio nei costi ed un miglioramento delle prestazioni: dispositivi potenti e costosi e "data base" remoti sono utilizzabili da ciascun computer del sistema.

Software standard di comunicazione: file transfer

Oltre ai dispositivi elettronici necessari alla comunicazione tra computer, quali porte logiche di ingresso/uscita, è necessaria la presenza di software di sistema che gestisca lo scambio delle informazioni.

Un esempio è il "file transfer", che permette lo scambio di flussi di informazioni tra computer.

Un protocollo file transfer (FTP - File Transfer Protocol) definisce un set di regole per trasferire dei file da un sistema di computer ad un altro.

Il file transfer KERMIT, realizzato alla Columbia University, permette il collegamento tra due elaboratori attraverso una porta seriale TTY (RS-232).

L'idea base del Kermit non è quella di un programma portatile, ma bensì di un protocollo portatile. Ciò significa che ogni computer potrà avere la sua versione di Kermit che differirà anche in modo sostanziale dalle altre implementazioni ma tutte queste comunicheranno attraverso un protocollo comune.

Le funzioni basilari del Kermit sono sostanzialmente due:

- può essere usato per far lavorare un PC come terminale di un mainframe (terminale virtuale);
- per trasferire dei file da un computer ad un altro (file transfer).

Per maggiori dettagli sulle modalità di funzionamento del KERMIT si veda [3].

Descrizione del sistema realizzato

Le macchine utilizzate per realizzare una rete locale attorno al processore 4X sono, come evidenziato in figura 3:

- VAX 8600
- PDP 11
- PC IBM
- APPLE II
- ATARI SC

Il nodo UNIPAD della rete DECnet è costituito da un computer VAX/8600 della Digital Corporation dotato di sistema operativo VMS che supporta un certo numero di terminali in multiutenza.

Il PC è un microcomputer compatibile IBM ad 8 bit basato sul microprocessore 8086 della INTEL, e dispone di una porta seriale compatibile RS-232.

L'APPLE II è un microcomputer ad 8 bit basato sul microprocessore 6050 e dispone anch'esso di uno slot fornito di porta logica RS-232.

Infine l'ATARI SC è un microcomputer a 16 bit basato sul microprocessore 68000 della Motorola e dispone oltre che della porta RS-232 anche di una interfaccia MIDI.

Collegamento VAX PDP

Attualmente, in via del tutto provvisoria, i collegamenti si effettuano tramite un PC mediante le linee di terminale VAX e PDP. Un collegamento più raffinato sarà reso possibile dall'installazione di una rete locale (MNET) con struttura a bus, costituita da tante linee quanti sono i computer disponibili.

Ogni stazione di lavoro disporrà di una serie di slot che corrisponde alle diverse linee, per cui sarà sufficiente cambiare lo slot per connettersi al sistema desiderato. È da notare che questo tipo di collegamento prescinde, per quanto riguarda lo scopo di questo lavoro, dal particolare protocollo di comunicazione utilizzato.

Obiettivi di un ambiente sviluppo

Si può pensare di inserire il sistema PDP-4X, attualmente inscindibile in quanto le due macchine sono da considerarsi un tutt'uno dal punto di vista hardware (per il fatto che condividono fisicamente l'uso di alcune memorie), in una rete di computer più potenti e versatili per quanto riguarda la velocità di calcolo. La potenza di tali sistemi si estrinseca anche in una maggiore disponibilità di memorie e di "tools" di programmazione più potenti, quali compilatori di linguaggi più avanzati, routine di gestione grafica, data base più estesi ed altro. Inoltre un sistema a rete presenta il vantaggio di poter mettere a disposizione ad un numero maggiore di utenti alcune delle risorse del processore e dà la possibilità di poter utilizzare il 4X in modo "server" tramite il PDP, mediante il quale l'elaborazione dei testi musicali può essere effettuata ordinate da una qualsiasi località remota tramite un terminale video ed un collegamento alla rete telefonica.

Il primo passo di tale politica di gestione delle risorse disponibili consiste nel predisporre un collegamento permanente tra il sistema 4X e la rete DECnet per il computer VAX. Il secondo passo è il trasporto del software scritto per il pdp su VAX, che tende a migliorare l'uso del processore 4X (quando è possibile) oppure a snellire le operazioni del computer di controllo, realizzando comunque una sorta di "multiutenza virtuale" del processore in quanto almeno per la parte di editing, precompilazione ed analisi un utente può lavorare esclusivamente ad un terminale VAX, pur non "vedendo" il processore 4X, se non nella fase di esecuzione vera e propria.

Realizzazione dell'interprete 4R su VAX 8600

Nella realizzazione dell'interprete su VAX, per rendere compatibili le due versioni, si è

cercato di mantenere il più possibile inalterata la struttura del programma principale, e di fatto molte delle sue librerie in Fortran sono copie delle originali.

Le librerie scritte in Macro 11 sono state radicalmente modificate nel tentativo di tradurle in Fortran, ma di fatto le subroutine in esse contenute eseguono lo stesso tipo di operazioni.

L'eliminazione del software scritto in linguaggi fortemente dipendenti dalla macchina, quali ad esempio quelle in Macro, e la parziale modifica delle librerie Fortran rendono il programma 4R portabile. Il software implementato su VAX viene ad assumere così una caratteristica di indipendenza dalla macchina, e lo sforzo rivolto alla sua portabilità verso altri sistemi diventa pressoché trascurabile.

Saranno ora brevemente descritte le modalità tecniche di implementazione.

Nel PDP vengono mappate tutte le DM e solo una AM alla volta (compresa la IM), fatto che è gestito automaticamente dall'hardware. Non essendo possibile ciò nel VAX, le memorie sono costantemente mappate in memoria centrale mediante array multidimensionali i cui indici rappresentano il numero della unità e l'indirizzo relativo di memoria.

I registri di controllo sono delle semplici variabili di memoria, ma di fatto solo per alcune di esse il programma di simulazione assegna un significato reale, mentre le altre sono ignorate.

Poiché l'interprete dei comandi del PDP può accedere alle memorie del 4X solo tramite alcune routine macro, si è cercato di mantenere lo stesso tipo di organizzazione rendendo globali a tutto il programma sia gli array rappresentanti le memorie stesse che le variabili rappresentanti i registri di controllo. In questo modo le strutture di dati sono condivise da tutte le sezioni del programma che ne fanno uso, proprio come lo sono nella realtà le memorie del 4X, e si è tra l'altro realizzata l'assoluta trasparenza della presenza della macchina o della sua simulazione nei confronti dei segmenti di codice che non sono stati modificati nel trasporto da PDP a VAX.

Il trattamento di segmenti di memoria riservati ad un particolare uso del 4X quali i registri di interfaccia e controllo e la Intconnection Memory sono stati riprodotti fedelmente per permettere il salvataggio del contenuto delle memorie con lo scopo di poterle trasferire da VAX a PDP e quindi su 4X per la sintesi effettiva. Per maggiori dettagli sull'implementazione delle librerie di sottoprogrammi si veda [3].

Simulatore di una unità di sintesi.

Lo scopo della simulazione del funzionamento di una unità di sintesi del processore 4X è quello di poter verificare, seppur grossolanamente, il funzionamento degli algoritmi progettati. A tale proposito, non essendo la velocità di calcolo l'obiettivo preponderante, il simulatore è stato munito di alcuni comandi di debugging che ne rallentano l'esecuzione, ma permettono una più accurata fase di check degli algoritmi.

L'idea sulla quale ci si è basati nella implementazione dell'algoritmo di simulazione è quella della costruzione di una sequenza di azioni che più si avvicini a quella reale della macchina. Ciò consente anche la simulazione accurata della struttura del microprogramma, che può essere modificato variando un insieme di flag che rappresentano il firmware di una scheda.

Il nucleo del programma consiste in un ciclo di 32 iterazioni (il miniciclo del 4X) durante ognuno dei quali si legge la parte indirizzo delle corrispondenti locazioni della AM ed il data puntato in DM. Tale dato viene poi passato alla routine di simulazione della Functional Unit (FU) allo stesso modo in cui la logica di indirizzamento interno del 4X passa il dato al bus apposito si veda figura 2.

La routine preposta all'esecuzione delle operazioni aritmetiche utilizza un insieme di flag contenuti in alcune matrici che rappresentano il microprogramma 4R, in quanto nella 4X il passaggio dei dati ad un registro della FU avviene solo se è settato il suo segnale di abilitazione che corrisponde ad un segnale proveniente dalla Program Unit.

Secondo questa logica si è pervenuti alla costruzione di diverse strutture dati rappresentanti le ROM dei microprogrammi, in cui per ogni microciclo sono disponibili i "segnali" di abilitazione o disabilitazione dei vari registri.

La routine di simulazione della unità aritmetico-logica consiste dunque nel testaggio, microciclo per microciclo, di tutti i flag in una sequenza che riproduce tutti i ritardi causati dalle linee di trasmissione e dalla diversa velocità dei dispositivi di calcolo. Di volta in volta, a seconda delle necessità, verranno attivate le routine ausiliarie che effettuano la moltiplicazione su operandi a 16 bit e restituiscono il risultato su 24, e per l'accesso in lettura e scrittura alla WM etc.

Una volta fissata la programmazione della AM e collocati gli opportuni valori nella DM è possibile effettuare una simulazione dell'elaborazione dei campioni mediante il comando di accensione della macchina (SYON). Per ovvie esigenze di visualizzazione del risultato, è possibile testare uno solo dei 16 canali alla volta. Dopo l'avvio della simulazione compare un menu principale mediante il quale si può scegliere tra le modalità "simulazione" o "debugging".

Per quanto riguarda la simulazione si distinguono la modalità con output continuo e quella con output finale. Per entrambi i casi è necessario fissare il numero di millisecondi di elaborazione "reale", calcolati in base al tempo di esecuzione di un miniciclo che è di 62.5 microsecondi (in pratica si tratta di indicare il numero di minicicli che si intende far eseguire alla macchina).

Nel primo caso l'uscita viene visualizzata su di un grafico verticale a valori reali normalizzati a $+/- 40$ che si aggiorna dopo l'esecuzione di 64 cicli macchina completi, che eventualmente può essere stampato su carta a modulo continuo ottenendo così l'evoluzione della forma d'onda nel tempo. Nel secondo caso invece viene effettuata una stampa normalizzata al valore massimo e minimo della forma d'onda sugli ultimi 80 punti calcolati.

Scegliendo l'opzione debugger si accede ad un altro menu che permette la scelta di sei alternative. Tre di queste riguardano il debugging continuo su un certo numero di minicicli, mentre le altre il debugging passo-passo di ogni miniciclo. È possibile visualizzare il contenuto dei principali registri della FU con informazioni riguardanti il numero di microciclo ed il codice dell'operazione ALU, oppure il contenuto di 5 locazioni definibili dall'utente ed infine le uscite dei moduli di sintesi.

La prima versione della simulazione dell'ambiente 4R funziona sul VAX 8600 dell'Università di Padova. L'accesso al programma (4R.EXE) e alla documentazione (4R.DOC) richiede la specifica seguente:

```
UNIPAD: DISK $ USERS: MUSIC01 (4R/4R.DOC).
```

Per maggiori dettagli si veda [3].

Conclusioni

L'analisi dei risultati del lavoro svolto da M. Rizzo presso il CSC dell'Università di Padova portano a dover fare alcune considerazioni. Ad esempio per quanto riguarda la parte strettamente connessa alla realizzazione di un collegamento permanente tra i computer del

centro, le uniche funzionalità di una rete locale finora implementate riguardano il trasferimento dei file. Le operazioni che si possono eseguire dal sistema di comunicazione riguardano quasi esclusivamente file eseguibili dal sistema 4X o dal sistema simulatore VAX. In particolare è possibile:

- trasferire la mappa delle memorie del 4X; ciò consente di costruire e provare un algoritmo di sintesi (strumento) in un ambiente esterno al sistema 4X, per poi farlo eseguire realmente al processore di suoni;
- trasferire file di campioni audio digitalizzati in formato 4X;
- trasferire file di testo contenenti la definizione di partiture per i programmi di compilazione e di esecuzione di partiture ed eventualmente i file eseguibili prodotti dal compilatore di partiture.

La realizzazione del sistema di comunicazione consente inoltre il trasferimento di tali file tra PDP (o VAX) e un qualsiasi Personal o Home computer. Ciò significa che per editare file di comandi e partiture non sarà più necessariamente richiesto l'accesso ad un terminale connesso al 4X.

I miglioramenti indotti dalla disponibilità di una stazione di lavoro virtuale per il 4X sono svariati. Si possono ricordare tra tutti:

- la possibilità di utilizzare un debugger per la messa a punto degli algoritmi di sintesi;
- la possibilità di visualizzare i campioni elaborati in modo grafico;
- il fatto che almeno per la parte di editing e prova degli algoritmi (che richiedono la maggior parte del tempo di una sessione di lavoro) si può utilizzare una delle stazioni virtuali;
- la portabilità del programma di simulazione unita alla disponibilità della rete DECnet moltiplica ad un numero teoricamente indefinito il numero delle stazioni di lavoro;
- per quanto riguarda le partiture il suo utilizzo è limitato solo alla editazione od al massimo alla precompilazione dei testi musicali;
- l'ascolto dei risultati sonori avviene dopo il trasferimento dei file contenenti la mappa delle memorie del 4X, da VAX a PC e quindi a PDP.

Prospettive future

Si sta lavorando attualmente ad una seconda versione più sofisticata che permette di sviluppare del software di base per altri tipi di microprogrammi (4A, 4M, 4C). Quest'ultima versione consente inoltre di sviluppare i microprogrammi stessi, poiché verrà simulato l'intero svolgimento di indirizzamento delle ROM delle piastre. Il primo obiettivo di verifica di questa simulazione è la realizzazione, per il 4X, di un microprogramma 4I. Questo microprogramma sarà effettivamente integrato nelle ROM delle 4U, e ci consentirà di rendere ancora più compatibili i sistemi 4I e 4X. Sarà in effetti possibile trasferire sul 4X l'esperienza e i programmi acquisiti sulla 4I. È utile segnalare che disponendo di risorse hardware più sofisticate, si prevede la realizzazione di una seconda versione del 4I con l'aggiunta di oscillatori ad interpolazione (uso dello shift-register).

A questo punto il simulatore del processore 4X sul VAX può essere utilizzato da chiunque (purché sia collegato in rete con l'Università di Padova), per sviluppare firmware software per il processore 4X stesso.

D'altra parte, è stato attuato un programma di sviluppo di software su personal computer ATARI e PC IBM per l'esecuzione di partiture "acustiche e gestuali" in tempo reale con il processore 4X. In questa ottica di sviluppo, il PDP-11/34 è sempre più ridotto allo stato di "slave" di un sistema modulare più complesso e diversificato. Alla fin fine, il PDP viene usato essenzialmente per la gestione dei timers del processore e la temporizzazione dei parametri inviati nelle memorie dei dati delle 4U. Si tende quindi a realizzare al di fuori del sistema 4X-PDP dei programmi musicali generali, seguendo una implementazione stratificata, il cui ultimo strato viene legato alla struttura fisica del processore audio-numerico, ma i cui punti di partenza sono indipendenti da questo.

Bibliografia

- 1). V. Giannotti: *Moduli di trattamento in tempo reale del suono sul processore 4X*, Università di Padova, 1986.
- 2). G. Zorio Maulen: *Gestione del processore 4X per l'esecuzione in tempo reale di partiture*, Università di Padova, 1986.
- 3). M. Rizzo: *Ambiente di sviluppo per processore di suoni 4X*, Università di Padova, 1986.
- 4). S. Sapir: *Descrizione del programma NOT4I*, Bollettino LIMB 4, 1984.
- 4). V. Asta, A. Chaveau, G. Di Giugno, J. Kott: *Il sistema di sintesi digitale in tempo reale 4X*, *Automazione e strumentazione*, Volume XXVIII, Febbraio 1980.
- 6). S. Sapir, A. Vidolin: *Interazioni fra tempo e gesto*, Bollettino LIMB 5, 1985.
- 7). K. Gee: *Local Area Networks*, Manchester, NCC Publications, 1982.
- 8). S. Sapir: *Processeurs audio-numériques temps réel et informatique musicale: mise en œuvre d'un système d'exploitation et utilisations pour la recherche, la production et la pédagogie musicales, thèse de Doctorat d'Etat*, Université d'Aix-Marseille II, 1987.

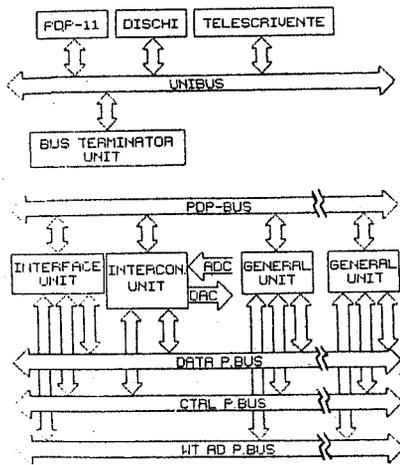


fig. 1:
architettura del 4X e
struttura dei bus

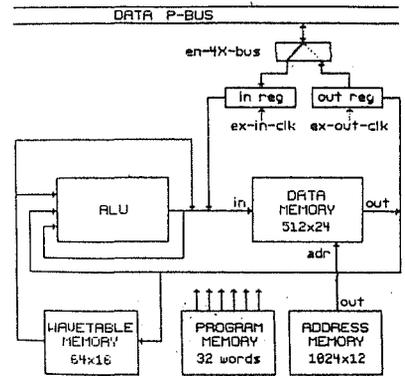


fig. 2:
architettura delle unità di
elaborazione

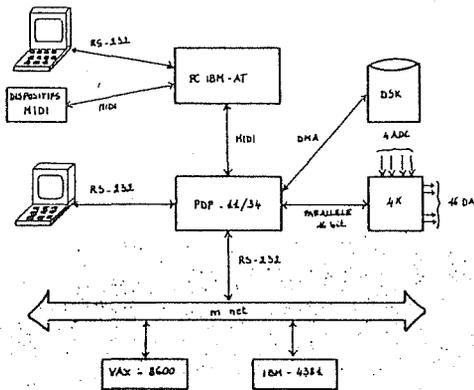


fig. 3:
stazione di lavoro musicale 4X

CONTROLLO MIDI NON CONVENZIONALE PER IL SISTEMA 4X

E. Martin, S. Sapir
C.S.C. - Università di Padova - PD

1. INTRODUZIONE.

I problemi qui discussi riguardano le possibilità di sfruttamento dei dispositivi commerciali MIDI, già connotati nell'ambito delle applicazioni musicali, per controllare in modo non convenzionale un processore audio-numeric programmabile di grossa potenza come il processore 4X.

Uso convenzionale del MIDI.

Il MIDI, è un protocollo di comunicazione che consente di trasmettere e ricevere informazione gestuale ben collocata nell'uso delle tastiere e altri strumenti digitali commerciali. L'uso convenzionale del MIDI presenta alcuni vantaggi. Uno di questi è la possibilità di suonare non appena lo strumento è stato acceso. In questo caso l'informazione gestuale proveniente dai dispositivi periferici (tastiere, cursori, pulsanti ecc.) è direttamente acquisita ed "interpretata" dallo strumento in funzione della propria configurazione per la sintesi del suono. Questa elaborazione fondamentale è in genere effettuata grazie ad un microprogramma inerente lo strumento.

Un secondo vantaggio deriva dalla semplicità del protocollo, e quindi dalla possibilità di elaborare in altre sedi di quelle dello strumento stesso l'informazione MIDI. Si tratta di gestire più dispositivi gestuali, di lavorare con dei "sequencer", di registrare e di editare le funzioni gestuali, ecc..

Il musicista può basare parte del suo lavoro in tempo reale su dei dati relativamente astratti, il cui "significato sonoro" verrà dato in un secondo momento dallo strumento di sintesi stesso.

Gli svantaggi derivanti dall'uso del MIDI sono noti e dipendono dai limiti intrinseci di uno standard e di scelte commerciali. Alcuni vengono superati grazie all'uso dei "dati esclusivi", che consentono di definire il proprio pacchetto di "istruzioni gestuali", e pilotare i particolari dei sintetizzatori, aumentando di conseguenza le possibilità del MIDI. A questo punto, il protocollo diventa sempre meno standard, e si sfrutta solo la possibilità di comunicare tramite una linea seriale relativamente veloce.

Le macchine "aperte" e lo standard MIDI: esempio del processore 4X.

È possibile connettere dei dispositivi MIDI a qualsiasi computer che disponga almeno di una interfaccia appropriata e del software di base per gestire l'acquisizione o la trasmissione dei dati gestuali. In questo caso si possono pilotare tramite gesti dei processori audio-numeric non commerciali. L'interesse risiede nella completa libertà di definire per i dati MIDI dei nuovi "significati sonori", tramite programmi adatti a questo compito, al di fuori degli standard abituali.

Quando il sintetizzatore è di uso generale e ha una struttura "aperta", cioè quando è

interamente programmabile, come nel caso del processore 4X, allora diventa difficile utilizzare il MIDI. Se si vuole rimanere "fedeli" allo standard, si perde la generalità del processore e si tende ad utilizzarlo in configurazioni convenzionali, che impediscono di sfruttare la sua potenza. Se si pensa ad utilizzare il MIDI in modo "esclusivo", allora bisogna creare delle biblioteche di casi "MIDI", cioè di configurazioni di sintesi con i loro programmi di controllo MIDI associati.

In queste condizioni, si crea un'ambiente relativamente modulare che dipende poco dal MIDI; nel quale ognuno può creare le proprie regole interpretative in funzione dell'uso e delle configurazioni particolari delle schede del processore 4X.

2. INFORMAZIONI CONTENUTE IN UN FLUSSO DI DATI MIDI.

Consideriamo un flusso di dati MIDI: essi possono essere considerati come il "filmato" di quello che sta succedendo sui dispositivi gestuali utilizzati. Se ci fosse un unico tipo di dati, ad esempio un unico potenziometro funzionante, avremmo allora una successione di dati rappresentanti i singoli campioni di una funzione del tempo, icone dei movimenti del cursore. Siccome esistono vari tipi di dati e una unica linea seriale di trasmissione, i dati effettivi sono preceduti da altri dati che consentono di identificare l'informazione trasmessa e la sua provenienza. In effetti il formato MIDI segue lo schema: STATUS-IDENTIFICAZIONE-VALORE/I. Questo flusso, viaggiando a 31.25 Kbaud, ci consente di simulare un certo tipo di parallelismo nell'uso dei dispositivi gestuali.

Tale flusso di dati contiene quindi informazioni svariate che possono essere impiegate per vari tipi di controllo. Purtroppo l'uso convenzionale del MIDI ripropone i modelli esecutivi tradizionali della musica; nei quali la produzione del suono è dipendente dall'utilizzo di mezzi meccanici (tasti ma senza martelli né corde). La novità introdotta per prima dalla musica elettronica poi dalla musica informatica, è la possibilità di produrre suoni senza vincoli meccanici; quindi è possibile con l'avvento dei sistemi in tempo reale proporre nuove relazioni "suono-gesto". Vedremo più avanti che tipi di interazioni ci possono essere, e in quali misure lo standard MIDI ci vincola o meno ad un unico tipo di relazione.

Gesti e suoni.

Tradizionalmente la prassi musicale si svolge in due fasi separate: la fase di composizione che porta alla definizione "qualitativa" della partitura (simbolica), e la fase di interpretazione di questa partitura che consente la conversione dei dati qualitativi in dati quantitativi da utilizzare sullo strumento produttore di suoni.

Con l'avvento dei sistemi audio-numeric tempo reale, è possibile ritrovare le due fasi del processo musicale, ma con nuove possibilità. L'esecuzione viene svolta fra più interpreti: un'interprete "sintetico" (il processore di suoni e il suo controllore) e gli eventuali interpreti "umani" che agiscono durante l'esecuzione su dispositivi che pilotano in funzione di una data programmazione dell'ambiente esecutivo il processore audio-numeric. In questa situazione è possibile dividere i compiti di esecuzione fra gli interpreti e instaurare vari tipi di relazioni fra gesto e suono. Le possibilità sono:

- 1) Un gesto definisce un parametro acustico; situazione che si adatta a poche situazioni musicali se si vuole definire gestualmente tutti i parametri, vista l'enorme quantità di dati che bisogna in genere fornire.

- 2) Un gesto definisce più parametri acustici; situazione più realistica che richiede l'inserzione di algoritmi di interpretazione e di conversione per fornire più dati utili a partire da un dato gestuale.
- 3) Un gesto controlla un processo di elaborazione di parametri; situazione di controllo gestuale a più alto livello che consente di pilotare uno o più processi realizzati dall'interprete "sintetico". Questa situazione ricorda il ruolo del direttore d'orchestra.

In questi tre casi l'elaborazione ha un ruolo "passivo", nel senso che è subordinato ad azioni gestuali: il gesto agisce in "master" sul suono. Esiste però la situazione rovescia/CADOZ e al. 1984/:

- 4) Il comportamento meccanico dei dispositivi gestuali dipende dal comportamento del modello meccanico di produzione sonora simulato nel processore audio-numerico; la relazione suono gesto è in questo caso bi-direzionale.
- 5) L'interprete "sintetico" è stato addestrato per l' "ascolto", e può reagire a seconda delle informazioni gestuali mandate dall'interprete "umano"; quest'ultimo caso mette quasi allo stesso livello i due interpreti; la macchina ha un ruolo "attivo"/Veroce 1984, 1985/.

Quest'elenco è molto schematico, e riguarda la programmazione dell'ambiente esecutivo previsto per l'esecuzione di un pezzo, quindi le interazioni previste fra i due tipi di interpreti. Possiamo distinguere due tipi di operazioni da effettuare sui dati gestuali: operazioni di interpretazione o conversione alla ricezione di questi dati e operazioni di sincronizzazione al momento della sintesi.

Informazione esplicita.

Un flusso di dati MIDI contiene delle informazioni esplicite ed implicite.

L'informazione implicita verrà notata (t_n , valore): il cambiamento di stato assunto ad un'istante t_n da un dispositivo gestuale. Data la velocità di 31,25 Kbaud, e una media di 3 bytes trasmessi (30 bits) per ogni cambiamento di stato, riceviamo un'informazione completa di circa 1ms. Questo tempo di trasmissione è molto importante perché fissa i limiti della sensazione di simultaneità e quindi del tempo reale. Con questa velocità, possiamo indicativamente pensare di simulare 20 eventi in parallelo, supponendo di non dover superare una soglia di percezione di simultaneità pari a 20 ms (ordine di grandezza dei tempi di attacco di suoni relativamente percussivi) e di non dover elaborare i dati ricevuti.

Nella pratica questo numero massimo si riduce notevolmente se si devono fare delle conversioni sui dati o gestioni particolari per sincronizzare più dati fra di loro.

Nella elaborazione tradizionale dei dati MIDI, i programmi sono ottimizzati per privilegiare il tempo reale e simulare il parallelismo. Ci sono due tipi di parallelismo:

- 1) Il parallelismo interno ad un canale MIDI, cioè la polifonia. In questo caso, il programma deve gestire una famiglia di algoritmi identici, cercando ogni volta che arriva una copia (t_n , valore) per il canale considerato, di assegnare i dati al primo algoritmo libero. I dati vengono convertiti nello stesso modo, la difficoltà della programmazione consiste nella risoluzione di casi critici come: polifonia richiesta / polifonia prevista.
- 2) Il parallelismo riguarda gli interessi: più canali MIDI contemporanei o più dispositivi in parallelo. In questo caso, i programmi devono collegare ad ogni canale MIDI un algoritmo di sintesi particolare con i suoi parametri di ingresso specifici. L'elaborazione effettua-

ta sulla copia (tn, valore) dipende quindi dall'algoritmo coinvolto, e conviene prevedere una serie di trasformazioni o conversioni dei dati molto efficienti (conversioni tramite lettura di tabella, riscalamenti semplici ecc.).

In entrambi i casi il parametro temporale indica l'inizio o la fine di un evento sonoro (singolo per quanto riguarda i dati in tempo reale e i dati comuni). Il tempo viene utilizzato come informazione di tipo "trigger" e nel migliore dei casi per sincronizzare più dati fra di loro (frequenza di una nota che riguardano ad esempio le definizioni di una portante e di una modulante, e l'inizio di uno inviluppo di ampiezza).

Informazione implicita.

Il MIDI contiene però altri tipi di informazione. Sono delle informazioni implicite di due tipi:

- 1) Informazione sullo svolgimento dinamico del gesto nel tempo. Disponiamo in questo caso di informazioni riguardanti le regole interpretative dell'esecutore significative da un punto di vista musicale (interpretazione del tempo, passaggi fra due eventi sonori o fraseggio, dinamica ecc.). Se esistono delle regole formalizzate nell'interprete sintetico che prendono in considerazione dati derivati dai dati MIDI stessi, e che li associano a comportamenti dinamici dei parametri acustici, allora sarà possibile rendere il controllo gestuale meno meccanico.
- 2) Informazione differenziale istantanea fra gesto effettivo e gesto previsto (ad esempio da partitura). In questo caso si tratta di paragonare, ad un dato istante, due eventi (l'evento avvenuto e l'evento previsto secondo la partitura) e di decidere in funzione del risultato varie strategie di controllo per l'evento sonoro considerato. L'esempio tipico è il caso della variazione dinamica del tempo metronomico (accelerandi e rallentandi) durante un accompagnamento in tempo reale da parte di un interprete sintetico (caso che verrà trattato più avanti).

L'informazione implicita contenuta nel MIDI può quindi essere sfruttata in un modo più "musicale", o almeno più vicino ai concetti di una vera interpretazione. Lo scopo non è quello di riproporre i modelli interpretativi tradizionali, bensì nuove regole adatte alla manipolazione in tempo reale degli algoritmi di sintesi che sono alla base della produzione sonora.

Lo standard MIDI, anche se limitato per la sua velocità di trasmissione dati e per la struttura interna dei dati, ci consente di inventare vari tipi di relazioni fra interpreti umani e sintetici. Il software di controllo deve essere più raffinato possiamo suddividerlo in tre parti:

- 1) Acquisizione dei dati MIDI.
- 2) Processi di conversione, di elaborazioni o di "interpretazioni" dei dati MIDI in dati acustici.
- 3) Processi di sincronizzazione per il controllo degli eventi sonori.

L'uso convenzionale del MIDI tende a ottimizzare il software ed a integrarlo sullo strumento di sintesi, questo implica dei processi di interpretazione e di sincronizzazione limitati. È però possibile, portando l'elaborazione dei dati MIDI all'esterno rispetto al processore di sintesi (su uno o più controllers) di potenziare i processi 2 e 3 e di proporre nuovi tipi di ambienti esecutivi. Questi ragionamenti sono stati alla base della definizione della stazione musicale MIDI-4X.

3. LA STAZIONE MUSICALE MIDI-4X.

La stazione musicale MIDI-4X è stata costituita attorno al nucleo PDP-11/4X come è stato descritto nell'articolo /RIZZO SAPIR 1988/. Come primo passo è stata realizzata una interfaccia MIDI per il PDP-11 che consente al sistema PDP-4X di ricevere dati MIDI. Poiché la scheda DS11A del PDP-11 permette di accedere al calcolatore attraverso una linea seriale (solitamente utilizzata per collegare un terminale video) con una velocità massima di 9600 baud, modificando il quartz della DS11A è stato possibile raggiungere i 31.25 Kbaud richiesti dal MIDI /SANI 1988/.

In un secondo momento questo sistema è stato inserito nella rete locale del C.S.C.; questo consente di usufruire delle risorse del centro (VAX, IBM-PC, ATARI ecc.) per sviluppare parte del software all'esterno del PDP-11, e scaricare di conseguenza i suoi compiti di controllo.

Queste ultime considerazioni, che suggeriscono come caso ideale un PDP avente esclusivamente la funzione di interfaccia tra un'elaborazione esterna dei dati ed il 4X, unite all'esigenza, ben soddisfatta dal MIDI, di sviluppare progetti di tipo modulare, hanno fatto nascere l'idea di inserire, tra dispositivi MIDI e 4X, un COMPUTER MIDI (nel nostro progetto abbiamo scelto un ATARI). A questo spetta il compito di ricevere i dati MIDI, elaborarli, comunicando col sistema 4X attraverso una linea che, è MIDI nel senso della velocità di trasmissione, ma non in quello del formato dei dati. Per sfruttare al meglio il processore 4X, si è pensato di moltiplicare il numero delle stazioni su cui effettuare tutta quella parte di programmazione che non interessa il tempo reale (e che necessita solo di una presenza simulata del 4X), lasciando all'accesso monoutente, attraverso il terminale del PDP, tutta la parte del real-time.

Il computer MIDI.

Il COMPUTER MIDI è caratterizzato da una struttura particolare in cui si possono individuare tre blocchi principali:

- 1) Blocco di acquisizione: è la parte che comunica direttamente con i dispositivi MIDI. Traduce i dati in formato MIDI in informazioni che siano interpretabili dal blocco seguente.
- 2) Blocco di interpretazione/elaborazione: a questo livello si possono effettuare due tipi di operazione. La prima consiste nel semplice trasferimento dei messaggi ricevuti e tradotti in un opportuno formato (interpretazione). La seconda aggiunge a tutto questo la possibilità di operare delle manipolazioni sui dati stessi (elaborazione) come vedremo più avanti nei paragrafi dedicati all'accompagnamento in tempo reale.
- 3) Blocco di invio dati: è l'interfaccia verso l'uscita audio, nel nostro caso il sistema 4X. Da questo blocco esce l'informazione esecutiva (parametri di sintesi) che viene ottenuta dall'informazione operativa (byte di stato + 2 bytes di dati) in base alle esigenze di esecuzione (conversione, sincronizzazione).

Comunicazione MIDI-PDP-4X.

La trasmissione di un messaggio MIDI completo richiede circa 1 ms. Il COMPUTER

MIDI ha quindi 1 ms di tempo per pre-elaborare il dato ricevuto e presentarlo al PDP nel formato stabilito. Il formato più semplice che si potesse pensare, per un uso del 4X come strumento MIDI standard, prevede l'invio del dato e dell'indirizzo dove il dato andrà scritto.

Per quanto riguarda la gestione del 4X, da parte del PDP, la tendenza è quella di considerare il calcolatore come un "slave", limitando i suoi compiti alla configurazione delle schede del processore, stabilendo così l'algoritmo di sistemi utilizzato, e l'aggiornamento in tempo reale dei dati ricevuti (coppie indirizzo-valore). Infine, poiché il problema più serio del MIDI sta nella lentezza dovuta alla trasmissione di tipo seriale, eventuali tentativi di ottimizzare vanno necessariamente indirizzati verso la possibilità di ricevere una maggior quantità di dati nell'unità di tempo, o aumentando la velocità di trasmissione (il cosiddetto SUPERMIDI), o lavorando con una interfaccia che preveda più canali MIDI in parallelo.

4. ESEMPIO APPLICATIVO: ACCOMPAGNAMENTO IN TEMPO REALE DI UN ASSOLO MIDI.

Esistono varie possibilità per utilizzare un'elaboratore come accompagnatore di un assolo. Se in particolare si vuole usufruire dell'elaboratore in rappresentazioni dal vivo per realizzare un accompagnamento real-time, si deve sostituire l'esecutore umano con un esecutore sintetico, che non sia dotato solo di un output di esecuzione, ma che, soprattutto, abbia la possibilità di modificare quest'ultimo in funzione di uno o più input (audio, visivi, audio-visivi).

Un modello di esecutore sintetico è caratterizzato da due fasi distinte: ascolto ed esecuzione. Nella prima fase avviene l'analisi dei dati di input, con l'estrazione di una serie di informazioni quali la posizione nella partitura, il metronomo, l'altezza del suono, etc.. Nella seconda fase si ha l'organizzazione e la sincronizzazione della performance in base ai riconoscimenti effettuati nella fase di ascolto ed alla partitura che deve essere eseguita. C'è inoltre la possibilità di avere una terza fase (apprendimento) in cui l'esecutore sintetico impara dall'esperienza di performance (un po' come farebbe l'esecutore umano), nell'eventualità di ritrovarsi, in futuro, a dover prendere delle decisioni in situazioni analoghe.

L'algoritmo di Roger B. Dannenberg.

L'accompagnamento in tempo reale permette di superare i problemi di sincronizzazione e di interpretazione che si avevano con il nastro registrato. Sorgono però altri problemi quali l'acquisizione dell'assolo, il suo confronto con una partitura di assolo previsto e la generazione delle informazioni temporali necessarie all'esecuzione dell'accompagnamento.

A differenza di un accompagnatore umano, il computer può rilevare una classe limitata di eventi dall'esecuzione di un assolo monofonico di tipo audio. Questo fatto impone che, sia l'assolo che la corrispondente partitura attesa possano essere modellati come sequenze di valori. La partitura sarà quindi costituita dalla lista di eventi che dovrebbero essere rilevati durante una corretta esecuzione dell'assolo.

L'algoritmo di Dannenberg permette di determinare la migliore corrispondenza tra assolo eseguito e partitura di assolo previsto, che significa: la più lunga sottosequenza comune alle due liste di eventi. Le uniche informazioni che l'algoritmo deve fornire alla parte di accompagnamento, la cui partitura è stata memorizzata in una fase preliminare, sono di carattere temporale. Si tratta cioè di segnalare quando ogni evento dell'accompagnamento deve

essere eseguito. Poiché nel progetto di Dannenberg l'accompagnamento non risponde ad informazioni che non siano di carattere temporale, quali ad esempio l'intensità, l'articolazione o altre sfumature interpretative, l'algoritmo ha, in questa direzione, un largo margine di miglioramento. In particolare, acquisendo un assolo di tipo MIDI, abbiamo la possibilità di prendere in considerazione tutta una serie di informazioni aggiuntive, rispetto a quelle temporali e di pitch solamente, grazie alla molteplicità di informazioni che il MIDI invia ad ogni nostro gesto musicale.

Generalizzazione dell'algoritmo di Roger B. Dannenberg.

Sulla base dell'algoritmo monofonico precedentemente descritto, Roger B. Dannenberg ha realizzato un programma che permette di verificare se l'accompagnamento è in grado di seguire effettivamente l'assolo, e quindi di testare la bontà dell'algoritmo stesso.

Tale programma è dotato di una struttura modulare che ne consente un facile adattamento al contesto di lavoro definito dalla stazione musicale MIDI-4X. Esso può essere spezzato in due parti distinte: la prima, riguardante le parti di acquisizione dell'assolo e di matching, da implementarsi sul Computer MIDI; la seconda, riguardante la parte di accompagnamento vera e propria, da realizzarsi per ora sul PDP, con l'intento di portare anche questo blocco all'esterno dell'elaboratore.

Codifica di una partitura MIDI.

La nostra ipotesi di lavoro stabilisce il tipo di codifica per una partitura MIDI prendendo lo spunto da un linguaggio a livello operativo tipo MUSIC V, con l'idea di realizzare poi un compilatore per partiture MIDI la cui uscita sia costituita da una sequenza di dati MIDI uguale a quella che si otterrebbe da un'esecuzione della partitura descritta. A questo punto, resi compatibili tra loro i due input dell'algoritmo di matching (e cioè l'assolo MIDI e la partitura MIDI dell'assolo previsto), si tratta di modificare l'algoritmo stesso affinché possa prendere in considerazione degli eventi sonori descritti con un livello di specificazione molto elevato, grazie al MIDI.

All'interno del COMPUTER MIDI la partitura dovrà essere spezzata in due parti che abbiamo indicato con blocco di "sintesi" e blocco "differenziale" (Fig. 1).

Questo perché i dati MIDI in ingresso si riferiranno a gesti che potranno assumere due diversi significati:

- 1) controllo in tempo reale della parte di sintesi;
- 2) input di un blocco interno al COMPUTER MIDI, dove verrà effettuato il confronto tra il dato ricevuto ed il dato previsto da partitura e da cui si potrà ricavare un'informazione di tipo differenziale (variazione del metronomo se l'evento previsto è in ritardo o in anticipo rispetto all'evento ricevuto).

Sul PDP sarà presente la cosiddetta partitura "acustica". Essa riguarderà, da un lato alcune voci controllate in tempo reale direttamente dai dispositivi gestuali esterni, dall'altro delle voci (di una partitura di accompagnamento nel nostro caso) che potranno subire delle modifiche, rispetto all'andamento prestabilito, in base alle informazioni differenziali, che il COMPUTER MIDI invierà al PDP.

Realizzazione della parte di accompagnamento.

Uno dei requisiti fondamentali richiesti ad un accompagnamento in tempo reale, è dato dalla possibilità di saltare avanti o indietro nella partitura che deve essere eseguita. Presso il C.S.C. è stato realizzato un programma (PAR4X) /ZORIO 1986/ che consente l'esecuzione, da parte del 4X, di partiture scritte in un linguaggio (NOT4I) simile al MUSIC V. L'attuale struttura dati utilizzata dal programma PAR4X consente però il solo avanzamento all'interno di una voce.

Considerando il caso di un accompagnamento monofonico, verifichiamo quali modifiche si sono dovute apportare alla struttura dati, per poter rendere bidirezionale la possibilità di movimento all'interno della voce.

Si è dovuta aggiungere, per ogni macroevento considerato, una nuova locazione di memoria contenente il numero di microeventi del macroevento precedente.

Per un accompagnamento di tipo polifonico invece, si doveva pensare a come collegare tra loro le varie voci, giacché lo spostamento in avanti o indietro all'interno di una partitura polifonica comporta, per ogni voce, modifiche di tipo diverso ma comunque interdipendenti. Si è pensato di attribuire ad una di queste, il compito di voce "master", e cioè di voce guida per tutte le altre (Fig. 2). In particolare, per essa è richiesta la bidirezionalità di movimento, essendo l'unica delle voci su cui il programma esercita un controllo diretto. In più, vi è la presenza, per ogni macroevento, di una nuova locazione di memoria il cui contenuto permetta di effettuare il "link" con la voce successiva. Questo registro aggiunto, contiene un puntatore al macroevento della voce successiva che deve essere ON in corrispondenza al macroevento corrente della voce master.

Le voci "comuni" sono dotate anch'esse, in ogni loro macroevento, di questo puntatore alla voce successiva, mentre non è richiesta la bidirezionalità di movimento. Infatti, un salto all'indietro nella partitura, che generalmente induce degli spostamenti a ritroso nelle varie voci (o perlomeno in alcune di esse), può esser totalmente controllato dalla voce master che produrrà, in queste voci comuni, uno spostamento all'indietro che potremmo definire "indiretto". Le voci comuni quindi, non sono propriamente dotate della possibilità di movimento a ritroso nella partitura, ma attraverso la voce master.

Un'ulteriore sviluppo per la voce master, visto che i suoi compiti sono essenzialmente di interfacciamento tra il programma e le altre voci, è quello di renderla "muta", assegnandole esclusivamente un ruolo di controllo. La sua struttura sarà più semplice di quella vista in Fig. 2, poiché si tratterà, in pratica, di una sequenza di puntatori ad opportuni macroeventi della voce successiva.

5. CONCLUSIONI.

L'ambiente per l'uso in tempo reale della stazione musicale MIDI-4X è ancora in fase di realizzazione. Il progetto è stato interamente definito, con le specifiche delle strutture dei dati ed i vari protocolli di comunicazione fra i blocchi che costituiscono la stazione musicale. La realizzazione segue le tre direzioni seguenti. La prima riguarda lo sviluppo del linguaggio di codifica di partiture MIDI che viene effettuato sia su VAX che su ATARI. La seconda riguarda l'implementazione sul computer MIDI (ATARI) di algoritmi interpretativi per l'uso non convenzionale dei dati MIDI. La terza riguarda la parte specifica all'output audio 4X; cioè la comunicazione ATARI-PDP durante l'esecuzione di partiture MIDI, e la conversione dei dati gestuali in dati esecutivi 4X.

Bibliografia

Bloch J., Dannenberg R. (1985) — *Real time accompaniment of keyboard performance*. Proc. of the 1985 International Computer Music Conference, 279-290.

Cadoz C., Luciani A., Florens J.L. (1984) — *Responsive input devices and sounds synthesis by simulation of instrumental mechanisms: the CORDIS system*. Comput. Music J. 8, 3, 60-73.

Dannenberg R.B. (1984) — *An on line algorithm for real time accompaniment*. Proc. of the 1984 International Computer Music Conference, 193-198.

Dannenberg R.B. (1986) — *The CMU MIDI Toolkit*. Proc. of the 1986 International Computer Music Conference, 53-56.

Martin E. (1987) — *Gestione di dati MIDI per l'accompagnamento in tempo reale con il processore 4X*. Tesi di Laurea, Univ. di Padova.

Rizzo M. (1987) — *Ambiente di sviluppo per processore di suoni 4X*. Tesi di Laurea, Univ. di Padova.

Rizzo M., Sapir S. (1988) — *Ambiente di sviluppo multiutente per il processore 4X*. Atti del 7° Colloquio di Informatica musicale, Roma.

Sani F. (1987) — *Interfaccia MIDI per PDP-11*. Rep. C.S.C., Univ. di Padova.

Sapir S. (1987) — *Processeurs audio-numériques temps réel et informatique musicale: mise en oeuvre d'un système d'exploitation et utilisations pour la recherche, la production et la pédagogie musicales*. Thèse de Doctorat d'Etat. Université d'Aix, Marseille II.

Vercoe B. (1984) — *The synthetic performer in the context of live performance*. Proc. of the 1984 International Computer Music Conference, 199-200.

Vercoe B., Puchette M. (1985) — *Synthetic rehearsal: training the synthetic performer*. Proc. of the 1985 International Computer Music Conference, 275-278.

Zorio Maulen G. (1986) — *Gestione del processore 4X per esecuzione in tempo reale di partiture*. Tesi di Laurea, Univ. di Padova.

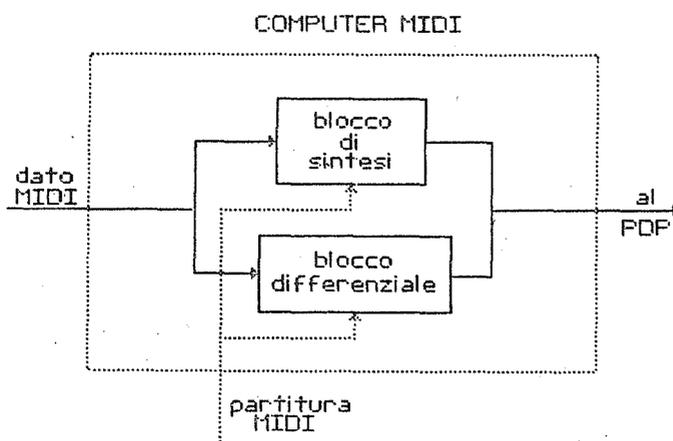


Fig. 1 - Blocco di sintesi e blocco differenziale nel COMPUTER MIDI.

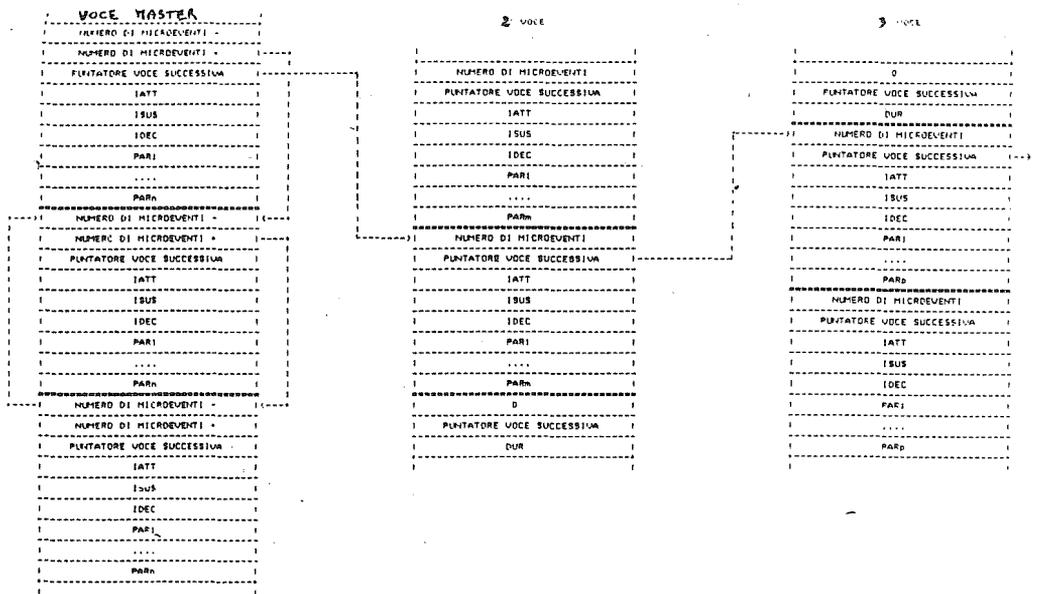


Fig. 2 - Accompagnamento di tipo polifonico mediante l'uso di una VOCE MASTER.

CAMPIONAMENTO DEL SEGNALE CON IL SISTEMA SOFT MACHINE

N. Bernardini, A. D'Agata, F. Galante, G. Nottoli, L. Spagnoletti
S.I.M. - Società per l'Informatica Musicale - Via Grazioli Lante, 30/76 - Roma

1. INTRODUZIONE.

Nel campo degli strumenti musicali digitali si è imposta di recente la tecnica di simulazione dei suoni detta "a campionamento". Detta tecnica consiste nell'acquisire eventi sonori esterni (= praticare una sorta di "registrazione digitale"), e riprodurre in seguito gli eventi acquisiti variandone la frequenza di campionamento (= la velocità di riproduzione).

Contemporaneamente allo sviluppo del circuito integrato *VLSI AG96-PCM*, dedicato alla generazione di suoni acquisiti mediante il metodo *PCM*, la Società d'Informatica Musicale (*SIM*) di Roma ha sviluppato una ricerca riguardante la simulazione di suoni acquisiti e la riproduzione a frequenza di campionamento fissa (1).

Lo strumento di questa ricerca è il sistema *Soft Machine*, un'elaboratore numerico di segnali audio microprogrammabile basato sul *DSP* integrato *Texas Instruments TMS32010* (2, 3).

2. FREQUENZA DI CAMPIONAMENTO IN RIPRODUZIONE VARIABILE O FISSA?

Una delle trasformazioni fondamentali operabile sui suoni campionati riguarda la manipolazione della frequenza del suono acquisito.

Ci sono, essenzialmente, due modi di modificare questa frequenza (4):

- a) variare il periodo generale della frequenza di campionamento, mantenendo costante il passo di lettura della tabella dei campioni (passo sempre uguale ad 1);
- b) variare il passo di lettura della tabella di campioni, mantenendo fissa la frequenza di campionamento.

Il metodo a) viene utilizzato in quasi tutti gli strumenti commerciali a campionamento attuali, mentre il metodo b) ha trovato sinora la sua utilizzazione in sistemi sperimentali, in particolar modo quelli in tempo differito (e.g. programmi quali *MUSIC V* ecc.).

Il successo in ambito industriale del metodo a) è da attribuirsi alla semplicità estrema di questa tecnica di simulazione, ed ad una (inizialmente) generosa approssimazione delle caratteristiche sonore (S) (6, 2).

Nelle applicazioni musicali sperimentali si è preferito in genere il metodo b) a causa delle configurazioni dei sistemi e della qualità superiore della simulazione (8).

Di recente, tuttavia, anche in ambito industriale l'interesse per il metodo b) è andato crescendo, grazie anche alla disponibilità a basso costo di tecnologie più sofisticate, nel tentativo di ovviare ai problemi posti dal metodo a) (vedi § 2.1) (9, 10).

2.1 Svantaggi della frequenza variabile di campionamento in riproduzione.

Dato che il campionamento a frequenza variabile è attuato modificando la frequenza

di conversione del *DAC (Digital-to-Analog Converter)*, questo metodo necessita di una catena di conversione completa (DAC + filtro anti-aliasing) per ogni "voce" dello strumento.

In questa catena, l'elemento critico è il filtro analogico anti-aliasing, il quale deve avere una frequenza di taglio pari a $F_s/2$ (dove F_s = Frequenza di campionamento) per eliminare le "immagini" di modulazione dello spettro del segnale. Questo filtro deve avere caratteristiche di risposta in frequenza ed in fase ottimali, pena una pesante distorsione del segnale riprodotto. Cosa, questa, difficile (se non impossibile) da ottenere con un filtro che debba "inseguire" una frequenza di taglio $F_s/2$ continuamente variabile.

Questo approccio, inoltre, rende poco pratica l'elaborazione digitale dei segnali poiché, nel caso polifonico, ciascun segnale riprodotto è completamente asincrono rispetto agli altri. Ogni elaborazione ha quindi bisogno di operatori distinti, eliminando così il vantaggio della virtualità inerente ai processi digitali, vantaggio che permette una utilizzazione più efficiente delle risorse.

2.2 Vantaggi della frequenza fissa di campionamento in riproduzione.

I difetti citati al § 2.1 vengono ovviati definendo una frequenza di campionamento fissa sia in acquisizione che in riproduzione.

In questo modo, infatti, è possibile costruire ed ottimizzare un filtro anti-aliasing con caratteristiche lineari e tempo-invarianti, semplificandone così notevolmente la realizzazione ed ottenendo risposte in frequenza ed in fase nettamente migliori.

Diventando inoltre possibili combinazioni di elaborazioni incrociate tra voce e voce, poiché queste si muovono ora sincronamente. Queste possibilità danno luogo a utilizzazioni più efficienti delle strutture di calcolo. Naturalmente, la conseguenza più evidente è l'indipendenza ottenuta tra il numero delle voci del sistema, (ora dipendente soltanto dalla potenza di calcolo di quest'ultimo) e il numero delle catene di conversione disponibili.

Un corollario di queste possibilità è l'omogeneità tra il campionamento ed altre tecniche di sintesi (*additiva, modulazione di frequenza, distorsione non-lineare, ecc.*). Questa omogeneità permette l'utilizzazione di segnali acquisiti per sintesi incrociate, ecc..

È naturalmente possibile diversificare la frequenza di campionamento in acquisizione con quella in riproduzione (e.g. *over* o *under-sampling*). Nel caso degli strumenti musicali a campionamento, tuttavia, l'identità tra le due frequenze presenta notevoli vantaggi per la sua semplicità e per l'efficienza delle elaborazioni. Tratteremo quindi d'ora in avanti con un'unica frequenza di campionamento identica in acquisizione ed in riproduzione.

3. ELABORAZIONI CON RIPRODUZIONE A CAMPIONAMENTO FISSO.

La ricerca svolta alla *SIM* si è concentrata su due elaborazioni essenziali:

- a) manipolazioni di frequenza;
- b) manipolazioni di tabelle.

3.1 Manipolazione di frequenza.

La manipolazione frequenziale di eventi sonori con frequenza di campionamento fissa

in riproduzione implica che la lettura delle forme d'onda acquisite e memorizzate venga eseguita con un passo di lettura (= incremento dell'indirizzo della tabella letta) variabile.

È evidente che una lettura a passo numericamente intero consente variazioni di frequenza troppo grandi per scopi musicali. Un passo siffatto permette esclusivamente la riproduzione a frequenze armoniche di quelle dell'evento campionato.

Si rende così necessaria una lettura con passo numericamente frazionario. La definizione frequenziale della riproduzione si ricava da:

$$f_{\text{delta}} = F_0 \cdot 2^{-n} \quad (1)$$

dove

f_{delta} = definizione frequenziale

F_0 = Frequenza nominale del segnale acustico

n = numero di bit della parte frazionaria

Dalla (1) si deduce che la definizione frequenziale dipende soltanto dalla frequenza nominale del segnale campionato e non dalla frequenza di campionamento né dalla lunghezza della tabella di forma d'onda.

Ammettendo l'identità tra frequenze di campionamento di acquisizione ed in riproduzione, la frequenza di riproduzione del segnale è data da:

$$f = s F_0 \quad (2)$$

2.2 Manipolazione di tabelle.

L'*editing* delle tabelle di memoria contenenti le forme d'onda acquisite diventa notevolmente più sofisticato con un passo di lettura delle tabelle con parte frazionaria.

Diventa infatti possibile una lettura a modulo frazionario: il *wrap-around* della tabella, cioè, può essere eseguito conservando la parte frazionaria. Questa caratteristica permette un andamento della fase del segnale assolutamente regolare.

La conseguenza più evidente è la possibilità di effettuare delle iterazioni su segmenti di tabella con indirizzi frazionari dei limiti. I risultati così ottenuti sono di qualità nettamente superiore a quelli raggiungibili dai sistemi oggi disponibili sul mercato.

3. Conclusioni.

Questa ricerca è stata ideata e messa a punto da Francesco Galante e Giorgio Nottoli. Arianna D'Agata ha scritto l'implementazione degli algoritmi nel microprogramma per la *Soft Machine*. Luca Spagnoletti ha realizzato gli esempi sonori.

Riferimenti Bibliografici

- 1). Nottoli, G., *Studio di realizzazione di strumenti a campionamento mediante il metodo PCM a frequenza di sampling fissa*, SIM - Rapporto interno, Roma 1988.
- 2). Lupone, M., *Lo Studio per l'Informatica Musicale di Roma / Il Sistema per la Sin-*

tesi in Tempo Reale, in Atti del 5° Colloquio di Informatica Musicale - Ancona, AIMI - Università di Ancona - Iselqui 1984 pp. 62-69.

3). Nottoli, G., Galante, F., *SOFT MACHINE: a real time fully programmable music system*, in Proceedings of the 1986 Computer Music Conference - Den Haag, Computer Music Association San Francisco, Ca. 1986.

4). Crochiere, R., E., Rabiner, L., A., *Multirate Digital Signal Processing*, Prentice-Hall, Englewood Cliffs N.J. 1983.

5). Massie, D., *The Emulator II Computer Music Environment*, in Proceedings of the 1985 Computer Music Conference - Vancouver, Computer Music Association, San Francisco, Ca. 1985.

6). Rossum, D., *Some Aspects of Sample Rate Conversion*, in Proceedings of the 1985 Computer Music Conference - Vancouver, Computer Music Association, San Francisco, Ca. 1985.

7). Bernardini, N., *Tecniche di Sintesi: la Sintesi per Campionamento*, in AUDIO Review n. 49 Aprile 1986, Technimedia Roma 1986.

8). Mathews, M.V., *The Technology of Computer Music*, MIT Press, Cambridge 1969.

9). Massie, D.C., Misesk, S.M., *Multitasking Operating System Design for Digital Music*, in Proceedings of the 5th AES International Conference - Music and Digital Technology, Audio Engineering Society Publications, New York 1987.

10). Freed, A., *MacMix: Recording, Mixing and Signal Processing on a Personal Computer*, in Proceedings of the 5th AES International Conference - Music and Digital Technology, Audio Engineering Society Publications, New York 1987.

A.S.F.: AN AUDIO SYNTHESIS FAMILY OF VLSI CHIPS

Giorgio Nottoli (S.I.M. Srl. Roma)
Francesco Galante (S.I.M. Srl. Roma)

Descrizione del progetto.

Nel presente progetto viene realizzata una famiglia di chip VLSI denominata A.S.F. (Audio Synthesis Family) che costituiranno un set generale e completo per la progettazione di strumenti musicali digitali.

Nella lunga concretizzazione del progetto A.S.F. si è passati dall'idea iniziale legata ad un sistema per un'unica tastiera, all'ipotesi di realizzare una famiglia di chip con utilizzo molto più vasto, di conseguenza, si sono dovute osservare regole di flessibilità superiori a quelle inizialmente previste.

Le prestazioni definitive di questa famiglia di chip sono le seguenti:

- 96 canali interconnettibili in vari modi.
- Numero di voci variabile, in quanto ogni voce può avere da 1 a 24 canali e diversi metodi di sintesi.
- Ciascuna voce, una volta completato il calcolo del singolo campione, deve poter essere inviata ad uno dei dodici canali d'uscita del sistema.
- Dodici uscite disponibili ciascuna con una dinamica di 16 bit (una dinamica globale di 115 dB).

Per raggiungere quest'obiettivo il lavoro prevede varie fasi: si passa dalla definizione dello scopo, in relazione alle prestazioni suddette, alla concretizzazione dell'architettura generale, quindi all'identificazione dei componenti, per giungere in ultimo alla progettazione e realizzazione VLSI.

Realizzazione del sistema definitivo.

Allo scopo di rispettare le caratteristiche sopracitate sono stati progettati e realizzati quattro chip che, in relazione alle convenzioni definite nell'architettura generale della A.S.F., assolvono ognuno una specifica funzione e possono essere inclusi in un ambiente ASF senza bisogno di ulteriore hardware esterno, se si escludono le memorie che dovranno contenere i campioni (tabelle di look-up). Un sistema completo prevede l'utilizzo di almeno 3 di questi chip, come è osservabile in figura (Fig. 1).

Attualmente già 4 chip sono nello stadio di Lay-out finale.

- FG (Function Generator) produce gli involuipi. Si possono creare 96 involuipi diversi, e questi vengono realizzati in combinazione fra FG e microprocessore di controllo: quest'ultimo invia i dati relativi al segmento successivo a quello in esecuzione. Il numero di segmenti in questo modo è slegato dalla generazione degli involuipi ed è interamente affidato al processore di controllo, lasciando così la scelta del numero di segmenti libera da condizionamenti hardware.

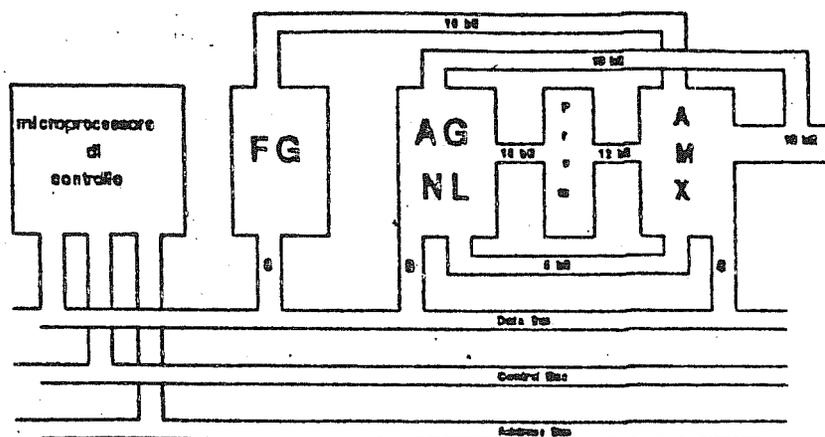


Figura 1. Schema a blocchi di un sistema di sintesi realizzato con i chip ASF.

— I chip AG (Address Generator) generano la fase degli oscillatori, l'indirizzo quindi, per l'accesso alle tabelle di memoria opportunamente configurate.

Esistono due versioni dell'AG che si possono scegliere alternativamente per applicazioni diverse: la prima si chiama AGNL (Non Linear Address Generator) orientata ai metodi di sintesi non lineari come la N.L.D. (distorsione non lineare) e la F.M. (modulazione di frequenza), mentre la seconda versione AGPCM (Pulse Code Modulation Address Generator) permette la realizzazione di campionamento e PCM classica.

AGNL genera fino a 96 oscillatori completamente indipendenti sia per quanto riguarda la frequenza, l'involuppo, che la modulazione in fase.

In pratica, ogni oscillatore può fare riferimento, in modo autonomo, ad una tabella contenuta in una memoria (RAM o ROM) esterna. AGNL può indirizzare fino a 128 tabelle senza bisogno di hardware esterno; questa caratteristica consente alla device di eseguire efficacemente sintesi lineari (forme d'onda fisse in sintesi additiva) e sintesi non lineari (N.L.D. e F.M.).

AGPCM è un generatore di indirizzo per 48 canali paralleli, orientato alla lettura di tabelle fisse con interpolazione per tecniche di sintesi PCM. Tra le caratteristiche più interessanti di questo chip spicca la possibilità di effettuare un looping frazionario del campione.

Da notare, per entrambi i chip AG, la particolare attenzione riposta nei modi d'inizializzazione della fase, un problema normalmente trascurato. Quando questo chip è connesso con FG, l'unità di controllo dell'inizializzazione della fase permette tre tipi di sincronismo tra le fasi di un gruppo di canali (voce di ASF): fase 0, fase determinata e copia di fase. In questo modo è possibile ottenere vari effetti quali la fase casuale, utile per simulare sezioni di strumenti oppure far partire tutti i canali di una voce con la fase attuale di una data voce presa come riferimento.

In conclusione ogni canale può essere programmato indipendentemente ed i canali possono essere sincronizzati, secondo la filosofia di raggruppamento della ASF, con il chip FG allo scopo di attribuire un involuppo ad ognuno di essi.

L'ultimo chip realizzato nel progetto si chiama AMX (Amplitude Modulation and Mixing) ed è dedicato a due funzioni principali: la prima conclude il "blocco oscillatore", invilupando la forma d'onda creata da AG; la seconda interconnette il canale in questione con eventuali altri canali sintetizzando un mixer a 96 ingressi e 12 uscite, in modo da formare l'algoritmo di sintesi desiderato.

Ad AMX è affidato inoltre il controllo del convertitore digitale/analogico, funzione svolta come un vero e proprio mixer con possibilità di "clipping" (saturazione elettronica di tipo analogico) per dare la possibilità all'utilizzatore di un controllo dinamico libero anche nella distorsione. AMX è stato progettato per lavorare in connessione con un microprocessore general-purpose che può variare in tempo reale sia l'attenuazione che le funzioni svolte da ciascun particolare canale. Nell'architettura generale di ASF, AMX viene connesso con FG, AGNL e AGPCM. Quando è collegato con il primo chip ogni canale è controllato dinamicamente con involuppi ad n segmenti in cui l'identificazione del segmento d'attacco permette di caratterizzare l'involuppo con una curva di tipo lineare, esponenziale o logaritmica.

Connettendo AMX con AGNL o AGPCM e con la memoria RAM o ROM (gestita da AG), è possibile effettuare l'interpolazione fra campioni successivi e questo consente di minimizzare lo spazio occupato dalle tabelle di look-up e di realizzare a pieno la sintesi PCM a campionamento fisso.

Oltre alle funzioni, già viste, di regolazione, controllo e mixing di un canale d'ingresso, AMX può eseguire un set di funzioni alternative che consentono di processare il segnale in ingresso utilizzando più di un ciclo (slot) di calcolo. In tal caso il numero di canali in ingresso è ridotto proporzionalmente. Tra queste funzioni alternative è importante far notare che AMX può sintetizzare anche filtri passa-alto e passa-basso del primo ordine, adatti a realizzare la risposta timbrica alla dinamica della tastiera, oppure il filtraggio dinamico durante il tempo di release in un algoritmo PCM con looping.

Nelle figure successive (Fig. 2 - Fig. 4) si possono osservare alcuni esempi di accoppiamento tra funzioni principali e funzioni alternative in più slot.

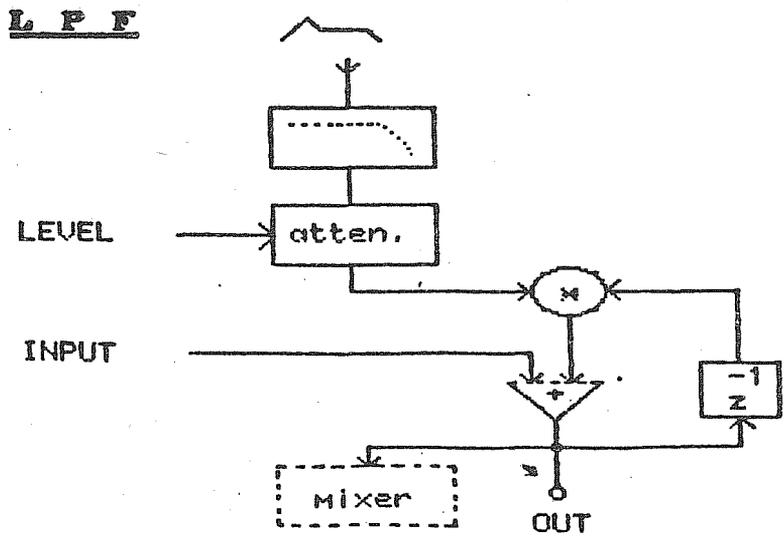


Figura 2. Filtro passa basso

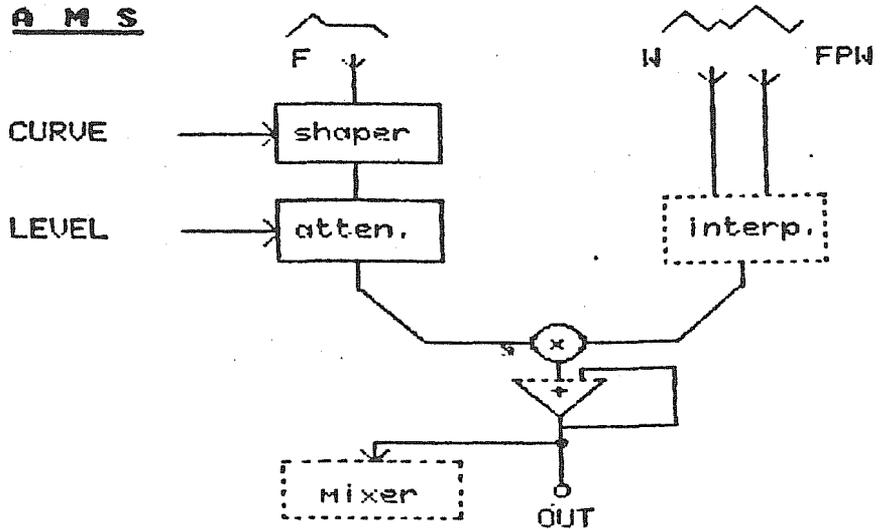


Figura 3. Involuppo del segnale in ingresso.

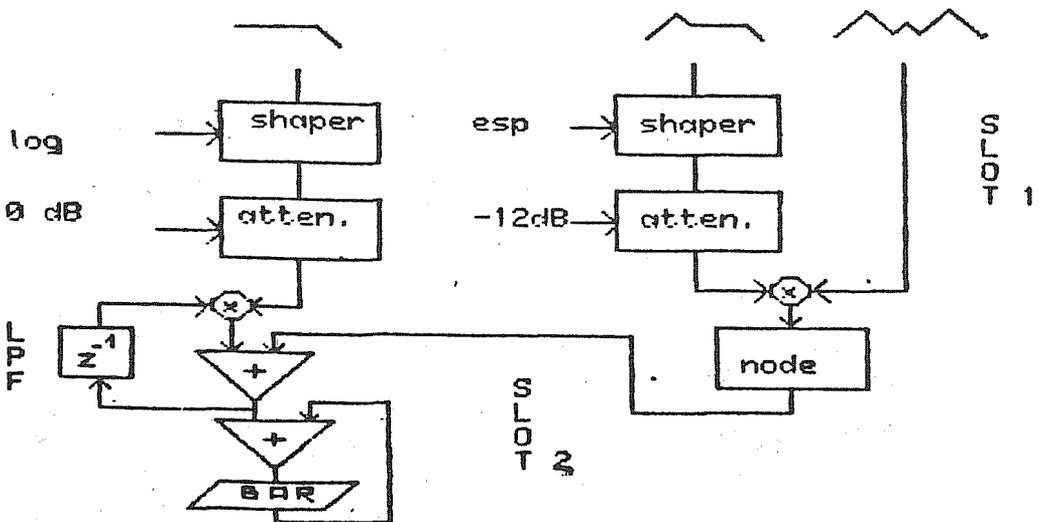


Figura 4. Schema a blocchi del processo AMX per la simulazione dell'unità analogica di involuppo e filtro passa basso.

Conclusioni

La realizzazione dei chip ASF, con le caratteristiche sopra descritte, lascia presagire strumenti di potenza finora sconosciuta in campo musicale commerciale. A tale proposito, va ricordato che una futura espansione del progetto prevede la realizzazione di almeno altri due chip:

- DFB che dovrà sintetizzare un banco di filtri digitali;
- REV che costituirà un'unità di riverberazione e chorus/flanger questi, assieme ai quattro chip già esistenti costituiranno una famiglia di componenti estremamente flessibili e dalle molteplici possibilità di connessione.

È importante notare che la strategia di progettazione di ASF deriva dalla versatilità degli strumenti della Computer Music e permette quindi di utilizzare questi chip in strumenti commerciali anche radicalmente differenti e che potranno coprire diverse fasce di mercato.

Con essi sarà possibile realizzare sia piccoli strumenti compatti ed economici, sia sofisticati sistemi "aperti" atti a soddisfare le esigenze e le problematiche legate alla composizione di Computer Music. In quest'ottica va sottolineato che tale sistema, implementato con opportune interfacce grafiche, potrà divenire un utile sostegno alla didattica musicale sia per quanto riguarda l'aspetto compositivo, sia per la sintesi e l'analisi del suono.

SESSIONE SOFTWARE

INTERAZIONI MUSICA - MOVIMENTO: UN SISTEMA PER LA GENERAZIONE DI PARTITURE DA ANIMAZIONI 3D

Camurri A., Canepa C., Orlich F., Zaccaria R.
DIST - Università di Genova, Italia

1. Introduzione

Riteniamo che il settore riguardante le interazioni tra musica ed altre forme artistiche ricopra un ruolo fondamentale nell'ambito della Informatica Musicale.

In particolare, il presente articolo affronta il problema dell'interazione tra musica ed arte visuale, o, più semplicemente, *movimento*, inteso in senso oggettivo come animazione di entità, oggetti, attori in uno spazio tridimensionale. Viene descritto un ambiente software per lo studio di tale interazione, in grado di gestire il processo di rappresentazione ed esecuzione di musica in funzione di eventi visuali, descritti all'interno di sistemi per la generazione di animazioni tridimensionali.

È noto l'intimo legame tra musica e movimento: anche dal punto di vista puramente fisico, è dal movimento — dell'esecutore così come della fonte sonora — che nasce il suono; da un punto di vista psico-percettivo esiste una induzione spontanea al movimento determinata dalla presenza di suono (13). Anche dal punto di vista della notazione musicale, esiste sin dall'antichità una stretta correlazione con il movimento. Ad esempio, la notazione chironomica riflette il movimento, la gestualità della mano per descrivere gli andamenti melodici, la rapidità di mutamenti ritmici in frasi musicali nell'ambito della musica antica. In generale, la terminologia e la notazione musicale — tradizionale e non — si poggiano su metafore basate sul movimento (1) (indicazioni agogiche). Anche recentemente è stata sviluppata dal punto di vista teorico la relazione tra arte — e la musica in particolare — e movimento (2) (3). Il punto di partenza è che, comunque, musica e movimento si pongono come modi di scolpire, dare forma al tempo mediante suoni od immagini. Zuckerkandl afferma che "... musica è movimento — movimento tonale è melodia — movimento di accordi è armonia..." (4).

Numerosi esempi possono essere citati sul rapporto tra musica e movimento nel contesto della danza. Un recente lavoro a questo proposito riguarda un esperimento di "automazione teatrale" (5), in cui il movimento di attori sulla scena influisce sulla generazione del commento musicale. Altre relazioni tra musica e movimento si avrebbero in una sorta di *partitura animata* in cui al movimento nello spazio di un singolo oggetto venga associato un parametro musicale, oppure in una esecuzione di processi musicali che abbia come conseguenza particolari movimenti in un dato contesto spaziale. Infine, oggetti musicali possono essere sincronizzati su determinati movimenti: caso tipico la sonorizzazione di un film.

Whitney introduce il concetto di *aural / visual composition* (6), sottolineandone l'analogia con una composizione per duo strumentale: il movimento di immagini nello spazio entra a far parte della espressione artistica musicale, con delle proprie regole di "composizione visuale" in analogia con quelle musicali (armonia, consonanza e dissonanza visuale). In questo contesto, la correlazione tra musica ed animazione non implica necessariamente una subordinazione dell'una rispetto all'altra.

La composizione di arte aurale/visuale sarà indubbiamente favorita dall'attuale disponibilità di strumenti tecnologicamente avanzati come videodischi ed elaboratori ad elevate prestazioni.

Su queste basi è iniziato il presente lavoro, con l'obiettivo di creare un ambiente software con spiccate caratteristiche di generalità applicativa e di apertura.

2. Descrizione del Sistema

Il sistema MANI (Music and ANimation Interface) è costituito da una collezione di programmi atti a gestire il processo di rappresentazione e generazione di musica correlata con scene in movimento.

Prima di entrare nei dettagli del sistema, vogliamo evitare l'equivoco che MANI sia un generatore automatico di colonne sonore. MANI non si sostituisce il compositore, bensì intende porsi come *strumento* di espressione per allargarne l'orizzonte creativo.

Il sistema si articola in tre fasi:

- 1) rappresentazione del movimento,
- 2) estrazione di informazioni,
- 3) interpretazione musicale di tali informazioni.

Nell'ambito di ciascuna fase sono disponibili diverse opzioni che garantiscono un'ampia possibilità di intervento.

3. Rappresentazione del Movimento.

In questa prima fase sono utilizzati diversi programmi di modellamento geometrico e di gestione e generazione di animazioni. La compatibilità tra essi è garantita da opportune interfacce software che producono una rappresentazione standardizzata, sotto forma di database, come input per la fase 2. Descriviamoli brevemente.

Nem (the NEw Man) è un linguaggio di programmazione sviluppato precedentemente (7), che consente di definire oggetti nello spazio tridimensionale e gestirne il movimento. La sua connotazione originaria è quella di linguaggio avanzato di simulazione nel settore della Robotica. Nem produce, oltre all'uscita grafica, una rappresentazione simbolica della animazione, detta tale.

VU3D consente di descrivere uno scenario (statico) tridimensionale e di osservarlo tramite una *telecamera virtuale* che si muove all'interno di esso. I movimenti di camera (traslazioni, rotazioni, zoom) sono controllabili in modo interattivo, da tastiera alfanumerica o da mouse, o in modo *batch*, attraverso un file di comandi. Il programma produce, oltre all'output grafico in tempo reale, un file di traccia del percorso della telecamera e un file di playback per la visualizzazione dell'animazione a cadenza costante.

Inbetween è un programma in grado di produrre un'animazione bidimensionale effettuando un'interpolazione fra due o più *immagini chiave*, disegnate sullo schermo con l'ausilio del mouse. È uno strumento molto veloce per ottenere immagini in movimento in tempo reale.

Paint è un package grafico per la produzione di animazioni bidimensionali attraverso il disegno dei singoli fotogrammi. La sua caratteristica fondamentale è quella di permettere la creazione di primitive grafiche definibili da utente a livello strutturale, ossia la gestione

di *forme*. Allo scopo di creare su schermo, fotogramma per fotogramma, l'animazione desiderata, si hanno a disposizione funzioni per la definizione e composizione di forme (elementari e non), mediante triangoli, poligonali, cerchi, splines, texttes, ecc., ed una serie di facilities di editing grafico estese anche alla gestione globale dell'animazione.

4. Estrazioni di informazioni

La seconda fase si realizza attraverso delle primitive di monitoraggio, che rappresentano sensori virtuali utili ad operare un filtraggio, che può essere anche tempo-variante, sugli attributi spaziali degli oggetti costituenti l'animazione. In pratica, l'utente scrive un programma col quale specifica quante grandezze intende osservare dal processo animazione, e fornisce, per ciascuna di esse, l'espressione che la definisce. Nella scrittura di queste espressioni si potrà fare riferimento alle primitive di monitoraggio predefinite o ad altre funzioni definite dall'utente. Le primitive possono riferirsi sia a caratteristiche a basso livello dell'animazione (di tipo puntuale: distanze, velocità, ecc...) che ad alto livello di astrazione (forme).

La sintassi utilizzata è pressoché uguale a quella del linguaggio di programmazione C.

L'esecuzione del programma di monitoraggio (il quale deve essere compilato con cmon, il compilatore del monitor) produce, per ciascuna grandezza che si è definita, una sequenza di lunghezza pari al numero di fotogrammi che compongono l'animazione.

Utilizzando nel programma di monitoraggio la variabile riservata *time*, che identifica il numero d'ordine del fotogramma corrente, è possibile realizzare il già citato filtraggio tempo-variante.

Inoltre, ad ogni istante è possibile fare riferimento non solo al fotogramma corrente, ma anche a quelli passati, sia per quanto riguarda il database di input, sia le stesse grandezze in corso di monitoraggio.

Le uscite numeriche del programma di monitoraggio possono, prima di essere utilizzate dai programmi della fase 3, venire normalizzate nel campo di valori fra 0 e 1 con il programma *normal*, il quale consente anche di definire *gruppi* di grandezze per conservarne le relazioni di disuguaglianza (fatto non garantito qualora ciascuna sequenza fosse normalizzata in modo indipendente dalle altre).

5. Interpretazione delle Informazioni

I dati ricavati dalle animazioni vengono in questa fase utilizzati a livello di partitura, associandoli a parametri musicali. Anche in questa fase dell'elaborazione è possibile operare a diversi livelli di astrazione: sia al livello di *nota*, intervenendo per esempio sui parametri di frequenza, ampiezza, inviluppo, ecc., sia a livello di struttura musicale.

Expand è il programma che, a questo punto, esegue trasformazioni su un testo musicale dato, utilizzando le grandezze precedentemente osservate nella animazione. *Expand* sostituisce nel testo tutte le occorrenze di *out(i)*, dove l'indice individua la *i*-esima componente delle uscite. Per utilizzare i valori riferiti ad un particolare fotogramma, sono disponibili dei comandi di posizionamento temporale, che consentono posizionamenti assoluti ($\llcorner 5$ per portarsi al fotogramma numero 5) o relativi ($\llcorner 10$ per avanzare di 10 fotogrammi). Anche l'identificazione *time* dà luogo a sostituzione, con il numero d'ordine del fotogramma corrente. *Expand* permette anche l'uso di costrutti iterativi e condizionali (*for*, *if*).

Utilizzando quindi un linguaggio per la descrizione di partiture musicali per scrivere il testo da espandere, è possibile realizzare un'interazione fra l'animazione che ha prodotto le grandezze osservate e lo *score* ottenuto dall'espansione.

In particolare, sono stati utilizzati *cmusic*, linguaggio di sintesi sonora sviluppato al CARL-UCSD (8) e *adagio* [CMU Midi Toolkit (9)].

Un'ulteriore possibilità fornita da *expand* è quella di interfacciarsi con il sistema *Petrex*, un interprete in tempo reale di reti di Petri (10) (11). Le reti di Petri si prestano alla descrizione di processi concorrenti e quindi di fenomeni formati da un certo numero di oggetti musicali (12). Con *expand* è possibile generare una marcatura dinamica dei posti di una rete di Petri musicale eseguibile da *Petrex*, ed intervenire quindi anche in questo modo sulla struttura dello *score* prodotto.

Un altro aspetto riguardante l'interazione immagini/musica che abbiamo voluto trattare è quello della *sincronizzazione*. Anche in questo caso si è proceduto definendo un supporto linguistico adatto al problema. Con esso si dichiarano i processi (musicali, ma non necessariamente) suscettibili di sincronizzazione e si definiscono alcuni parametri significativi dell'animazione (ad esempio la durata di un fotogramma). Inoltre occorre introdurre delle *clause* per specificare quali siano le azioni da intraprendere in corrispondenza di appropriate precondizioni. Nelle precondizioni si può fare uso di espressioni contenenti riferimenti alle uscite osservate, così come di funzioni che riportino lo stato dei processi (attivo, quiescente, appena__terminato, ecc.); le azioni riguardano l'attivazione o disattivazione di processi.

Un programma di sincronizzazione di questo tipo può anche prescindere dall'animazione, realizzando quindi una *sincronizzazione interna*. In questo approccio, a differenza da quello basato sulle reti di Petri, si pongono dei vincoli non solo sull'istante di attivazione di un processo, ma anche sulla sua durata: esso sarà quindi compresso o dilatato per soddisfare questi ulteriori vincoli temporali.

6. Note sulla implementazione

Esistono attualmente due versioni del sistema MANI: in ambiente Unix™ * ed in ambiente MS-DOS™ *. In entrambe il software è scritto in linguaggio C portabile, ad esclusione dei moduli grafici.

* Unix è un marchio registrato della Bell Labs. MS-DOS è un marchio registrato della Microsoft.

7. Alcuni esempi

Come primo esempio, consideriamo una animazione wire-frame raffigurante la stilizzazione di una figura umana, i cui elementi fondamentali sono le articolazioni. Un possibile programma di monitoraggio è il seguente:

```
N = 5;          /* numero grandezze osservate */
float a, b, c;
```

```
/* definizione del baricentro: le primitive di tipo __center calcolano le
componenti (x,y,z) del punto medio di un insieme di punti dati */
```

```

a = x_center("head left__hand left__foot right__hand right__foot",0);
b = y_center("head left__hand left__foot right__hand right__foot",0);
c = z_center("head left__hand left__foot right__hand right__foot",0);
define("baricentro", a, b, c); /* denominazione del punto */

/* quota della mano destra all'istante / fotogramma corrente */
out(0) = z("right__hand",0);
/* quota della mano destra all'istante / fotogramma precedente */
out(1) = z("right__hand", -1);
/* distanza tra mano sinistra e piede destro */
out(2) = dist("left__hand", "right__foot",0);
/* velocità del baricentro rispetto al sistema di riferimento */
out(3) = speed("baricentro",0);
/* distanza tra il baricentro ed il punto della figura ad esso più vicino */
out(4) = dist("baricentro",nearest("baricentro",0),0);
/* quantità proporzionale alla occupazione spaziale della figura */
out(5) = energy("head left__hand left__foot right__hand right__foot",0);

```

L'ultimo argomento di ogni funzione di monitoraggio specifica l'offset temporale espresso in numero di fotogrammi.

Le uscite *out(i)* possono essere utilizzate all'interno di una partitura: nell'esempio utilizziamo il linguaggio *cmusic*.

```

—> 5          /* inizializzazione al quinto fotogramma */
for i in 1 to 10
note time strumento1 out(4) out(0) out(1) out(2) out(3) out(5) p10;
>>          /* posizionamento al fotogramma successivo */
endfor

```

La associazione scelta in questo caso fa corrispondere ad ogni singolo fotogramma un suono, i cui parametri sono determinati dalle uscite osservate. In particolare, il costrutto iterativo *for* consente di scorrere i fotogrammi dell'animazione nell'ordine specificato; la variabile predefinita *time* assume il valore del fotogramma corrente, e nella interpretazione musicale definisce l'istante iniziale di emissione di ogni singola nota. La durata di ogni nota è stata arbitrariamente assegnata alla distanza tra il baricentro ed il punto ad esso più vicino (*out(4)*).

Riferimenti Bibliografici

- 1). Camurri A., Morasso P., Tagliasco V., Zaccaria R., *Dance and Movement Notation*, in "Human Movement Understanding", Morasso P. and Tagliasco Eds., North Holland, 1986, pp. 85-124.
- 2). Popper F., *Origins and Development of Kinetic Art*, New York: Little, Brown and Company, 1968.
- 3). Malina F.J., Ed., *Kinetic Art: Theory and Practice*, selections from the Journal Leonardo, New York, Dover, 1974.
- 4). Zuckerkandl V., *Sound and Symbol*, Music and the External World, Bollingen Series XLIV.1, New York, Pantheon, 1956, p. 109.
- 5). Cavaliere S., Papadia L., Parascandolo P., *From Computer Music to the Theater: The Realization of a Theatrical Automaton*, Computer Music Journal Vol. 6 (4), 1982, MIT Press.
- 6). Whitney J., *Digital Harmony*, Byte Books, McGraw-Hill, Peterborough, New Hampshire, 1980.
- 7). Marino G., Morasso P., Zaccaria R., *Motor Knowledge Representation*, Proc. IJCAI '85.
- 8). *CARL Startup Kit*, Computer Audio Research Laboratory, Center for Music Experiment, University of California, San Diego, 1985.
- 9). Dannenberg R.B., *The CMU Midi Toolkit*, Proc. ICMC 86, pp. 53-56.
- 10). Haus G., *Petrex: User and Reference Manuals*. Dip. Scienza dell'Informazione, Univ. di Milano, 1987.
- 11). Camurri A., *Petrex 2.0: Technical Notes*, Int. Rep.DIST, Univ. di Genova, 1987.
- 12). Camurri A., Haus G., Zaccaria R., *Describing and Performing Musical Processes by means of Petri Nets*, Interface, Swetz & Zeitlinger, Vol. 15 (1986), pp. 1-23.
- 13). Fraisse P., *Psicologia del ritmo*, Armando Ed., 1974.

RAPPRESENTAZIONE DELLA CONOSCENZA MUSICALE: SISTEMI ANALOGICI E SISTEMI LOGICO-SIMBOLICI

Antonio Camurri (*), Goffredo Haus (•) e Renato Zaccaria (*)
(*) DIST - Università di Genova, Via Opera Pia 11A, 16145 Genova.
(•) Dip. di Scienza dell'Informazione - Università di Milano
Via Moretto da Brescia 9, 20133 Milano

1. Introduzione

Questo articolo descrive un approccio per la rappresentazione della conoscenza musicale basato su reti semantiche di *frame* con eredità multipla e reti di Petri. Usiamo le reti semantiche per la rappresentazione della *informazione descrittiva* (1), le reti di Petri per la *informazione fattuale* su attività concorrenti e temporizzate. In particolare, usiamo un sistema basato su *frame* per rappresentare la conoscenza (ci riferiamo al guscio per lo sviluppo di sistemi esperti KEE (2) nella attuale implementazione prototipale) che concerne definizioni generali su processi temporizzati, sotto forma di reti di *frame*; rappresentazioni di processi individuali o attuali possono essere generate come *istanze* di una definizione generale, e reti di Petri sono prodotte automaticamente per descriverle.

Le informazioni memorizzate nella base di conoscenza sono la rappresentazione astratta simbolica in grado di generare automaticamente molte specifiche reti di Petri: denominiamo tale informazione simbolica *A—Net* (3). Le *A—Net* possono essere viste come una “super-classe” di reti di Petri ad un livello di astrazione superiore. A questo livello elaboriamo oggetti più astratti sui quali possiamo intervenire in due modi: derivando particolari oggetti individuali (reti di Petri) o ragionando attorno alle loro proprietà come in qualsiasi sistema di rappresentazione della conoscenza. Ad esempio, possiamo dire che il livello più alto consente di descrivere *azioni* di tipo generali quali “comporre una fuga” o “improvvisare su un tema”; il corrispondente livello di rete di Petri (il più basso) è, ad esempio, in questo caso “componi una fuga a 3 voci, in Re maggiore, utilizzando il soggetto X ed il controsggetto Y” o “improvvisa sul tema X...”.

Le *A—Net* possono essere applicate per la descrizione di una grande varietà di sistemi: questo lavoro è orientato verso la musica e l'intelligenza artificiale, campi che richiedono metodologie in grado di modellare la concorrenza, la temporizzazione, la sintesi ed in generale la programmazione automatica e/o guidata.

La musica, dal punto di vista della rappresentazione, può essere vista come un sistema cognitivo in cui l'aspetto temporale gioca un ruolo fondamentale. In generale, un fenomeno musicale può essere rappresentato mediante *attori* musicali interagenti e temporizzati. Il sistema descritto nell'articolo permette di rappresentare distintamente due livelli di rappresentazione: un livello di *ragionamento simbolico* (*A—Net*) ed un livello *analogico*, che racchiude gli aspetti temporali, di causalità e sincronizzazione tra particolari istanze di attori musicali (reti di Petri).

2. A—Net

Analizziamo le caratteristiche essenziali delle *A—Net*, che costituiscono lo strumento for-

male per la rappresentazione della conoscenza.

La base di conoscenza è strutturata nel modo seguente: una Long Term Memory (LTM) racchiude la conoscenza del mondo (musicale) in termini di entità chiamate *azioni* ed *attori*; la Working Memory (WM), contiene istanze di azioni per una sessione di lavoro.

Ogni entità definibile appartiene ad una delle due seguenti classi: la classe delle *ACTIONS* e la classe degli *ACTORS*. Questo implica una strutturazione della base di conoscenza che prevede due radici (frame) omonime delle due classi (vedi fig. 1), contenenti le proprietà generali che possono essere ereditate dalle entità.

Esistono differenze basilari tra le entità di tipo *ACTORS* e quelle di tipo *ACTIONS*. Un attore rappresenta una attività elementare (ad esempio una cellula tematica od un frammento ritmico) codificata in un singolo frame; una azione rappresenta una attività eseguita da più attori. Le azioni sono sottoalberi i cui frame radice sono i frame delle singole azioni, collegati ai frame/attori componenti. Un attore può comportarsi come sub-azione, caratteristica molto potente che consente di ottenere diverse rappresentazioni in corrispondenza di livelli differenti di astrazione.

Nel seguito faremo riferimento a terminologia riguardante reti di Petri, come definito in (4) e (5) e (6).

La classe *ACTIONS* è costituita dalle seguenti *slot* (strutture dati per la rappresentazione di proprietà di frame), ereditate da ogni azione:

INIT_MESSAGES	messaggi per inizializzare l'azione: essi vengono associati alla transizione iniziale (<i>start transition</i>).
END_MESSAGES	messaggi attivati alla fine della azione.
MULTI_MESSAGES	lista delle transizioni con più di un posto/attore, per indicare i posti connessi in ingresso alla medesima transizione.
LIST_ACTORS	una slot per ogni attore appartenente all'azione; ogni slot contiene i messaggi associati allo specifico attore, tipicamente le connessioni (archi).
PN_CODE	codice Lisp che implementa la rete di Petri di questa particolare azione.
ACTION_DATA	lista dei parametri dell'azione.

La classe *ACTORS* è costituita dalle seguenti slot, ereditabili da ogni attore in una base di conoscenza:

BEHAVIOUR	(valori: actor/action) un frame attore si comporta da attore o da azione a seconda dei livelli di dettaglio.
ACTOR_DATA	lista dei parametri dell'attore
ACTOR_CODE	attività dell'attore
ACTOR_TIME	durata temporale di ACTOR_CODE: se non è noto a priori, ACTOR_CODE viene calcolato direttamente durante la sua esecuzione.
GLOBAL_STATE	dati accessibili da altri attori
LOCAL_STATE	dati intermedi generati ed utilizzati da ACTOR_CODE.

Si noti che gli attori non contengono alcun tipo di messaggio: una azione contiene tutti i messaggi scambiati tra i suoi attori.

La creazione e la gestione della base di conoscenza sono realizzate mediante il *Music Task Planner*, che ha i seguenti compiti principali:

1. Creare istanze di azioni,
2. Creare il codice lisp della rete di Petri la cui esecuzione produrrà l'azione,
3. Eseguire il codice dell'azione (la rete di Petri).

Il primo compito consiste semplicemente nel generare istanze di azioni (sotto-alberi di frame) dalla LTM alla WM, avendo cura di cambiare i nomi degli attori coinvolti. Ciò equivale, ad esempio, alla creazione di una A—Net per un particolare brano musicale, a partire da una descrizione formale in LTM, in termini degli oggetti musicali (attori) costituenti il brano. La creazione del codice della rete di Petri consente di passare dalla meta-descrizione (A—Net) ad una descrizione operativa corrispondente (Rete di Petri). A livello di reti di Petri, gli attori corrispondono a posti mentre le transizioni sono gli elementi di sincronizzazione tra gli attori. Nel seguito viene analizzato un algoritmo in grado di creare istanze di reti, a partire da una base di conoscenza. L'obiettivo è generare il codice Lisp corrispondente alla rete di Petri dell'azione ed assegnarlo allo slot PN_CODE; il codice della rete verrà quindi eseguito, causando la performance della azione, ovvero del particolare brano musicale.

Da notare che i nomi in grassetto significano *slot della azione* (ad esempio LIST_ACTORS).

PN—CREATION Algorithm

1. Generation of the initialization code (variables initialization):

- 1.1 world_time = 0
- 1.2 start_place = 0 (marked at time 0)
- 1.3 places_list = (LIST_ACTORS) +
(start_place end_place deadlock_place),
list of the places/actors involved in the action:
if exists a place *in* (LIST_ACTORS)
whose BEHAVIOUR is equal to action,
then it will be recursively expanded with its
sub-actors into places_list.
- 1.4 *For Each* place *in* places_list - (start_place) *do*
place = -1 (not marked initially)

2. Generation of the loop code (net execution):

- 2.1 generation of the start_transition and its net connections:
input place is start_place, the output places are the
actors to be activated initially, contained in INIT_MESSAGES.
- 2.2 generation of transitions code:
 - 2.2.1 *For Each* element in the list MULTI_MESSAGE *do*
generation of the transition code (condition,
consume input tokens, produce output tokens).

2.2.2 For Each element in place_list do
generation of the transition code (condition,
consume input tokens, produce output tokens).

2.3 generation of the DO_AGAIN code:

If has been processed a place of type signal or counter
then has a null temporale duration
else world_time = world_time + 1 (next time slot)

3. Generation of the termination code: END or DEADLOCK management.

4. End.

DO_AGAIN è una variabile utilizzata come flag, che viene posto al valore *true* se viene processato un posto che non consuma tempo (posto di sincronizzazione). Ciò causa una ulteriore esecuzione del codice iterato (punto 2) senza incrementare il world_time.

L'algoritmo di esecuzione delle reti è ispirato all'algoritmo che abbiamo proposto per modellare processi musicali usando reti di Petri (5).

3. Un Esempio Musicale

Discutiamo ora un possibile modo di rappresentare la conoscenza nel caso di una fuga a tre voci. In fig. 1 è descritta a livello di A—Net parte di una possibile struttura di una fuga. L'azione *fugue* sarà composta da tre attori: *exposition*, *development* e *conclusion*. Se non necessitiamo di entrare ulteriormente nel dettaglio della rappresentazione, possiamo utilizzare solamente questi tre attori — ed in particolare il loro codice, ovvero la loro descrizione in termini notazionali — per la generazione di una rete di Petri in grado di eseguire una "istanza" di tale brano. A tale livello il brano sarà descritto — e quindi eseguito — in una forma

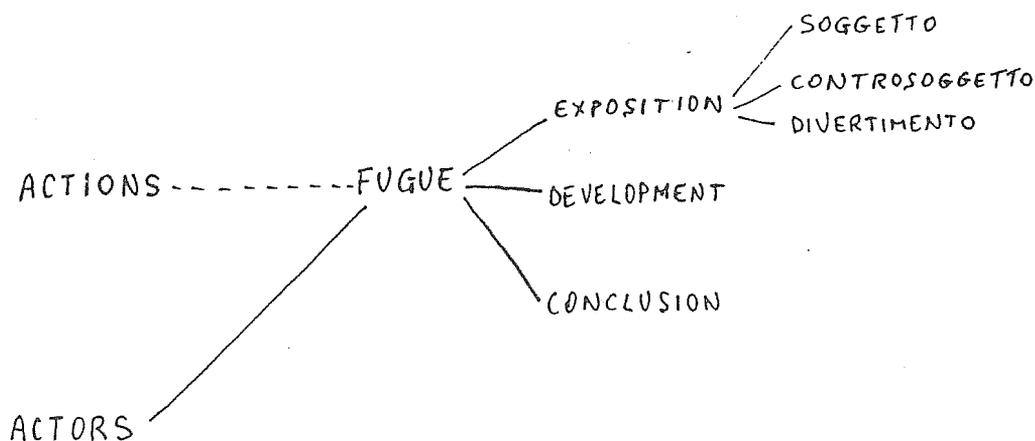


Figura 1. Livello di A—Net: Un possibile Fugue Knowledge Base (frammento)

approssimativa, poco dettagliata: non saranno evidenziate nella rappresentazione caratteristiche tematiche, contrappunti tra voci. In questo caso, la rappresentazione del brano si riduce ad una forma del tipo canzone tripartita, la cui prima parte corrisponde alla rete di Petri di fig. 2. Tuttavia, un attore può comportarsi come sub-azione, consentendo di ottenere diverse rappresentazioni della stessa azione, in corrispondenza di differenti livelli di dettaglio. In questo caso, non verrà considerata la rappresentazione contenuta in quell'attore, ma al suo posto entreranno in gioco gli attori costituenti tale sub-azione; in altre parole, un attore/posto di una rete viene espanso in una sottorete di Petri. In generale, nella rappresentazione di una azione musicale, il modello potrà usare diversi gradi di precisione e generalità per diversi contesti: potremmo volere rappresentare nei dettagli più fini l'esposizione, lasciando lo sviluppo — o parte di esso — in termini più approssimati. Pertanto, nell'esempio, è possibile considerare ad esempio *exposition* come sub-azione, composta dagli attori *soggetto*, *controsoggetto* e *divertimento*: a questo livello di dettaglio siamo in grado di rappresentare e generare l'esposizione della fuga in termini di soggetto, controsoggetto ed eventuali divertimenti (brevi episodi che possono procedere l'entrate del soggetto nelle diverse voci). In questo caso, la rete di Petri generata dal sistema è in fig. 3.

Analizziamo più in dettaglio il meccanismo di generazione della rete di Petri di fig. 3, a partire dal Fugue Knowledge Base di fig. 1. L'azione *fugue* conterrà innanzi tutto nella slot INIT_MESSAGES le informazioni per l'inizializzazione della rete da generare: il posto start, non essendo un attore, comparirà sia nella slot che nel codice della rete solamente come variabile da inizializzare ad 1 (posto marcato), così come CNT, che verrà inizializzato a 2 (contatore di iterazioni). INIT_MESSAGES conterrà inoltre le informazioni riguardanti le connessioni iniziali della rete, ovvero gli archi tra start, tr_start e Soggetto. La slot LIST_ACTORS della azione *fugue* sarà espansa dall'algoritmo nella lista degli attori che partecipano in questa particolare azione, includendo ricorsivamente anche gli attori componenti eventuali sub-azioni (nell'esempio, la slot BEHAVIOUR dell'attore *exposition* avrà valore *action* quindi sarà espansa nei suoi attori componenti). Esisterà inoltre, in *fugue*, una slot contenente la lista delle connessioni tra gli attori coinvolti nell'azione, che verranno utilizzate per la costruzione della rete di Petri.

Model Fugue - Net Exposition - Time Unit: 1 millisec

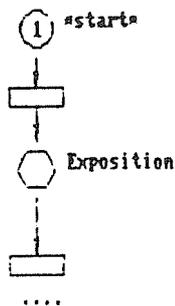


Figura 2. Una possibile rete di Petri ricavata dal frammento di Fugue Knowledge Base.

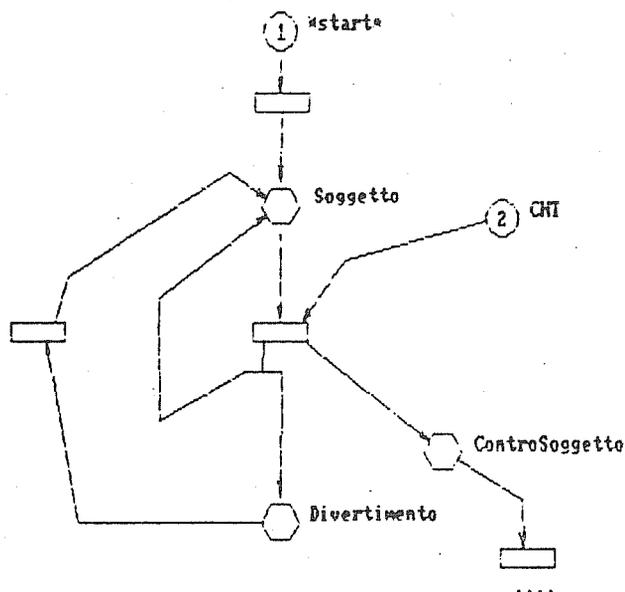


Figura 3. Una possibile rete di Petri, a maggiore dettaglio, ricavata dal frammento di Fugue Knowledge Base.

L'esempio discusso usa una rappresentazione con livello di dettaglio variabile tra i differenti attori (*exposition* è espanso in attori a maggior dettaglio rispetto *development* e *conclusion*). Se volessimo cambiare il livello di dettaglio occorrerebbe innanzi tutto definire compiutamente a livello di A—Net la base di conoscenza (ad esempio, definire l'attore *development* come sub-azione, ecc.); in tal caso sarebbe possibile generare diverse reti di Petri a partire dalla stessa base di conoscenza corrispondenti a qualsiasi livello di rappresentazione. Comunque, non è possibile, in generale, sostituire una sottorete di Petri con una "più dettagliata" senza dover cambiare la topologia generale. Ciò deriva dal fatto che, per esempio, l'espansione di un posto/attore richiede in generale nuovi archi che connettono il resto della rete. Quindi, *la riscrittura di una rete di Petri — per cambiare alcune rappresentazioni in sub-azioni — non può essere realizzata usando conoscenza locale, ma deve compiersi ad un più alto livello (di descrizione di rete)*. Questo fatto è una motivazione basilare che giustifica l'introduzione delle A—Net.

4. Applicazioni

La metodologia descritta è alla base di un sistema — *JAM* — per la rappresentazione di conoscenza musicale. In esso è stata introdotta conoscenza riguardante armonia funzionale, pattern melodici e ritmici, riguardante un ben determinato stile jazzistico: a questo proposito è stato intervistato il M.o F. Crivelli (intendiamo con il termine intervista il processo di acquisizione di conoscenza specifica). Il sistema è attualmente in grado di generare automati-

camente improvvisazioni su basi armoniche fissate, eventualmente interagendo con l'utente in tempo reale nella fase di improvvisazione. La base di conoscenza è strutturata in forma di A—Net, mentre particolari esecuzioni sono reti di Petri derivate con livelli variabili di automaticità. Verranno proposti esempi musicali generati dal sistema.

Il medesimo approccio è stato utilizzato per la rappresentazione e la generazione di *story-board*. Come esperimento, è stato analizzato e descritto in termini di A—Net un romanzo giallo di Agatha Christie: il sistema si è rivelato in grado di sintetizzare trame inedite, sulla base di quella descritta inizialmente in termini di azioni ed attori. Rimandiamo, per esigenze di brevità, ad altra sede la descrizione di queste ed altre applicazioni sviluppate.

Ringraziamenti

Ringraziamo particolarmente l'Ing. Poggi per il suo fondamentale contributo nella definizione delle A—Net. Inoltre siamo estremamente grati al M.o F. Crivelli per la pazienza e la collaborazione prestate in occasione del processo di acquisizione di conoscenza musicale nel sistema JAM. Un ringraziamento è dovuto anche agli studenti Giacomini e Ponassi, cui è dovuta l'implementazione della attuale versione del sistema JAM.

La presente ricerca è stata sviluppata parzialmente grazie al supporto finanziario del Ministero della Pubblica Istruzione (MPI 40% Informatica Musicale).

References

- 1). Brachman R.J., Fikes R.E., Levesque H.J., *Krypton: A Functional Approach to Knowledge Representation*, in Brachman, Levesque eds., *Readings in Knowledge Representation*, Morgan Kaufman Publishers, 1985.
- 2). KEE software Development System, Reference Manual 7th Ed., Intellicorp, 1985.
- 3). Camurri A., Vercelli G., Zaccaria R., *A—Nets: Structured Representation of Time and Actions using Petri Nets*, submitted to the Ninth European Workshop on Applications and Theory of Petri Nets, June 1988, Venezia.
- 4). Peterson J.L., *Petri Net Theory and the Modeling of Systems*, Prentice-Hall, 1981.
- 5). Degli Antoni G., Haus G., *Music and Causality*, Proceedings of the '82 International Computer Music Association Ed., San Francisco, 1982.
- 6). Camurri A., Haus G., Zaccaria R., *Describing and Performing Musical Processes by Means of Petri Nets*, Interface, Vol. 15 (1986), pp. 1-23.
- 7). Hale R., *Temporal Logic Programming*, in *Temporal Logics and their Applications*, A. Galton ed., Academic Press, 1987.

DEFINIZIONE DI SPECIFICHE PER IL COLLEGAMENTO FRA
TASTIERA MUSICALE E SISTEMI DI SINTESI DIGITALE DEL SUONO
VIA STANDARD MIDI: UN ESEMPIO APPLICATIVO

S. Soliani, M. Rubbazer
C.S.C. - Università di Padova - Via Gradenigo 6/A - PD

I musicisti che si occupano di musica all'elaboratore elettronico hanno a disposizione due classi di strumenti. La prima, tipicamente software, fa uso del solo calcolatore, perciò è rappresentata da una serie di programmi che accettano in ingresso una partitura di comandi, generano ed elaborano dei segnali in modo opportuno e forniscono in uscita una serie di campioni da passare al convertitore D/A.

Il limite di tali strumenti consiste nel fatto di non poter intervenire in tempo reale durante l'esecuzione.

Alla seconda classe, tipicamente hardware, appartengono i processori elettronici di segnale programmabili. La loro caratteristica è quella di permettere un elevato grado di interazione durante la generazione del suono, consentendo un utilizzo più conforme alla gestualità tipica dei musicisti.

Per contro, tali strumenti sono caratterizzati da una minore potenza e flessibilità, e inoltre richiedono una certa conoscenza dell'hardware della macchina.

Il lavoro svolto cerca di eliminare le limitazioni inerenti a queste due classi di strumenti coniugando l'elevata potenzialità offerta dai calcolatori con la praticità e la funzionalità dei processi elettronici programmabili.

Sfruttando le caratteristiche dello standard MIDI come protocollo di comunicazione, si è potuto definire il "computer MIDI" cioè un sistema formato da una tastiera musicale, un elaboratore personale e un sintetizzatore digitale di suoni.

Lo scopo che si vuole raggiungere è quello di eseguire una partitura, suddivisa in sezioni, con la possibilità di controllare la sintesi sonora in tempo reale, cioè il più velocemente possibile nei limiti della risposta del sistema. Col termine sezione si intende il periodo durante il quale l'algoritmo di sintesi, le funzioni memorizzate, e le modalità di controllo restano fisse. La sezione definisce pertanto il quadro di riferimento all'interno del quale si può intervenire con comandi gestuali durante l'esecuzione.

La partitura, invece, comprende sia la descrizione degli eventi sonori, sia la definizione dei controlli lasciati all'esecutore durante la sintesi.

Una partitura completa è perciò divisa in tre parti:

- 1) un riferimento strumentale relativo alla definizione dei parametri di ingresso per gli algoritmi di sintesi. A questo livello si definisce lo strumento computerizzato; si è ideata perciò una serie di routines che, attraverso scelte successive, guidano il musicista nella definizione di tutti i parametri propri del suono sintetizzato, cioè nella creazione di un vero e proprio strumento musicale.
- 2) Un riferimento gestuale che riguarda non solo il collegamento fra le tastiere musicale e numeriche ed i vari parametri ma anche il loro modo di controllo. Questa è stata la parte maggiormente sviluppata, in quanto si è cercato di ottenere una ottimizzazione ed una standardizzazione nella definizione e nel riconoscimento di un certo numero di "gesti/comandi musicali" in modo da facilitare la progettazione di pacchetti software da implementare su un qualsiasi computer.

- 3) La partitura musicale vera e propria. Quest'ultima parte è quella che riguarda l'esecuzione del brano musicale, perciò è stata presa in considerazione solamente come risultato sonoro del lavoro svolto.

A questo punto rimane da definire il computer MIDI prima di passare a descrivere l'applicazione di trasformazione della partitura gestuale in suono.

Il computer MIDI non è altro che un qualsiasi computer, corredato da una serie di routines, in grado di poter riconoscere i messaggi dello standard MIDI provenienti da diverse tastiere o dispositivi "MASTER", elaborarli a seconda delle partiture create, ricodificarli per uno o più sintetizzatori sonori di uscita.

Il tradizionale collegamento tastiera musicale/sintetizzatore sonoro può essere spezzato per poter inserire quello che è stato appena definito "computer MIDI". La figura 1 ne mostra la divisione nelle tre parti sopra menzionate.

La prima sezione, chiamata riconoscitore, è un decodificatore dello standard MIDI e svolge un lavoro di riconoscimento e decodifica dei comandi gestuali del musicista provenienti da tastiere musicali e dispositivi di input diversi.

La seconda sezione, chiamata programmazione, è quella dedicata alla creazione di strumenti computerizzati. Racchiude l'elaborazione vera e propria dei parametri degli algoritmi di sintesi.

La terza sezione, chiamata codificazione, trasforma le informazioni riguardanti l'evento sonoro in messaggi per i sintetizzatori a disposizione. Se questi ultimi riconoscono lo standard MIDI, il codificatore ritrasforma in codice MIDI il risultato dell'elaborazione, altrimenti lo converte in un protocollo di comunicazione adatto.

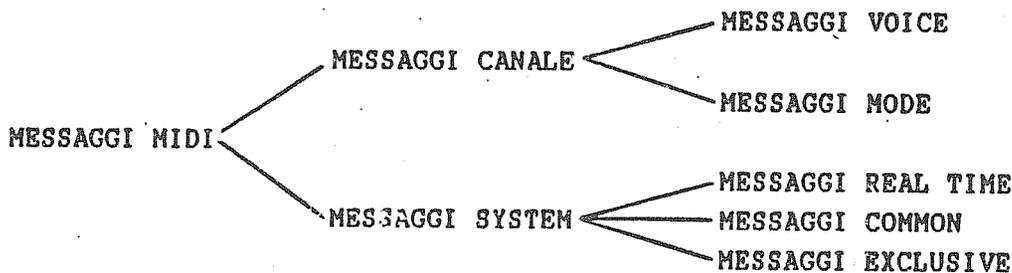
Esempio applicativo.

Si è voluto applicare quanto più sopra visto a un caso reale, pilotando un sintetizzatore digitale realizzato presso il C.S.C. dell'Università di Padova (Rubzazer, Santoiemma, Pata: Some advances in the development of a new architecture for a digital sound synthesizer - ICMC 86 Proceedings), per mezzo di un personal computer collegato a una Master Keyboard MIDI.

Lo scopo è quello di arrivare a progettare delle routines per il computer MIDI in modo da tradurre la gestualità del musicista in eventi sonori. Si cerca qui di seguito di dare un'idea dei problemi che si possono incontrare nella decodifica del protocollo MIDI, nell'intento di suggerire eventuali soluzioni attraverso la spiegazione dei diagrammi di flusso delle routines svolte. Seguendo un processo top-down si parte da quella che potrebbe essere la versione generale del programma e si scende poi, di volta in volta, all'interno dei vari blocchi, per chiarire meglio i problemi che possono nascere nei vari passaggi.

I messaggi MIDI

Per capire meglio i problemi da risolvere accenniamo brevemente a quali sono e come sono fatti i messaggi MIDI:



Lo specchietto mostra i due tipi principali di messaggi: i primi sono i messaggi CANALE ed i secondi sono i messaggi SISTEMA.

Poiché il protocollo MIDI prevede l'indirizzamento di 16 canali diversi, cioè di 16 "uscite sonore" diverse, questi messaggi indicano, istante per istante, quale delle uscite deve essere resa attiva nell'esecuzione di una partitura.

I messaggi SISTEMA non hanno niente a che fare con i canali, ma controllano l'intero sistema ed includono, ad esempio, quei messaggi che sincronizzano due o più strumenti oppure messaggi esclusivi di un particolare sintetizzatore.

I messaggi CANALE includono quelli VOCE e MODE.

I messaggi VOCE portano le informazioni relative alla partitura in esecuzione, come ad esempio i comandi di NOTA ON e NOTA OFF oppure quelli relativi al PITCH BENDER, mentre i messaggi MODE contengono le informazioni relative al MODO, in altre parole se l'esecuzione deve essere polifonica, monofonica o se l'accensione di una nuova nota deve comportare lo spegnimento della precedente oppure no. I messaggi SISTEMA includono quelli REAL TIME, COMMON, EXCLUSIVE.

I messaggi REAL TIME sono quelli necessari ai sequencer, alle batterie elettroniche o ad altri dispositivi MIDI per sincronizzarsi ed eseguire la partitura. I messaggi COMMON comprendono i messaggi di numero di misura, numero brano, richiesta di accordatura ecc..

I messaggi EXCLUSIVE sono quelli che servono a programmare ogni singola apparecchiatura MIDI "slave" tramite comandi specifici di ogni casa costruttrice.

I messaggi MIDI possono essere formati da 1, 2 o 3 bytes eccetto quelli EXCLUSIVE che hanno una parola di controllo come fine messaggio.

Il primo byte è sempre il byte di controllo ed ha il bit più significativo settato ad 1. I rimanenti bytes sono i bytes dati ed hanno il bit più significativo settato a 0.

I messaggi SISTEMA sono di solito formati da 1 solo byte, e quindi facilmente riconoscibili e trattabili.

Nell'esempio applicativo qui esaminato, tale byte è usato come indirizzo indiretto ad una tabella dove sono memorizzati gli indirizzi delle routines di servizio dei comandi stessi.

In questo modo viene effettuato il riconoscimento e il servizio del comando, tutto in una volta sola, con un salto indiretto alla routine selezionata.

I messaggi CANALE sono invece più difficili da trattare. Infatti nel byte di controllo risiede anche l'informazione relativa al canale attivato ed inoltre sono seguiti da altri due byte di dati.

Anche in questo caso il riconoscimento del byte di controllo è stato fatto mascherando,

in un primo tempo, il nibble meno significativo (contenente l'informazione relativa al canale attivo) e usando il byte risultante come indirizzo alla tabella degli indirizzi delle routines di servizio dei comandi.

Una volta selezionata la routine si recupera l'informazione riguardante il canale e la si smista fra le varie uscite. Intanto si prelevano i bytes dati ed a questo punto si possono o elaborarli mediante operazioni aritmetiche per ottenere effetti sonori particolari oppure usarli come indirizzi a tabelle dove sono stati preventivamente memorizzati i parametri necessari agli eventi sonori.

Questo secondo metodo è stato preferito al primo, perché di esecuzione molto più veloce. Vediamo come può essere strutturato un programma di massima attraverso i blocchi del flow-chart di figura.

Preparazione dei default.

Sono le routines di inizializzazione di tutti quei parametri e quei coefficienti necessari per la creazione dell'evento sonoro.

In questa sezione del programma si trovano le inizializzazioni che riguardano:

- i contatori: nel caso in oggetto sono i contatori di "scrittura" e di "lettura" della FIFO che contiene i dati MIDI in arrivo;
- i registri per i confronti: registri che verranno utilizzati come maschere o limitatori delle dimensioni di alcune tabelle;
- il registro "modo": si impone un certo modo di funzionamento in fase di preset da scegliere fra i modo MONO e POLY; durante l'esecuzione della partitura, questo "modo" potrà essere cambiato con un semplice comando, dato tramite la tastiera musicale;
- i registri di controllo di potenziometri: ogni tastiera musicale MIDI nella sua configurazione minima, comporta la "presenza" di almeno due potenziometri, di cui uno a zero centrale. Servono per ottenere degli effetti tipo vibrato o tremolo o delle variazioni continue di tonalità o di volume ecc.. È quindi necessario inizializzare alcuni registri che contengono questi valori di "zero" in modo da poter generare l'effetto voluto in base alle variazioni dei valori associati ai potenziometri;
- il registro dimensione della FIFO: si può decidere in fase di preset le dimensioni della FIFO che conterrà i dati in arrivo dalla tastiera musicale MIDI. È comodo poter variare queste dimensioni perché, essendo alta la velocità di trasmissione dei dati MIDI, si deve cercare di ottimizzare la memoria occupata dalla coda senza rischiare di perdere dati. Questi casi si possono verificare quando si utilizzano degli effetti come il vibrato, perché, muovendo il potenziometro dei PITCH BENDER, il MIDI trasmette i dati alla massima velocità consentita, essendo continua la variazione che si impone al potenziometro, perciò si ritrova nella FIFO un nuovo dato ogni 0,7 msec. Se il tempo di elaborazione dei dati ricevuti è superiore si corre il rischio di riempire tutta la FIFO e perdere delle informazioni.

Oltre a queste routines si hanno quelle di costruzione degli algoritmi sonori.

Poiché si è fatta la scelta di elaborare i dati attraverso tabelle prememorizzate, essendo ciò molto più veloce dell'elaborazione con operazioni aritmetiche, bisogna preparare tutte queste tabelle algoritmiche. Si passa così attraverso il seguente menù di scelte:

- delle forme d'onda;
- degli involuipi;
- della scala musicale: temperata o propria;
- della assegnazione dei vari potenziometri a parametri particolari, ecc..

Va sempre tenuto presente che in questa sede ci si riferisce a un particolare esempio applicativo; in altri casi le opzioni potranno essere ovviamente differenti.

Con l'inizializzazione di tutti questi parametri e con l'inizializzazione della tastiera musicale MIDI, in pratica si è creata la PARTITURA GESTUALE perché ad ogni gesto compiuto sulla tastiera musicale, corrisponde un particolare evento sonoro creato dal computer MIDI.

Attivazione degli interrupts.

Con questo blocco di routines si preparano le varie schede di interfaccia del computer MIDI a funzionare in trasmissione (eventuale pilotaggio di sintetizzatori digitali), o in ricezione (ricezione dati dalla tastiera MIDI, o da sequencer, o batterie elettroniche ecc.).

Un computer MIDI potrebbe essere in "comunicazione" con più di una tastiera musicale e con più dispositivi MIDI di uscita, perciò è necessario predisporre tutte le schede nel modo di funzionamento opportuno.

Queste routines di solito sono date dai costruttori stessi delle schede e non fanno altro che "accendere" particolari locazioni di memoria in modo da poter accettare i dati provenienti da un ingresso, se la scheda è in ricezione, o provenienti dal buffer di memoria per essere trasmessi, se la scheda funziona da trasmettitore.

Attesa per la ricezione.

Questa non è altro che una routine che testa una particolare locazione di memoria e implementa dei cicli di attesa fino a che non si accorge dell'arrivo del primo messaggio MIDI dalla tastiera musicale.

Si basa sul confronto fra i contatori di lettura e di scrittura della FIFO: se risultano uguali vuol dire che non sono stati ancora ricevuti dei dati, oppure sono già stati letti tutti, perciò bisogna attendere l'arrivo di nuove informazioni.

Gestione FIFO.

Si è pensato di far funzionare il sistema di trasmissione nel seguente modo:

- 1) arrivo del dato alla scheda di interfaccia MIDI;
- 2) attivazione di un interrupt;
- 3) memorizzazione del dato in una coda circolare dinamica.

Le relative routines di gestione di quest'ultima hanno il compito di gestire questa coda circolare dinamica come FIFO. In pratica si confrontano i due contatori della FIFO e, se sono uguali, si implementano dei cicli di attesa.

Si confronta il contatore di lettura con il registro che contiene la lunghezza della FIFO; se il contatore è minore si legge il dato e si incrementa il contatore; se è uguale, si legge il dato e si riavvolge il contatore per ricominciare.

Estrazione comando MIDI.

La routine estrae dalla FIFO il byte MIDI e attua un primo test sul bit più significativo del byte:

- il bit è uno zero: il byte è un byte dato e va eliminato;
- il bit è un uno: il byte è un byte di controllo e si continua per vedere di che comando si tratta.

Dopo aver ripristinato il byte, si testa il nibble più significativo per vedere se può essere un comando di tipo CHANNEL, in caso affermativo si opera un salto alla routine di riconoscimento dei default, come già accennato in precedenza.

In caso contrario si deve controllare se il byte non sia un comando REAL TIME perché questi comandi sono trattati in modo particolare, in quanto possono arrivare in qualsiasi momento della trasmissione, anche fra due bytes dati.

Infine si ha l'ultimo confronto con una maschera che riconosce se il comando è di tipo EXCLUSIVE ed opera il salto alla routine appropriata.

Se tutti i confronti sono stati negativi significa che il byte MIDI è un comando di tipo COMMON, perciò si salta al "riconoscimento comandi SYSTEM COMMON".

Riconoscimento dei default.

In fase di preset della tastiera musicale, vengono fatte delle scelte sul funzionamento del sistema, cioè si decide se lavorare in modo OMNI o POLY.

A questo punto si verificano le informazioni di questi modi di funzionamento perché si avrà un comportamento diverso dell'intero sistema a seconda del modo selezionato.

La routine decodifica il modo di funzionamento del sistema. Il protocollo MIDI infatti possiede quattro differenti modi di configurare il sistema:

- MODO OMNI: tutti gli apparecchi riceventi accettano le informazioni MIDI in arrivo, senza discriminare il numero del canale nel quale le informazioni stesse sono codificate.
- MODO POLY: gli apparecchi riceventi eseguono soltanto le istruzioni codificate con il numero di canale uguale a quello che l'utente li avrà predisposti a ricevere.
- MODO MONO: è il modo più potente di funzionamento. In un apparecchio ricevente polifonico e politimbrico, ogni voce può essere indirizzata da un canale MIDI, in maniera tale che un sequencer potrebbe suonare otto note simultanee, ciascuna con un timbro diverso.

Una apparecchiatura MIDI, sia ricevente che trasmittente, può operare in una sola modalità alla volta.

Inoltre l'abilitazione del modo POLY esclude quella del modo MONO, mentre il modo OMNI può essere ON o OFF con entrambi i modi precedenti; ne segue che i modi di funzionamento sono solo quattro.

- 1) OMNI ON, POLY: i messaggi sono riconosciuti indipendentemente dal canale di trasmissione e vengono assegnati in polifonia.
- 2) OMNI ON, MONO: come sopra, ma con controllo di una sola voce in monofonia.
- 3) OMNI OFF, POLY: i messaggi sono riconosciuti solo sul canale N e assegnati in polifonia.
- 4) OMNI OFF, MONO: i messaggi vengono riconosciuti nei canali da N a N + M-1, e assegnati in monofonia per M voci. M rappresenta il numero delle voci ed è specificato dal messaggio stesso.

Spiegati quali sono i modi di funzionamento di un sistema MIDI, si può passare a descrivere il comportamento della routine.

Il primo test che si compie è quello di vedere se il sistema deve funzionare in modo OMNI ON o in OMNI OFF:

- OMNI OFF: non ci si deve preoccupare dell'informazione del canale, perciò si deve saltare alla routine RICONOSCIMENTO COMANDO CHANNEL.
- OMNI OFF: bisogna continuare il test sul canale per sapere se l'informazione è destinata a passare o no.

Per verificare se il canale è attivo, si maschera il nibble più significativo del comando e si usa il rimanente nibble come indirizzo ad una tabella, settata in precedenza, che contiene l'informazione cercata.

- Caso canale attivo: si salta alla routine RICONOSCIMENTO COMANDO CHANNEL per caricare le informazioni successive.
- Caso canale spento: bisogna eliminare il comando non accettato con i seguenti byte dati seguenti: si estraggono dalla FIFO i successivi bytes, si controllano che non siano comandi REAL TIME, si eliminano se sono byte dati, si salta alla routine ESTRAZIONE COMANDO e si verifica se viene caricato un byte di controllo al posto del byte dato.

Riconoscimento dei canali attivi.

Sempre in fase di preset si assegnano i canali MIDI alle diverse zone della tastiera musicale. Dopo di ciò si deve riconoscere quale strumento MIDI è attivo in un determinato istante, assegnandogli il comando MIDI relativo. Inoltre i comandi MIDI potrebbero essere rivolti a dispositivi diversi dal "computer MIDI" perciò si deve riconoscere questo evento ed eventualmente eliminare tutti quei comandi e dati del messaggio che non lo riguardano.

Riconoscimento dei comandi.

Se i comandi riguardano il "computer MIDI" questi vengono accettati ed il byte di controllo del messaggio MIDI viene usato come indirizzo indiretto alla routine di servizio relativa. Nel caso contrario si ha l'eliminazione del comando e dei suoi byte dati.

Attivazione delle routines di servizio con elaborazione dati.

Si estraggono dalla FIFO i bytes dati e si passa alla loro elaborazione.

Qui di seguito sono riportati alcuni esempi:

— Routines di servizio comandi system common.

Come è stato detto in precedenza è stata fatta la scelta di trattare tutti i dati come se fossero degli indirizzi in particolari pagine di memoria.

Se si usano anche questi dati come indirizzi alle locazioni da essi indicate, si possono memorizzare in queste locazioni gli indirizzi delle routines di servizio ai relativi comandi COMMON.

Fatto il salto alla routine di servizio del comando COMMON, si attua l'elaborazione del comando stesso, lo si trasmette all'uscita, poi si ritorna alla routine di ESTRAZIONE DEL COMANDO MIDI vista in precedenza.

— Routines di servizio comandi exclusive.

Il comando EXCLUSIVE serve, nel protocollo MIDI, per inizializzare i vari dispositivi MIDI con informazioni esclusive per ciascuna macchina.

La trasmissione di questo comando viene così effettuata: byte di controllo, byte contenente l'ID CODE della macchina che deve accettare i dati, bytes dati, EOX per finire la trasmissione.

La routine carica il secondo byte del messaggio MIDI e controlla che non sia un comando REAL TIME.

Poi si verifica se l'ID CODE è accettato, quindi si caricano i byte dati, controllando sempre che non ci siano in mezzo dei comandi REAL TIME.

Se l'ID CODE è accettato, i byte dati vengono trasmessi al dispositivo indicato, altrimenti vengono eliminati. La procedura termina quando viene riconosciuto il comando EOX poiché si effettua il salto alla routine di ESTRAZIONE COMANDO.

— Routines di servizio comandi real time.

Anche per questi comandi vale il discorso fatto per i comandi di tipo COMMON.

— Routines di servizio dei comandi channel.

La routine tratta, come per i comandi visti in precedenza, il byte di controllo come un indirizzo di memoria ad una tabella contenente gli indirizzi delle routines di servizio dei comandi selezionati.

Dopo il salto indiretto alla routine selezionata si passa al blocco di ELABORAZIONE DATI. Il blocco è solamente indicativo in quanto potrebbe contenere tutte quelle operazioni di elaborazione che un musicista potrebbe desiderare per arricchire di effetti la propria performance.

La fase successiva può essere: o la RICODIFICA MIDI, perché si ha a disposizione un dispositivo che accetta in ingresso i messaggi MIDI, perciò i dati elaborati vengono ricodificati in messaggi MIDI, oppure la CODIFICA IN PROTOCOLLI PARTICOLARI. Questo succede quando si hanno a disposizione dei sintetizzatori digitali sonori che necessitano, per funzionare, di un protocollo particolare di ingresso (questo è fra l'altro il caso dell'esempio applicativo qui esaminato).

— Trasmissione ai sintetizzatori.

A questo punto si ha, infine, la trasmissione dei vari messaggi ai sintetizzatori digitali. Questa

trasmissione può essere differenziata per ciascun sintetizzatore, nel caso si abbiano più sintetizzatori a disposizione, a seconda del canale MIDI selezionato.

Fig. 1

```

+-----+ /|
|SYNTH A |--=I |
+-----+ \|

+-----+ /|
| M |--|SEZ. | |SEZ. | |SEZ. | /| | | |
| I |--|RICO | |PRO | |CODI | |--|SYNTH B |--=I |
| D |--|NOSCI| |GRAMMA| |FICA | \|
| I |--|TORE | |ZIONE | |TORE | \|
+-----+ +-----+ +-----+ +-----+

: :
*---COMPUTER M.I.D.I.---* |SYNTH C |--=I |
+-----+ \|

```

Fig. 2

```

+-----+
| MAIN PROGRAM |
+-----+

|
+-----+
| PREPARAZIONE DEI DEFAULT |
+-----+

|
+-----+
| ATTIVAZIONE DEGLI INTERRUPT |
+-----+

|
+-----+
| ATTESA PER LA RICEZIONE |
+-----+

|
+-----+
| GESTIONE FIFO |
+-----+

|
+-----+
+-->-----| ESTRAZIONE DEL COMANDO MIDI |
+-----+

|
+-----+
+--<-----| RICONOSCIMENTO DEFAULT |
+-----+

|
+-----+
+--<-----| RICONOSCIMENTO CANALI ATTIVI |
+-----+

|
+-----+
+--<-----| RICONOSCIMENTO COMANDI |
+-----+

|
+-----+
| ATTIVAZIONE R.S. ED ELABORAZ. |
+-----+

|
+-----+
| CODIFICA PROTOCOLLO USCITA |
+-----+

|
+-----+
+--<-----| TRASMISSIONE AI SYNTH |
+-----+

```

LINGUAGGIO COMPOSITIVO IN AMBIENTE MIDI

G. Dimino

La possibilità di utilizzare apparecchiature operanti in ambiente MIDI nella sperimentazione musicale elettronica è fortemente ostacolata, a prescindere dai limiti dello standard e delle macchine che lo utilizzano, dall'assenza sul mercato di software musicale in grado di fornire al musicista il supporto necessario ad esercitare un controllo completo su tutti i parametri del suono.

Nel progettare un programma che potesse sopperire, almeno in parte, a questa mancanza, si è preferito rinunciare a notazioni grafiche (pentagrammatiche o meno) ed al controllo interattivo o real-time dei sintetizzatori in favore di una maggiore versatilità e potenza della tecnica di definizione degli eventi musicali.

È stato quindi realizzato un processore musicale che, ricevendo in ingresso una partitura descritta per mezzo di un linguaggio apposito, genera la sequenza di codici MIDI che ne permettono l'esecuzione.

Il processore permette di assegnare i valori ai parametri non solo esplicitamente, come avviene nei music composer commerciali, ma anche mediante espressioni; esso permette, inoltre, di utilizzare tabelle di dati predefinite e costrutti di gestione del programma tipici dei linguaggi tradizionali, quali loop, if, macro. Per costruire le espressioni è disponibile una libreria che comprende funzioni di tipo matematico, logico e di scansione di tabelle. La notazione utilizzata è di tipo Lisp:

```
(func-name lista-par)
```

dove i parametri possono a loro volta essere funzioni. Le espressioni vengono valutate dal processore non appena incontrate e sostituite con il valore calcolato. L'utente ha a disposizione un set di 128 registri dotati di autoincrement e autodecrement utili a memorizzare valori che devono essere riutilizzati in altre parti della composizione. La rappresentazione interna dei numeri è di tipi floating point su 4 byte, per garantire una dinamica sufficiente ad elaborazioni complesse. La conversione nel formato richiesto dalle singole istruzioni viene effettuata automaticamente dal processore. Nel caso in cui il valore assegnato ad un parametro si trovi al di fuori del range ammesso, viene applicato il criterio della saturazione.

Il linguaggio è composto da due tipi di statement: eventi musicali e istruzioni di controllo o directives. Una partitura è formata da una sequenza di eventi musicali inframmezzati da istruzioni di controllo.

Un evento musicale ha la seguente scrittura:

```
time stamp  
sequenza di istruzioni musicali  
carattere di fine evento
```

Il time stamp determina a quale distanza dall'ultimo evento processato va posizionato l'evento attuale. Questa distanza viene definita in termini di colpi di clock, e la sua conversione in secondi è perciò subordinata alla frequenza di clock che verrà scelta al momento dell'esecuzione. Per evitare problemi di ordinamento degli eventi e di risoluzioni illusorie, si assume che il time stamp sia rappresentato da un numero intero e maggiore di zero.

L'evento musicale viene descritto utilizzando una o più istruzioni musicali. Queste istruzioni permettono di generare tutti i messaggi MIDI di tipo Channel (ovvero Note on/off, Pitch

Bend, After Touch, ecc.) e tutti i messaggi di tipo Parameter Change definiti nel System Exclusive della Yamaha per le macchine DX7 e TX7. Ovviamente il set di istruzioni disponibili può facilmente essere ampliato in modo da generare anche altri tipi di messaggi. Per ogni messaggio dei tipi citati, esiste un'istruzione corrispondente, in modo da garantire la massima versatilità del linguaggio, inoltre sono definite alcune istruzioni a più alto livello in grado di generare contemporaneamente più messaggi MIDI e compiere trasformazioni sui parametri. Particolarmente interessanti fra queste sono le istruzioni che permettono di assegnare agli operatori del DX7 la frequenza di oscillazione in Hertz e le durate dei segmenti dei generatori di inviluppo in secondi, dando quindi la possibilità al compositore di lavorare con oggetti e unità di misura a lui familiari.

A differenza degli eventi musicali, le directives non generano alcun messaggio MIDI, bensì eseguono operazioni di controllo del flusso del programma, definizioni e chiamate di macroistruzioni, caricamento di tabelle ed inizializzazione di registri.

Per quanto riguarda il controllo del flusso esistono due tipi di costruttori: loop ed if. Il costrutto loop ha la seguente sintassi:

```
.LOOP count  
body  
.ENDL
```

Esso fa sì che body (una qualsiasi porzione di programma) venga processato un numero di volte pari al valore che assume il parametro count all'istante in cui viene incontrata la directive-loop, ed è quindi analogo ad un costrutto for di un linguaggio tradizionale. Esiste, inoltre, un'istruzione di break

```
.BREAK cond
```

che permette di terminare il loop non appena incontrata se cond assume un valore diverso da zero. Essa può essere inserita ovunque all'interno di un loop e permette perciò, fra le altre cose, di formare costrutti di tipo while e repeat.

Il costrutto if

```
.IF cond  
body-1  
.ELSE  
body-2  
.ENDIF
```

fa sì che se cond è diverso da zero venga processato body-1 altrimenti venga processato body-2.

Sequenze di statement utilizzate più volte possono essere definite come macroistruzioni tramite le directives

```
.MACRO macro-name  
body  
.ENDM
```

e richiamate con

```
.CALL macro-name
```

che produce una copia esatta di body, come se questo fosse stato scritto esplicitamente nel punto della chiamata.

Le directives

.LOAD file-name tab-name

e

.EQU reg par

servono, rispettivamente, a caricare in memoria tabelle di dati, a cui viene assegnato il nome logico tab-name, e ad inizializzare i registri.

Ovunque siano richiesti parametri di tipo numerico (istruzioni musicali, time stamp, directives), questi possono sempre essere assegnati tramite costanti, registri o impressioni.

Il processore produce un file oggetto costituito da una sequenza di record contenenti i messaggi MIDI e le relative temporizzazioni. Il formato dei record è il seguente:

Time stamp 16 bit
Lunghezza Dati 8 bit
Dati da 1 a 255 byte

Il codice di fine sequenza è definito come un record avente i campi Time Stamp e Lunghezza Dati messi a zero.

Questo file può essere eseguito direttamente tramite un programma che invii i dati sul canale MIDI rispettando le temporizzazioni imposte dai Time Stamp oppure convertito in un formato opportuno ed utilizzato da un music performer. Questa seconda alternativa appare particolarmente attraente in quanto consentirebbe di integrare le possibilità offerte dal processore ora presentato con quelle offerte dai pacchetti di software commerciali, di grande aiuto per ciò che riguarda le operazioni di montaggio di più sequenze, mixaggio e gestione dei canali MIDI e delle temporizzazioni.

Esempio di partitura.

.LOAD GAUSS.DAT TAB1

/il file GAUSS.DAT contiene una tabella che modella una distribuzione
/ gaussiana di valori compresi fra 0 e 255

.MACRO SEG1

/ esegue sul canale 2 un accordo di tre note con dinamica 90 e durata 6
4

N.2 C5 90 6

N.2 Bb5 90 6

N.2 D6 90 6

%

.ENDM

.MACRO SEG2

/ al registro R2 viene assegnato il valore 60(=C5)

.EQU R2 60

/ l'esecuzione di due note viene ripetuta tre volte

/ a distanza 2 tic l'una dall'altra;

/ P2 si incrementa di 1 ogni volta per effetto del +

```
.LOOP 3
```

```
2
```

```
N.2 R2+ 90 6
```

```
N.2 R2+ 90 6
```

```
%0
```

```
.ENDL
```

```
.ENDM
```

```
/ ad R1 viene assegnato un valore estratto in modo casuale dalla tabella
```

```
/ TAB1 e diviso per 4
```

```
.EQU R1 (RND TAB1 4)
```

```
.IF (GTE R1 10)
```

```
/ se R1 >= 10 chiama la macro SEG1
```

```
.CALL SEG1
```

```
.ELSE
```

```
/ altrimenti esegui il loop
```

```
.LOOP 5
```

```
.BREAK (EQU R1 10)
```

```
/ esci dal loop se eseguito 5 volte oppure se R1 = 10
```

```
.CALL SEG2
```

```
.ENDL
```

```
.ENDIF
```

LA SOUND PROCESSING UNIT 03 (SPU 03):
UN PICCOLO SISTEMA FLESSIBILE PER USO PREVALENTEMENTE DIDATTICO

A. D'Agata
S.I.M. - Via Riccardo Grazioli Lante 30 - Roma

La Sound Processing Unit (SPU 03) è un processore di segnali orientato alla sintesi del suono della serie SPU, progettato da Giorgio Nottoli della SIM di Roma e realizzato presso i laboratori SIM nel 1984.

Si tratta del più piccolo processore della serie SPU, ideato e realizzato per assolvere a compiti specifici. Infatti, a differenza di altri sintetizzatori esistenti sul mercato, è un sistema aperto e flessibile tale da permettere applicazioni sia nel campo della computer music che della didattica.

La board è infatti collegata ad un microcomputer di vasta diffusione e basso costo come il C64. Ce ne rende l'uso piuttosto semplice in quanto essa è completamente programmabile da C64 in un linguaggio ad alto livello come il Basic.

Con la SPU 03 è possibile generare 16 suoni sinusoidali in sintesi additiva oppure 8 strumenti in modulazione di frequenza, con una frequenza di campionamento di 16 KHz.

Sia i 16 suoni in sintesi additiva che gli 8 strumenti FM sono programmabili indipendentemente l'uno dall'altro per quanto riguarda frequenza, ampiezza ed inviluppo a quattro segmenti.

Descrizione hardware.

- Processore digitale di segnali TMS 32010 della Texas Instruments.
20 MHz/clock.
(Si tratta di un chip molto potente che unisce la specificità della elaborazione dei segnali alle operazioni tipiche dei microprocessori).
- Memorie di programma 1 KWord RAM (caricabile dal microcomputer di controllo C64).
- Interfaccia I/O da/a microcomputer di controllo (tramite PIA della famiglia MC6821).
(L'interfacciamento del sistema viene effettuato sulla porta del C64).
- DAC 12 bits.
- Filtro passa-basso BUTTERWORTH 48 db/ottava. Frequenza di taglio 4KHz.
L'alimentazione del blocco digitale è prelevata dal C64 (+5 Volts); il blocco analogico viene alimentato esternamente con un alimentatore - 15 Volts.

Come abbiamo visto il componente centrale del sistema è il processore digitale di segnali TMS 32010, lo stesso che troviamo nei moduli di Soft, Machine (SPU 02). Ciò rende i due sistemi tra loro compatibili (pur avendo la SPU 03 delle limitazioni rispetto a Soft Machine).

Infatti la SPU 03 è un sistema non espandibile (Soft Machine è un sistema modulare); inoltre ha soltanto 1K di memoria ed il convertitore è a 12 bits (contro i 16 della SPU 02).

La compatibilità dei due sistemi fa sì che la SPU 03 possa essere intesa, (riguardo alla serie SPU), come un sistema rivolto prevalentemente ad uso didattico. Ciò è consentito infatti anche dal basso costo e dalla programmabilità in Basic tramite C64.

Disponibilità software.

Il software attualmente disponibile è scritto in linguaggio Basic esteso (Simon's Basic): esso riguarda sia la sintesi additiva che la modulazione di frequenza.

Per quanto concerne la sintesi FM viene fornito di base, assieme alla SPU 03, il programma di inizializzazione del TMS 32010 ed un editore di timbri con involuppo a 3 segmenti (attack, sustain, release).

Per quanto riguarda invece la sintesi additiva, assieme al programma di inizializzazione del TMS 32010, viene fornito un software sviluppato dal Gruppo di Informatica musicale di Bari, riguardante varie serie armoniche e la relativa visualizzazione dello spettro.

È inoltre disponibile software per la sintesi additiva che distribuisce le frequenze secondo la formula di MacAdams, con cui è stato realizzato il lavoro "Specchi" di Serena Tamburini.

SESSIONE ELABORAZIONE NUMERICA
DEL SUONO

FORME D'ONDA PER LA SINTESI GRANULARE SINCRONA

G. De Poli - C.S.C. Dip. Elettronica e Informatica - Università di Padova
A. Piccialli - Dipartimento di Fisica - Università di Napoli

1. SINTESI GRANULARE.

Alla base della sintesi granulare sta l'idea che si può ottenere un suono complesso mediante la successione temporale di suoni semplici di breve durata. Qualcosa di analogo avviene nel cinema, dove si ottengono le immagini in movimento mediante la rapida successione di immagini statiche.

Nel caso del suono questa tecnica può essere realizzata prendendo vari suoni statici e agguinandoli uno dopo l'altro nel tempo. Evidentemente le giunzioni devono essere fatte con particolare cautela per riuscire a fondere bene due suoni elementari consecutivi, senza avere cioè discontinuità di ampiezza o fase.

I parametri su cui si può agire, sono le caratteristiche del suono elementare, determinato dalla forma d'onda del grano e dal suo involuppo, e la collocazione temporale dei grani stessi.

1.1 Caratteristiche dei grani.

Per quanto riguarda le caratteristiche del singolo grano, una prima strada consiste nello scegliere una forma d'onda già complessa, anche se statica, prendendo ad esempio una piccola porzione di un suono reale, in analogia a quanto avviene nel cinema. I vantaggi sono una buona riproduzione delle caratteristiche originarie, e una buona efficienza implementativa. Gli svantaggi derivano dalla grande quantità di grani diversi necessari (non si può quasi mai riutilizzare un grano) e dai problemi di controllo compositivo delle caratteristiche globali del suono risultante.

Dal punto di vista implementativo i singoli grani possono essere memorizzati in una serie di tabelle. Queste ultime possono essere precalcolate in modo da ottimizzare le giunzioni, come nella sintesi della voce per segmenti, oppure derivate in tempo reale da un segnale in ingresso, come ad esempio quando si vuole ridurre o aumentare la durata del suono senza cambiarne l'altezza.

Una seconda possibilità implementativa, che consente di ridurre il numero di tabelle necessarie, consiste nel modificare le tabelle durante la sintesi del suono, mediante semplici procedimenti; ad esempio scandendola con passo non unitario, attuando così una trasposizione in frequenza, o con passo non costante, attuando così una modulazione di fase.

Una terza possibilità è generare direttamente la forma d'onda di ogni grano a partire da parametri fissi per quel grano, come si fa ad esempio nella sintesi della voce, cambiando i coefficienti del filtro di predizione lineare ad ogni frame. Si può notare come questo metodo non necessiti di memoria per le forme d'onda dei singoli grani, a spese però di una maggiore complessità di calcolo. Inoltre essendo un metodo globale non sempre è agevole controllare compositivamente le singole caratteristiche del suono risultante.

Una strada opposta è la scelta di forme d'onda semplici per i singoli grani. Normalmente in questo caso si usa la sinusoidale e si ottiene la necessaria complessità dinamica del suono con una rapida sequenza e sovrapposizione di grani, i cui parametri vengono scelti con criteri

probabilistici all'interno di determinati intervalli che possono variare. Si ottiene così un effetto di "texture" o di massa sonora più o meno densa, chiara o scura, a seconda delle scelte effettuate.

Le caratteristiche del suono dipendono oltre che dalla forma d'onda del grano, anche dal suo involuppo. Esso infatti viene usato per limitare la durata del grano e per collegare i grani tra loro; deve essere scelto in modo da assicurare la continuità del segnale risultante e risulta acusticamente particolarmente importante quando la forma d'onda del singolo grano è semplice. Gli involuppi finora proposti sono scelti per lo più con criteri euristici. Si riconosce che l'involuppo corrisponde all'uso di finestre nell'analisi dei segnali. Nella sintesi, la finestra rettangolare può essere usata solo in casi particolari; più spesso essa deve essere opportunamente raccordata con lo zero. Per il raccordo sono stati proposti l'uso di mezzo coseno rialzato (finestra di Tukey), di mezza gaussiana (Roads), di una linea retta (finestra trapezoidale) (Truax). Questo ultimo caso corrisponde a fare la congiunzione con una interpolazione lineare tra due grani successivi.

1.2 Collocazione temporale dei grani.

La collocazione temporale dei grani in genere avviene ad intervalli fissi di $10 \div 20$ millisecondi, se si vuole ottenere una variazione continua, o ad intervalli maggiori se si vuole accentuare la riconoscibilità della granularità della "texture" o se è accettabile un suono risultante poco dinamico. Questo modo di aggiornamento, che chiameremo asincrono, in genere crea dei problemi nella sintesi di suoni quasi periodici, in quanto manca la continuità di fase tra grani consecutivi e si crea una fastidiosa intermodulazione tra segnale e frequenza di aggiornamento dei grani. La proposta di variare aleatoriamente questa frequenza, non dà risultati soddisfacenti. La continuità della fase può essere ottenuta invece manipolando le fasi dello spettro dei singoli grani (procedimento che risulta piuttosto complicato) oppure più facilmente aggiornando i grani in modo sincrono con il periodo.

In questo lavoro noi seguiremo questa strada, studiando le forme d'onda per la sintesi granulare sincrona con il periodo, con l'intento di fornire al compositore uno strumento facile e versatile per la sintesi di alta qualità, e che sia inoltre efficiente.

2. MODELLO STUDIATO.

Si vuole consentire al musicista di definire l'evoluzione del suono in termini percettivamente significativi; l'involuppo spettrale è un buon riferimento a questo proposito. La versatilità è ottenibile non limitandosi a poter controllare solo l'effetto globale, ma anche le singole posizioni, o formanti, dell'involuppo spettrale in maniera indipendente. A questo scopo sarà dedicata una sequenza di grani ad ogni zona dello spettro in movimento rispetto alle altre. Riprendendo l'analogia con il cinema, usiamo una tecnica analoga ai cartoni animati, in cui l'immagine è composta di varie parti, elementari, contemporaneamente in movimento tra loro.

Viene pertanto studiata una sintesi granulare additiva sincrona con il periodo, che viene ad avere aspetti in comune con la sintesi per formanti. Infatti il metodo di sintesi può essere pensato come un banco di filtri con eccitazione impulsiva, in cui ogni grano è la risposta all'impulso di ciascun filtro. Risulta pertanto che l'effetto di ogni singola sequenza di grani è determinato dalla risposta in frequenza del filtro e quindi dalla forma del grano in maniera indipendente dalla frequenza di eccitazione. Al variare di quest'ultima vengono esplorate parti

diverse dell'inviluppo spettrale, producendo quindi un suono variabile e più nettamente caratterizzato dalla costanza dell'inviluppo spettrale.

Da questo discende che la collocazione temporale dei grani sarà all'incirca periodica, con variazioni di tipo sinusoidale e di tipo aleatorio (ad es. $1/f$) coordinate per i vari grani contemporanei. A queste piccole variazioni saranno sovrapposte le variazioni necessarie per definire l'articolazione del suono complessivo, anche in relazione ai suoni adiacenti.

Per la scelta delle forme d'onda dei singoli grani, oltre a quanto detto precedentemente, si può osservare quanto proposto nella sintesi per formanti. Come per la sintesi granulare, anche qui possono distinguersi i metodi globali da quelli additivi. Fra i globali si possono citare Bass (81), che usa finestre di Blackman pesate, e De Poli (86), che usa filtri FIR ottimi a fase minima. Fra gli additivi, che sono quelli che qui ci interessano, si possono citare il VO-SIM (Kaegi Tempelaars) che usa sequenze decrescenti di impulsi sen^2 , Rodet (80) che usa un esponenziale decrescente, raccordato all'inizio con lo zero, moltiplicandolo per mezzo coseno rialzato, Lienard (87) che usa due mezzi coseni rialzati di durata diversa. Questi ultimi due sono poi moltiplicati per un coseno, per ottenere l'opportuno spostamento in frequenza. Tutti i metodi additivi esaminati però non hanno linearità di fase, per cui nella somma dei vari grani contemporanei risulta che per alcuna frequenza gli spettri si sommano in fase, per altre con fase opposta. Ne risultano effetti indesiderati e fastidiosi di cancellazione ed interferenze tra formanti adiacenti in varie zone, che alterano le caratteristiche del suono risultante. Le soluzioni proposte a questo problema, non risultano soddisfacenti.

La nostra proposta è di risolvere radicalmente questo problema usando forme d'onda derivate da filtri FIR a fase lineare. Ciò consente infatti di sommare le singole componenti senza problemi, nonché, tabulando la forma d'onda, si ottiene una implementazione efficiente.

3. CONTROLLO DINAMICO DELLE FORME D'ONDA NEL TEMPO.

Per ogni formante si tratta di progettare un filtro FIR in fase lineare passabanda, di cui si vuole controllare la frequenza, ampiezza, larghezza di banda e forma, possibilmente in maniera dinamica. Noi proponiamo di partire da alcuni filtri FIR passa basso prototipi e di alterarne le caratteristiche in modo dinamico, per maggior flessibilità, mediante trasformazioni elementari.

Sia $h_0(n)$ la risposta all'impulso (simmetrica e dispari) del filtro prototipo passa basso; la versione equivalente passa banda avrà la seguente risposta all'impulso:

dove f_1 è la frequenza normalizzata del formante. Per quanto riguarda il controllo della larghezza di banda del formante, si osserva che una riduzione della durata temporale NT della risposta impulsiva determina un allargamento inverso della larghezza di banda. Pertanto se R è il rapporto tra la larghezza di banda desiderata e quella del prototipo, la risposta all'impulso $h_d(n)$ desiderata diventa

$$h_d(n) = h_0(R*n)$$

Essa può essere ottenuta semplicemente leggendo i valori nella tabella contenente $h_0(n)$ con passo R non unitario. Per valori di R non interi si può ricorrere ad interpolazione oppure a tabelle più fitte. Queste ultime equivalgono a filtri FIR progettati per frequenza di campionamento superiori.

La forma del formante così prodotto dipende dal filtro FIR prototipo. Per cui un primo controllo di questo parametro si ottiene avendo più tabelle contenenti i prototipi adatti a diverse situazioni. Spesso si vuole fare variazioni continue e controllare indipendentemente varie parti della forma. Ad esempio controllare separatamente la larghezza di banda ed il piedistallo (parte dello spettro a minor energia adiacente alla parte centrale del formante) questo consente un più accurato controllo delle zone di transizione tra formanti adiacenti.

A questo scopo sono state individuate alcune tecniche di trasformazione dinamica del prototipo, in parte derivate da strategie in uso nella sintesi dei segnali musicali. Fra le più promettenti segnaliamo la lettura della tabella con passo non costante, la modulazione di ampiezza e la distorsione non lineare. In figura sono rappresentati ulteriori esempi di queste trasformazioni applicate ad una forma d'onda prototipo di tipo gaussiana.

4. FORME D'ONDA PROTOTIPO.

La scelta della forma d'onda prototipo, consiste nel progetto di un filtro passa basso FIR a fase lineare avente una risposta in frequenza che approssimi quella desiderata del formante. Inoltre si cercherà di ridurre la lunghezza, per evitare un eccessivo numero di sovrapposizioni tra grani adiacenti. Il progetto dovrà tener conto delle successive trasformazioni nel senso che il prototipo deve potersi adattare facilmente ai vari casi e non presentare comportamenti anormali o indesiderati. Abbiamo sperimentato sia forme d'onda derivate dalle tecniche standard di progetto di filtri FIR come il metodo della finestra e quello di approssimazione minimax che quelli derivanti da formule analitiche, alcune delle quali sono impiegate come finestre (Harris). Fra le analitiche si possono citare la gaussiana, la secante iperbolica, il rapporto tra secanti iperboliche. Forse il caso più interessante sono le finestre di Nuttall (83) e quelle basate sui polinomi di Gegenbauer (Streit), che consentono di controllare indipendentemente sullo spettro la pendenza dei lobi e il rapporto tra larghezza del lobo centrale e ampiezza dei lobi laterali. Esse prevedono come casi particolari le finestre di Kaiser e di Dolph-Chebyshev.

5. CONCLUSIONI.

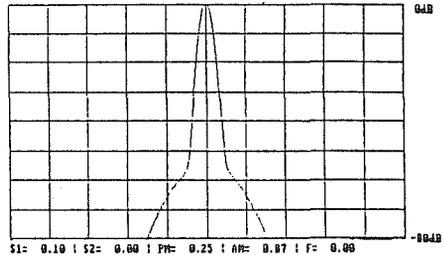
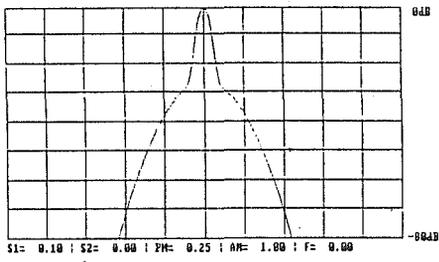
È stata esaminata la sintesi granulare discutendone gli svantaggi e i problemi ancora non risolti nelle proposte precedenti. Si è visto che la collocazione temporale di grani sincrona con il periodo, risolve alcuni importanti problemi. Sono state studiate le forme d'onda più adatte e le tecniche di modificazione per un controllo dinamico delle caratteristiche dei grani a partire da forme d'onda prototipo. Sono infine stati discussi i criteri di scelta delle forme d'onda prototipo, in relazione ai diversi obiettivi.

Bibliografia.

BASS S.C., GOEDDEL T.W. — *The efficient digital implementation of subtractive music synthesis*. IEEE Micro, vol. 1, n. 3, p. 24-37, Aug. 1981.

CAVALIERE S., PICCIALLI A. — *Phase modulation with interpolated time functions: synthesis by formants*. Proc. ICMC 86, p. 293-297, 1986.

- DE POLI G. — *A tutorial on digital sound synthesis techniques*. Comp. Music J., vol. 7, n. 4, p. 8-26, 1983.
- DE POLI G., LONGO G., MIAN G.A. — *An effective software tool for digital filter design*. Proc. IASTED Int Symp. MSPC, p. 237-242, 1986.
- FAIRBANKS G., EVERITT W.L., JAEGER R.P. — *Method for time or frequency compression - expansion of speech*. IRE Trans. AU, vol. 2, n. 1, p. 7-12, 1954.
- GABOR D. — *Theory of Communication*. J. of the IEE, vol. 93, p. 429, 1946.
- GABOR D. — *Acoustical quanta and the theory of hearing*. Nature, 159, 1044, p. 591-594, 1947.
- HARRIS F.J. — *On the use of window for harmonic analysis with discrete Fourier transform*. Proc. IEEE, vol. 66, n. 1, p. 51-85, jan. 1978.
- HOLMES J. — *Formant Synthesizers: Cascade or Parallel?'*. Speech communication, vol. 2, p. 251-273, 1983.
- KAEGI W., TEMPELAARS S. — *VOSIM: A new sound synthesis system*. J. Audio Eng. Soc., vol. 26, n. 6, 1978.
- JONES D.J., PARKS T.W. — *Time scale modification of signals using a synchronous Gabor technique*. Proc. IEEE Work. ASPAA, 1987.
- LIENARD J. — *Speech analysis and reconstruction using short-time elementary waveform*. Proc. ICASSP 87, P. 948-951, 1987.
- MATHEWS J.D., BREAKALL J.K. KARAWAS G.K. — *The discrete Prolate Spheroidal Filter as a digital signal processing tool*. IEEE Trans ASSP, 33, 6, p. 1471-1478, 1985.
- MEYER Y., JAFFARD S, RIOUL O. — *L'analyse par ondelettes*. Pour la Science, n. 119, p. 28-37, Sept. 1987.
- NUTTAL A.H. — *A two-parameter class of Bessel weighting for spectral analysis or array processing*. IEEE Trans. ASSP, 31, 5, p. 1309-1312, 1983.
- PORTNOFF M.R. — *Time-scale modification of speech based on short-time Fourier Analysis*. IEEE Trans. ASSP, vol. 29, n. 3, p. 374-390, June 1981.
- C. ROADS — *Granular synthesis of sound*. Comp. Music J., 2, 2, p. 61-62, 1978.
- C. ROADS — *Granular synthesis of sound: Past research and future prospects*. Atti 6 CIM, Napoli, 1985.
- RODET X. — *Time domain formant wave function synthesis*, in Spoken Language Generation and Understanding, ed. J. G. Simon, Dordrecht, 1980.
- STREIT R.L. — *A two-parameter family of wights for non recursive digital filters and antennas*. IEEE Trans. ASSP, 32, 1, p. 108-118, Feb. 1984.
- TRUAX B. - *Real time granular synthesis with the DMX-1000*. Proc. ICMC 86, p. 231-235, 1986.
- TRUAX B. — *Real time granulation of sampled sound with the DMX 1000*. Proc. ICMC 8), p. 138-145, 1987.



$$g(n) = e^{-2\left(\frac{n}{N \cdot S1}\right)^2}$$

$$f(x) = x(1 + PM \cdot x^{24})$$

$$h(n) = (f \text{ AM} \cdot g(n))$$

$$N = 512$$

UN CONVERTITORE *PITCH-TO-MIDI* STAND-ALONE

N. Bernardini, L. M. Del Duca

S.I.M. - Società per l'Informatica Musicale - Via Grazioli Lante, 30-76 - Roma

1. INTRODUZIONE.

Lo stato dell'arte delle tecnologie nel campo degli strumenti musicali elettronici ha raggiunto un livello tale da permetterne un uso piuttosto intensivo sia nel campo della musica industriale sia in quello più sperimentale.

Questo sviluppo è stato favorito, da una parte, dall'evoluzione delle tecniche di progettazione dei circuiti integrati a larghissima scala di integrazione (*VLSI*), e dall'altra parte dalla messa a punto, da parte dell'industria, del protocollo di comunicazione MIDI (che sta per *Musical Instrument Digital Interface*), per il collegamento digitale standardizzato tra unità diverse (1).

Il protocollo MIDI è un protocollo basato sulla generazione di segnali di controllo per eventi musicali definiti. Esso è di semplice e diretta utilizzazione per la definizione e la sincronizzazione di note da suonare, per cambi globali di parametri, per controlli locali in tempo reale, ecc. La sua utilizzazione per l'elaborazione del segnale in tempo reale è, se non impossibile, molto complicata.

1.1 Stato attuale della *Pitch-to-MIDI conversion*.

Sin dai primi passi del MIDI, quindi, si è rivelata necessaria una interfaccia segnale audio/codice MIDI almeno sul piano di eventi sonori comuni (note di strumenti musicali non elettronici, ad esempio) per rendere più flessibile l'utilizzazione di apparecchiature digitali in ambito musicale. Un'interfaccia di questo tipo comporta quasi obbligatoriamente una tecnologia in grado di effettuare rivelazioni di frequenza in tempo reale (*Pitch Detection*).

Sono quindi comparsi sul mercato numerosi *Pitch-to-MIDI converters* commerciali, sia a basso costo (tra le 400.000 e le 600.000 lire - prezzo di listino) che di costo elevato (dai due ai quattro milioni di lire - prezzo di listino). I *converters* a basso costo presentano caratteristiche inaccettabili per la utilizzazione in buona parte delle applicazioni musicali, mentre quelli ad alto costo non trovano utilizzatori disposti a pagare prezzi così alti per controllare unità di costi notevolmente inferiori.

I motivi di questa inadeguatezza derivano dal fatto che le tecnologie e gli algoritmi efficienti per il rilevamento frequenziale in banda audio sono stati sviluppati nelle applicazioni di telefonia riguardanti la voce (2), con caratteristiche troppo ristrette o comunque sostanzialmente diverse da quelle necessarie nelle applicazioni musicali. In alternativa possono essere utilizzati algoritmi generalizzati (ad es. l'autocorrelazione o la *Fast Fourier Transform*) che necessitano di una tecnologia troppo costosa.

1.2 Problematiche principali.

I principali problemi da affrontare sono, in questo caso, la precisione del rilevamento

in frequenza, la sua stabilità ed il tempo iniziale di calcolo. Tra l'altro, gli algoritmi utilizzati sinora debbono spesso attendere l'esecuzione di uno o più periodi della frequenza da rilevare, rendendo così il tempo iniziale di risposta pesantemente dipendente dalla nota suonata.

Nelle apparecchiature commerciali da noi provate il tempo iniziale di calcolo si aggira intorno ai 10 msec per note al di sotto del do centrale per crescere a tempi dell'ordine dei 30 msec su note molto gravi (e.g. do grave del violoncello). La stabilità della rilevazione e la sua precisione dipendono in maniera cruciale dal timbro del suono cui si intende rilevare la frequenza e dall'articolazione del passaggio musicale in questione. Data la lentezza dei sistemi studiati, lunghe teorie di semicrome suonate "staccato" a tempo metronomico $MM = 100$, tanto per fare un esempio assolutamente comune in campo musicale, sono al di fuori della portata dei sistemi a basso costo.

2. IL PROGETTO SIM-PITCH TO MIDI CONVERSION.

Abbiamo quindi intrapreso, in seno alla Società per l'Informatica Musicale (SIM) di Roma, uno studio per la realizzazione di una scatola di conversione *Pitch-to-MIDI* a basso costo, di caratteristiche elevate ed in grado di operare in completa autonomia (*stand-alone*).

2.1 Considerazioni iniziali.

Date le problematiche sopradescritte (vedi §1.2), abbiamo scartato l'ipotesi di algoritmi di rilevamento frequenziale nel dominio del tempo (e.g. *zero-crossing* e derivati), molto efficienti ma poco precisi e troppo dipendenti dalle durate dei periodi che si intendono rilevare (si tenga presente che tali durate si aggirano intorno ai 15 msec per le note più gravi).

Ci siamo quindi concentrati su algoritmi nel dominio della frequenza, ed in particolare su quelli che non comportassero "finestrature" (un'acquisizione di n. campioni senza poterli elaborare) troppo lunghe e gravose sul tempo di conversione, cercando di trovarne uno con caratteristiche efficienti nella nostra applicazione in particolare.

Abbiamo poi individuato la componentistica ed elaborato un progetto *hardware* per la realizzazione fisica dei tali algoritmi con lo scopo di ridurre al massimo la complessità del sistema (e quindi i suoi costi).

2.2 Strategie di realizzazione.

L'algoritmo prescelto viene descritto nella letteratura come *Algoritmo di Goertzel* (3, 4, 5). A grandi linee (l'algoritmo è descritto in dettaglio nel §2), si tratta di una trasformazione discreta di Fourier (DFT), svolta senza finestrature in intorni frequenziali disponibili a piacere. Nella realizzazione pratica, abbiamo affiancato a questo algoritmo una logica di decodifica ed un meccanismo adattivo per ottenere i risultati desiderati.

I componenti del sistema *hardware* sono soltanto due: un elaboratore numerico di segnali integrato *eprom*-programmabile *Texas Instruments TMS320C17* ed un convertitore integrato ADC *combo codec*. La caratteristica principale di questo sistema (descritto nel §4) è quindi l'estrema semplicità, alla quale si affianca una notevole potenza di calcolo e numerose possibilità d'interfacciamento con l'esterno (e.g., interfaccia MIDI interna, ecc.).

3. L'ALGORITMO DI GOERTZEL.

Consideriamo la formula della trasformata discreta di Fourier (DFT) applicata ad un segnale $x(n)$

$$(1) \quad X(k) = \sum_{n=0}^{N-1} x(n) W_N^{kn}$$

dove $W_N^{kn} = e^{-j2\pi nk/N}$. Dato che $W_N^{-kn} = e^{+j2\pi nk/N} = 1$, possiamo moltiplicare la (1) per W_N^{-kn} senza alterarne il valore:

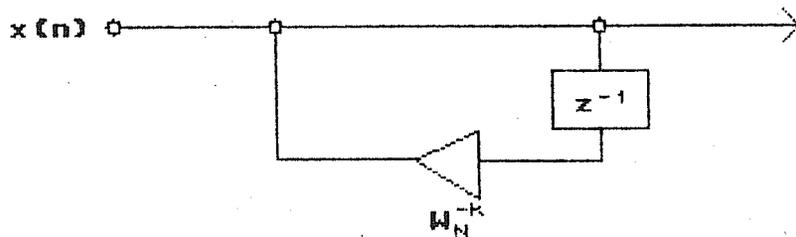
$$(2) \quad X(k) = W_N^{-kN} \sum_{r=0}^{N-1} x(r) W_N^{kr} = \sum_{r=0}^{N-1} x(r) W_N^{-k(N-r)}$$

Introduciamo, ora, una sequenza che potremmo definire "integrale":

$$Y_k(k) = \sum_{r=0}^{N-1} x(r) W_N^{-k(N-r)}$$

Dal punto di vista analitico, k esprime la k -esima riga dello spettro e l'indice N è legato al numero dei punti in esame.

La (2) descrive una convoluzione fra due sequenze, $x(r)$ e W_N^{-kn} . Data la sua definizione, W_N^{-kn} può essere considerata una risposta impulsiva. Il diagramma di flusso di un sistema con risposta impulsiva è:



e la funzione di trasferimento che ne risulta è:

$$(3) \quad H_k(z) = \frac{1 - W_N^k z^{-1}}{1 - 2 \cos(2\pi k/N) z^{-1} + z^{-2}}$$

Nella pratica, si applica l'equazione differenza corrispondente considerando il termine complesso $W_N^k z^{-1}$ solo alla N -esima iterazione. Come si può notare, la selettività del filtro corrispondente cresce col crescere di N (i.e. le righe spettrali diventano più strette).

Risolvere, quindi, M equazioni come la (3) corrisponde alla implementazione di un banco di filtri centrati sulle frequenze desiderate. La valutazione della grandezza di N è quindi critica perché da essa dipendono da un lato la selettività dell'analisi e dall'altro il tempo di calcolo necessario all'algoritmo per ogni iterazione di n campioni.

Si tratta quindi di risolvere l'equazione

$$Y_k(n) = x(n) + C_k Y_k(n-1) - Y_k(n-2) \quad (\text{dove } C_k = 2 \cos(2\pi k/N))$$

per n-1 iterazioni, indi

$$Y_k(N) = x(N) + C_k Y_k(N-1) - Y_k(N-2) - x(n-1) W_N^n$$

per la N-esima iterazione.

4. REALIZZAZIONE SU *HARDWARE* SPECIFICO.

Nella realizzazione dell'algoritmo di Goertzel in tempo reale è indispensabile valutare il carico di calcolo che comporta ciascuna trasformata tenendo conto dell'elaboratore di segnali impiegato.

Impiegando un elaboratore della famiglia *TMS32010* e dovendo realizzare un filtro che richiede fondamentalmente una moltiplicazione e due somme (a parte il calcolo dell'ultima moltiplicazione complessa):

il cui microprogramma è:

ZAC	; zero accumulator
SUB Y2, 1S	; subtract third term into accumulator
LTD Y1	; load t register of multiplier
MPY C	; multiply Y1 by C
APAC	; accumulata result in accumulator
ADD X, 1S	; add first term
SACH Y1, 1	; store result

si può notare che lo sviluppo dell'algoritmo richiede 1.4 usec per ogni campione. A frequenza di campionamento 8 kHz (frequenza peraltro imposta dall'uso del *combo codec*, e comunque sufficiente per l'applicazione specifica) il periodo di campionamento è di 125 usec: sarebbe quindi possibile calcolare circa 89 filtri (7 ottave). Tenendo conto, tuttavia, dell'ultima moltiplicazione complessa e dei tempi di calcolo necessari ai test logici conviene limitare il numero dei filtri a 36 (= 3 ottave). Questa limitazione è legata, oltretutto, ai limiti di memoria interna dell'*hardware* utilizzato.

Ad ogni modo, il valore in frequenza è pronto ad essere inviato via MIDI dopo circa 1,5 msec.

5. ANALISI DEI RISULTATI.

Optando per filtraggi non troppo selettivi (n. piuttosto ridotto), siamo riusciti ad ovviare

al problema postoci da strumenti scordati o non perfettamente intonati.

Nella Fig. 1 è illustrato il risultato spettrale dovuto ad a) una frequenza centrata, b) una frequenza calante e c) una frequenza crescente.

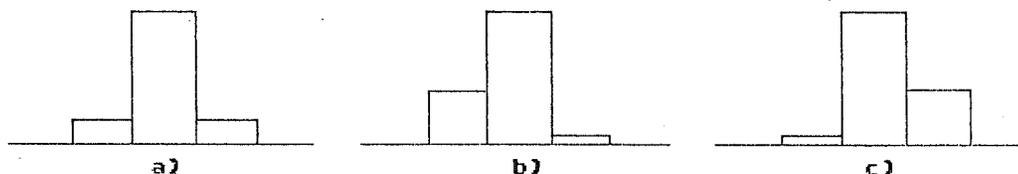


Fig. 1 -

Se i filtri sono poco selettivi un segnale di una data frequenza farà sollevare, oltre alla riga corrispondente, anche le righe adiacenti, seppur in maniera ridotta. Dalle righe adiacenti è possibile determinare la relazione tra la frequenza in ingresso e le frequenze del banco, come si vede nella Fig. 1.

Dato che ognuna di queste situazioni corrisponde a precisi valori scalari di ciascuna riga, è possibile: a) rigenerare coefficienti del filtro in modo da realizzarne una forma di adattività al segnale, oppure b) generare codici MIDI appropriati alla modulazione sul sistema di sintesi (e.g. *pitch-bend*, ecc.).

5. PROSPETTIVE.

I risultati ottenuti lasciano presagire un sistema del costo estremamente contenuto (si tenga conto che il prezzo di listino del *TMS32017* si aggira intorno alle 30.000 lire) in grado di rispondere con efficacia alla maggior parte delle applicazioni musicali. Naturalmente, l'utilizzazione di sistemi più potenti (e.g. *TMS320c25*, ecc.) permetterebbe, a costi più elevati, un'analisi su più ottave. È comunque possibile montare sul sistema attuale un selettore manuale di ambito di funzionamento per ovviare in maniera empirica al problema dell'utilizzazione con strumenti diversi.

Il sistema si presta inoltre ad ulteriori ricerche riguardanti la rilevazione di note polifoniche e, con alcune limitazioni, il riconoscimento del timbro strumentale.

Bibliografia.

INTERNATIONAL MIDI USER'S GROUP (IMUG, a cura di) — *MIDI Standard Specification 1.0*. International MIDI Association, Los Altos, Ca. 94022 USA, 1983.

HESS W. — *Pitch Determination of Speech Signals - Algorithms and Devices*. Springer-Verlag, Berlin New York, 1983.

GOERTZEL G. — *An Algorithm for the Evaluation of Finite Trigonometric Series*. Amer. Math. Monthly, vol. 65, January 1958, pp. 34-35.

GOERTZEL G., TRALLI N. — *Some Mathematical Methods of Physics*. McGraw-Hill, New York, 1960.

OPPENHEIM A.V., SCHAFFER R.W. — *Digital Signal Processing*. Prentice-Hall, Englewood cliffe N.J; 1975, pp. 287-289.

APPLICAZIONI DELLA TECNICA DI DISTORSIONE NON LINEARE A FUNZIONI DI INGRESSO IN REGIME NON SINUSOIDALE

E. Giordani, S. Gabrielli
ISELQUI - Ancona

Introduzione.

L'impiego della tecnica di sintesi nota con il nome di distorsione non lineare (NLD) presuppone implicitamente che, una volta definita la funzione di eccitazione generalmente un segnale sinusoidale del tipo $I(t) = a(t) * \cos(w*t)$, esista una classe di funzioni tali che una loro combinazione lineare sia in grado di produrre una serie di righe spettrali il cui numero e peso è determinato dal valore del parametro a . La classe di funzioni generalmente adottate sono i "polinomi di Chebychev".

Viene descritta in questa sede una possibile applicazione della NLD a segnali periodici del tipo:

$$s(t) = a(t) * \sum_i A_i * \cos(i * w * t) \quad \text{per } i=1,2,3,\dots,N \quad (0)$$

dove $a(t)$ è una funzione lentamente variabile.

In questo contesto la NLD viene vista come una tecnica di modificazione di suoni periodici il cui contenuto spettrale viene variato e ampliato attraverso una trasferimento non lineare in modo tale da poter simulare, almeno in primissima approssimazione, la variazione di timbro che si produce nei suoni strumentali al variare dell'energia fornita dall'esterno.

Questo processo è stato inserito in uno dei blocchi funzionali di un circuito integrato (sviluppati all'Iselqui) per applicazioni in sistemi musicali di basso costo.

Esc.

Il dispositivo (Economical Sampling Chip), in tecnologia CMOS 1.5um e un package di tipo 64 pin PLCC è stato progettato per svolgere le seguenti funzioni:

- 1) gestire il sampling di segnali audio e relativa memorizzazione su RAM esterna con codifica ADPCM;
- 2) gestire il de-sampling di 12 canali indipendenti;
- 3) realizzare la decodifica ADPCM dei campioni estratti;
- 4) eseguire il prodotto tra una funzione di inviluppo e la funzione campionata e contemporaneamente distorcere in ampiezza tale prodotto attraverso una trasferimento non lineare.

L'introduzione di spettrovarianza è ottenuta utilizzando funzioni distorcenti tabulate in ROM interna con possibilità di selezionare 4 diverse curve di cui una semplicemente il by-pass del blocco.

Dei 12 canali, 8 sono caratterizzati da una frequenza di campionamento variabile nel range 10-40 KHz con programmazione da uP e con risoluzione migliore dell'1% a 40 KHz. I cam-

pioni possono essere letti in "single mode" oppure in "loop mode" e la dimensione del blocco di memoria che contiene i campioni del loop può essere qualunque.

Blocco di distorsione.

I campioni in ingresso al blocco distorcente presentano una dinamica a 12 bit segnati. Con questa codifica, ogni funzione distorcente dovrebbe essere formata di 4096 valori per un totale di 12K valori per le 3 funzioni.

Una riduzione drastica del 50% si è ottenuta decidendo di impiegare solo trasferenze antisimmetriche in unione ad un complementatore mentre una ulteriore riduzione si è ottenuta con un accesso algoritmico in ROM e rappresentando le curve con spezzate lineari.

Il dato a 11 bit viene ripartito in due valori, rispettivamente di 7 e di 4 bit ciascuno. I primi 7 bit sono impiegati per accedere ai 128 dati che rappresentano i valori grossolani della funzione distorcente mentre i restanti 4 bit sono impiegati per accedere a 16 tabelle relative ad altrettante pendenze tabulate in una ROM 256×8 .

Sono state prese in considerazione funzioni distorcenti di tipo:

$$F(X) = -F(-X)$$

e non è prevista alcuna azione di normalizzazione per l'ampiezza. Inoltre la pendenza massima delle funzioni rappresentate deve essere:

$$0 < = D_{max} < = 16 \text{ (solo pendenze positive)}$$

oppure

$$-8 < = D_{max} < = 7 \text{ (pendenze positive e negative)}$$

Si è infine ipotizzato che le funzioni distorcenti rispecchino una caratteristica linearizzante nell'intorno dell'origine per consentire di trasferire all'uscita, per piccoli valori dell'inviluppo d'ampiezza, il segnale di ingresso praticamente indistorto.

Simulazione e test sperimentali.

I test sono stati eseguiti su una classe di funzioni di ingresso costituite dalla somma di N componenti armoniche ed equipotenti del tipo:

$$i(t) = (1/N) * \sum_i \cos(i * w * t) \quad (1)$$

e con classi di funzioni definite da n

- 1) polinomi del tipo $F(X) = X^n$
- 2) polinomi di Chebychev (T2, T3,...)

Per tutti i casi considerati la funzione di uscita risulta un segnale il cui numero di righe è pari a $n * n$ (per $N > 1$) dove si è indicato con n il grado della polinomiale ed N il numero di righe del segnale di ingresso.

Impiegando ad esempio polinomi del tipo (1) con $n = 2$ lo spettro del segnale di uscita $O(t)$ è costituito (qualunque sia il numero di componenti in ingresso) da una porzione che chiamiamo SBM (spettro base modificato) e da una porzione che chiamiamo SA (spettro addizionale).

La componente a frequenza maggiore di SBM (che coincide in frequenza con la più acuta di $i(t)$) è sempre attenuata del 50% (6 dB) rispetto alla fondamentale di $0(t)$.

Si può verificare facilmente che in queste condizioni il valore della singola componente Akdb della porzione SMB di $0(t)$ risulta:

$$Akdb = 20 \cdot \log(A1/2 \cdot (1 + (N-K)/N-1)) \quad (2)$$

con $K=1,2,\dots N$ e dove $A1 = N-1/M \cdot N$

Lo spettro addizionale SA è invece una replica di SBM scalata cioè:

$$Bkdb = 20 \cdot \log(A1/2 \cdot (1 + (N-H)/N-1 + A1-AN)) \quad (3)$$

con $H = N+1, N+2, \dots (n \cdot N)$

Sostanzialmente si individua una azione di tipo passa-basso sul segnale originale ed una contemporanea crescita spettrale delle componenti a frequenza maggiore della più acuta presente nello spettro base con un andamento non monotono nello spettro complessivo.

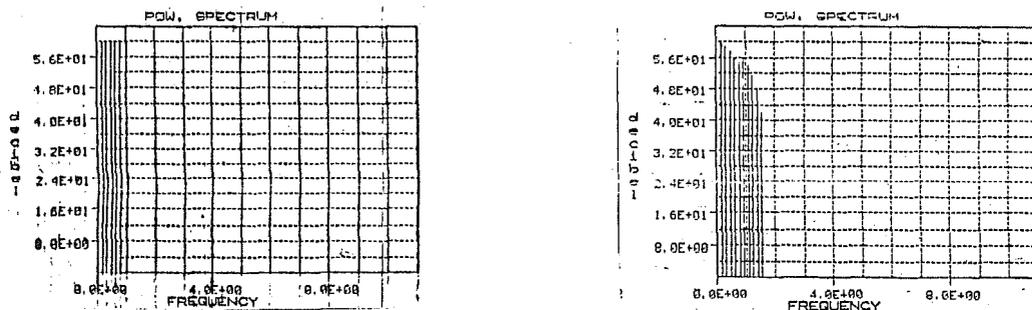


Fig. 1

Nel caso in cui $n=3$ ed N qualunque si ottiene egualmente uno shaping spettrale di tipo passa-basso ma con un andamento di tipo monotono.

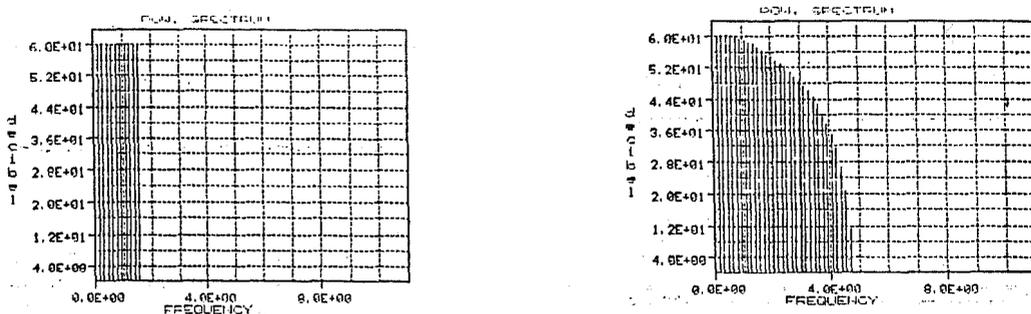


Fig. 2

Nel caso vengano impiegati i polinomi di Chebychev si osserva che quelli di ordine pari (T2, T4,..) producono un'azione di tipo passa-basso sullo spettro base mentre le dispari (T3, T5,..) una azione di tipo passa-alto.

Inoltre i polinomi di ordine pari presentano un numero di risonanze uguale al numero di antirisonanze mentre quelli di ordine dispari una risonanza in più rispetto alle antirisonanze.

Nelle figure 3a/b sono mostrati gli andamenti relativi allo shaping spettrale prodotti da T4 e T5 su un segnale di tipo (1) e con N=9.

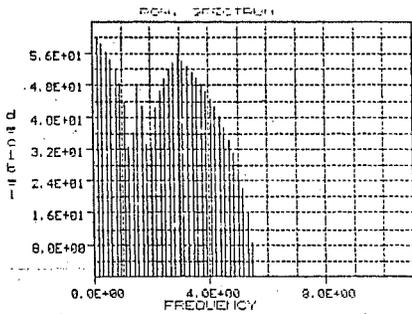


Fig. 3a

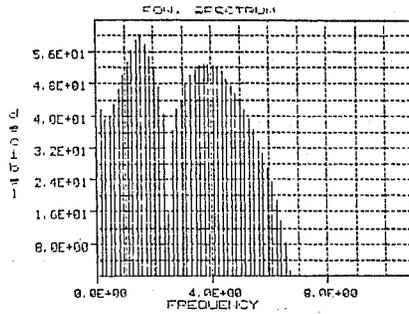


Fig. 3b

Nella figura 4 invece viene mostrato in rappresentazione tridimensionale (ampiezza, tempo, frequenza) l'andamento temporale delle componenti di un segnale di tipo (1) con N=5 involupato con una funzione triangolare (a(t) compresa tra 0 e 1) e distorto con una funzione puramente cubica.

Si può notare la perdita di linearità e i ritardi di attivazione delle singole componenti. Il punto di massima ampiezza corrisponde ad $a(t) = 1$.

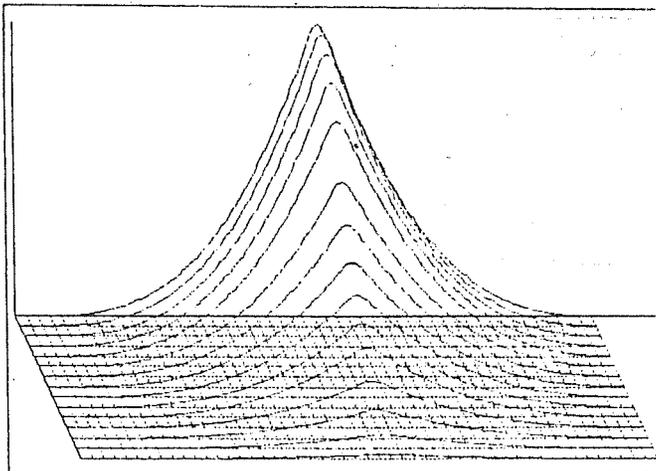


Fig. 4

Conclusioni

L'applicazione di funzioni distorcenti a segnali non sinusoidali si rivela un campo di indubbio interesse, ma la portata dei risvolti matematici che si celano dietro questo processo non ci hanno permesso, allo stato attuale, di formulare una ricetta semplice per ottenere risultati predicibili in modo puntuale ma solo considerazioni intuitive su test per lo più eseguiti su base empirica.

Tuttavia non si può certo negare che l'approccio non rigoroso riveli aspetti di fondamentale importanza come ad esempio il mantenimento delle caratteristiche spettrali salienti dei segnali di stimolo.

Se si considera ad esempio il caso tipico del clarinetto, il cui spettro medio presenta una predominanza di componenti dispari, il processo di distorsione introduce righe supplementari pur mantenendo tale caratteristica, almeno qualitativamente sullo spettro base.

Riteniamo quindi aperta la eventualità in futuro di estendere in un costesto di maggior formalizzazione le possibilità offerte dalla NLD applicata a segnali non sinusoidali anche rimuovendo alcune limitazioni imposte dal progetto specifico.

Bibliografia

- 1). *Digital Waveshaping Synthesis* M. Le Brun JAES 1979 Vol. 27 N. 4
- 2). *Digital Synthesis of Complex Spectra by Means of Multiplication of Non-linear Distorted Sine Waves* D. Arfib JAES 1979 Vol. 27 N. 10
- 3). *A Tutorial on Non-Linear Distortion or Waveshaping Synthesis* C. Road CMJ 1979 Vol. 3 N. 2
- 4). Rand Corp. *REDUCE* Reference Manual Santa Monica & University of Utah
- 5). Mit - Ems *CSOUND* Reference Manual MIT
- 6). *Handbook of Mathematical Functions* A. Abramowitz Dover Publ. NY

SINTESI PER FORMANTI: CONTROLLO MEDIANTE MODULAZIONE DI FASE

S. Cavaliere, I. Ortosecco, A. Piccialli

Dicembre 1987

Dipartimento di Scienze fisiche - Mostra D'Oltremare pad. 20

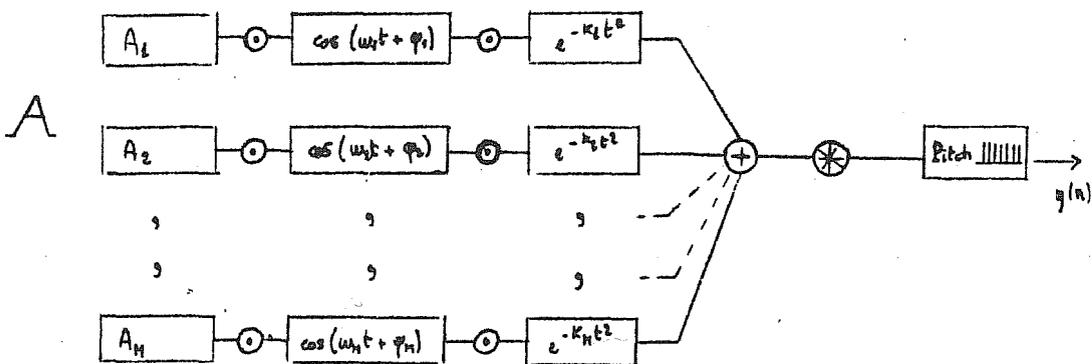
80125 Napoli

1. Sintesi per formanti mediante filtri FIR

Recentemente è stata sviluppata ([1], [2]) una tecnica di sintesi per formanti mediante filtri FIR a fase lineare. Il modello consiste in una struttura parallela di M filtri FIR passa-banda (dove M è il numero delle formanti), centrati sulle frequenze di risonanza f_i . Questi filtri sono ottenuti da filtri passabasso simmetrici, generalmente di tipo gaussiano, nella versione shiftata nel dominio delle frequenze: la risposta impulsiva dell' i -mo filtro assume la forma:

$$h_i(n) = a_i(n) \exp^{-k_i n^2} \cos(2\pi n f_i) \quad -\frac{N_0}{2} < n < \frac{N_0}{2} \quad (1)$$

dove k_i controlla la banda passante, f_i la pulsazione di risonanza ed a_i l'ampiezza della i -ma formante. La convoluzione della risposta impulsiva totale con un treno di impulsi di eccitazione, computazionalmente efficiente per via dei campioni nulli presenti tra un impulso ed il successivo, fornisce il segnale di uscita. Il controllo della dinamica del sistema di sintesi è assicurato dall'aggiornamento della a_i , K_i e delle f_i , generalmente sincronamente con il



periodo di pitch o un suo multiplo intero. In fig. 1 è mostrata una possibile implementazione dell'algoritmo: viene costruita la risposta impulsiva totale degli M filtri di eguale durata N

$$h(n) = \sum_{i=1}^M a_i(n) \exp^{-k_i n^2} \cos(2\pi n f_i) \quad (2)$$

e ne viene fatta la convoluzione con un treno di impulsi equispaziati.

In particolare, nel modello in esame, il controllo della banda può essere ottenuto cambiando il passo di lettura della tabella contenente la risposta impulsiva del filtro passa basso.

2. Controllo mediante distorsione di fase

Viene proposta qui una diversa implementazione dell'algoritmo, ricorrendo ad un controllo della banda e del piedistallo mediante distorsione di fase delle funzioni trigonometriche nella formula (2). La risposta impulsiva totale assume la forma più compatta

$$h(n) = \exp^{-k_0 n^2} \sum_{i=1}^M a_i(n) \cos(2\pi n f_i + \phi_i(n)) \quad (3)$$

dove $h_0(n) = \exp^{-k_0 n^2}$ è la risposta impulsiva di un filtro passa basso prototipo di durata fissa e la sommatoria è una sintesi additiva di M cosinusoidi con distorsione di fase, cui è devoluto il controllo dinamico dei parametri delle formanti. Anche in questo caso l'aggiornamento discreto delle a_i ed f_i è sincrono con il pitch o un suo multiplo intero. Il rumore introdotto risulta accettabile purché le variazioni dei parametri siano lente rispetto alla durata della risposta impulsiva.

Soffermiamoci adesso sulle caratteristiche del termine di fase $\phi_i(n)$. Una considerazione va fatta circa la simmetria della funzione: se $\phi_i(n)$ è pari, la frequenza istantanea sarà una funzione dispari e, in tali condizioni, non sarà possibile conservare la parità della funzione $h(n)$. Questa condizione di asimmetria crea qualche problema per la conservazione della linearità di fase del filtro. Nel caso di funzioni di fase dispari viene conservata la linearità di fase del filtro, ma nel dominio della frequenza le parziali risultano non simmetriche rispetto ad f_i ed in particolare su una delle bande esse presentano segni alternati. Per questo motivo quest'ultimo tipo di funzione di fase non risulta utile al nostro scopo e prenderemo in esame alcune funzioni di fase di tipo pari e di frequenza di tipo dispari. Adoperando un approccio tipico della modulazione in frequenza, viene introdotto un indice di profondità di distorsione I pari al rapporto tra deviazione massima in frequenza e frequenza della portante:

$$h_i(n) = h_0(n) \cos(2\pi n f_i + I_i \phi(n)) \quad -\frac{N_0}{2} \leq n \leq \frac{N_0}{2} \quad (4)$$

Come vedremo piccoli indici I_i sono sufficienti a controllare la banda.

3. Alcuni risultati

Alcune funzioni di fase sono risultate idonee per il controllo della banda e del piedistallo delle singole strutture formantiche. Sono qui forniti i risultati di due funzioni di fase particolarmente semplici da implementare. Le funzioni in esame sono:

$$\begin{aligned} 1) \quad \phi_a(n) &= k \left(n - \frac{N_0}{2} \right)^2 \quad -\frac{N_0}{2} \leq n \leq \frac{N_0}{2} \\ 2) \quad \phi_b(n) &= \exp^{-k \left(n - \frac{N_0}{2} \right)^2} \quad -\frac{N_0}{2} \leq n \leq \frac{N_0}{2} \end{aligned}$$

Ambedue le $\phi(n)$, come si è detto, introducono una certa asimmetria nella risposta impulsiva, con conseguente perdita della linearità di fase: piccole correzioni però, della fase ini-

ziale, rendono controllabile l'overlap delle formanti nel dominio della frequenza.

La funzione quadratica 1) permette agevolmente il controllo della banda passante operando sulla pendenza della funzione di frequenza, con risultati mostrati in fig. 3 e fig. 4 (in fig. 2 è mostrato lo spettro senza distorsione).

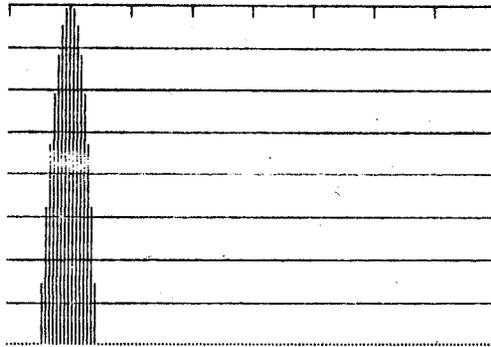


Fig. 2

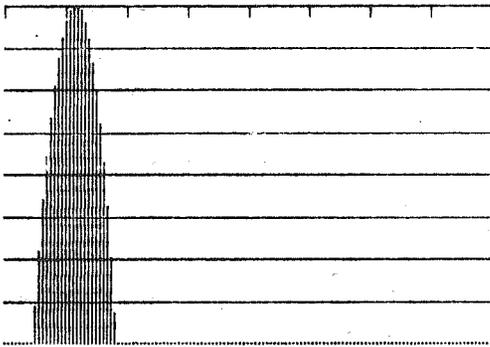


Fig. 3

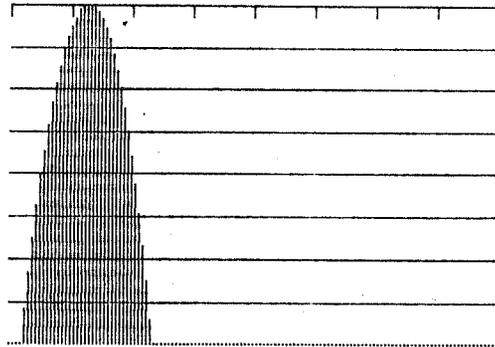


Fig. 4

Nell'esempio successivo (fig. 5 e 6) è mostrato il comportamento dell'overlap di due formanti attraverso il controllo della fase iniziale.

La funzione 2) (gaussiana troncata) si presta ad allargare sia la banda che il piedistallo. Nelle fig. 7 ed 8 è mostrato il controllo del piedistallo ottenuto con $k = 32$ e con indici di modulazione 0.5 ed 1.

Viceversa, se la costante k viene modificata ($k = 100$), con indici di modulazione 10 e 20 si ottiene il controllo della banda (fig. 9 e 10).

In conclusione il metodo è del tutto generale e piccole distorsioni di fase permettono di controllare la forma delle strutture formantiche.

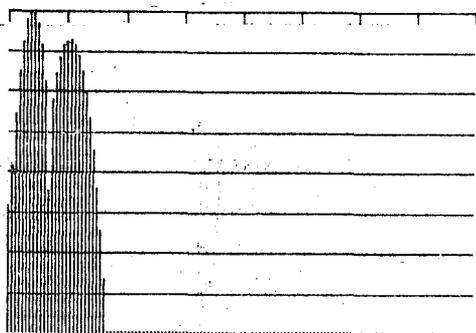


Fig. 5

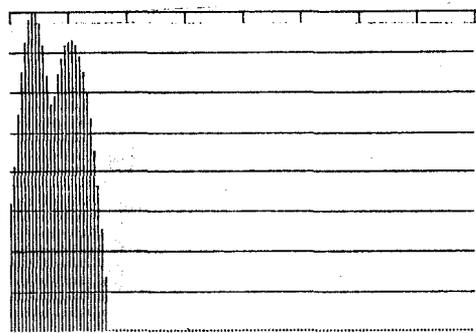


Fig. 6

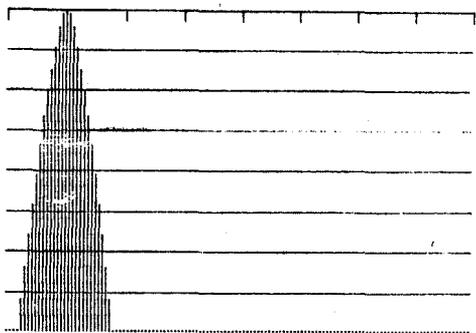


Fig. 7

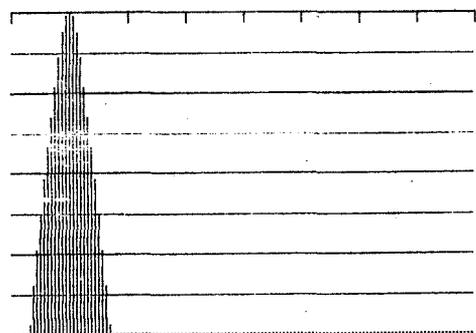


Fig. 8

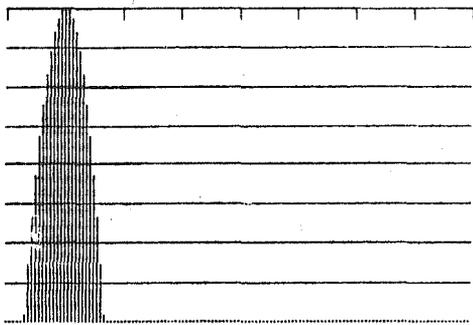


Fig. 9

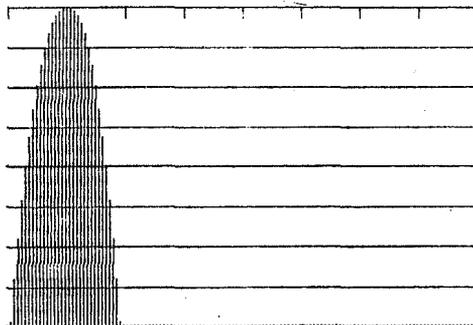


Fig. 10

Bibliografia

1). S. Cavaliere, I. Ortosecco, A. Piccialli. *Synthesis by formants: a new approach* 1986 IEEE Workshop on applications of Signal Processing to Audio and Acoustics.

2). G. De Poli, A. Piccialli. *Synthesis by Formants*. Sottoposto per la pubblicazione su Signal Processing.

SESSIONE APPLICAZIONI ALLA MUSICOLOGIA

PERFORMANCE RULES FOR COMPUTER MUSIC

Anders Friberg *, Lars Frydén **, Lars-Gunnar Bodin *** and Johan Sundberg *

* Royal Institute of Technology

Department of Speech Communication and Music Acoustics

S-100 44 Stockholm, Sweden

** Conservatory of the Swedish Radio

Edsberg, Sweden

*** EMS - Söder Mälarstrand 61 - Stockholm, Sweden

Introduction

Music composed by algorithms and performed by computers sometimes sound dull and lifeless, presumably depending on the absence of the many long- and short-term expressive variations normally produced by a human performer. While long-term events can be introduced in the performance in many ways, e.g. by means of hand movements recorded by various devices, well-controlled short-term events are more difficult to introduce. In this article we will show that both long- and short-term rules developed for the performance of traditional, tonal music can be used also, with slight modifications, for improving the performance of contemporary, atonal music.

The starting point was a rule system which automatically converts a music score to the equivalent sound sequences (Sundberg, & Frydén, 1984). It contains a number of performance rules, which, alone or in combination, have been found to improve the musical quality of the performance (Thompson & al., 1986). Originally developed for one-voice melodies the rules have recently been modified so that also polyphonic music can be automatically performed (Sundberg & al., 1987).

The purpose of the present investigation was to explore the possibilities of using these performance rules in the novel framework of non-tonal music either written to be performed by musicians or generated and performed by computer. The presence of an underlying tonal harmony in the traditional sense is missing, these rules had to be reformulated. Thus a modification of these rules was required.

It could be assumed that the rules would work better with music intended to be performed by musicians on traditional instruments, since the rules were developed for this kind of music. Computer music, by contrast, is often not intended to be performed in this way. Therefore, it was considered interesting to try the rules on both these types of music.

Rules

A group of five rules, all with contexts consisting of a few notes only, were selected from the performance rule system. In general, these rules were assumed to be the strongest candidates for this music. They operated on the amplitude and duration parameters and were thus directly applicable to piano performance. In addition, as a substitute for the rule reflecting harmonic progressions in traditional music a new rule was formulated, which worked with a longer context.

A subset of six rules was used in the experiment;

- 1) *The shorter the softer* shortens the notes in proportion to their duration.
- 2) *Tempo increase in ascending melodic motion* shortens notes occurring in a sequence of rising pitch intervals. The effect of the rule is to make the shortened notes sound as if they aim at the "target" note terminating the ascending motion.
- 3) *The shorter the shorter* increases the contrast in duration by making short notes relatively shorter.
- 4) *Lengthening of shorter note at halved duration* reduces the contrast in duration in cases when a short note is preceded by a note of twice its duration and followed by a longer note.
- 5) *Lengthening of extremely short notes* adds duration to very short notes surrounded by longer notes whereby the preceding note is shortened by the same amount.
- 6) *Chromatic-charge* is an equivalent to the melodic and harmonic charge previously used for tonal music. Each note is assigned a charge value that is inversely proportional to the note's chromatic distance to following note, if needed transposed to the same octave. Then, a mean value filtering with a window containing five notes is applied. The resulting average is then used for increasing the sound levels and durations of the notes. In this way, the rule generates crescendos and diminuendos.
- 7) *Micropauses in leaps* inserts very short silent intervals in melodic leaps. The duration of these micropauses are proportional to the width of the leap.
- 8) *Synchronization of voices* applies in compositions containing more than one voice. The synchronization is achieved by either synchronizing all notes occurring on the first beat in each bar or, better, by devising a synchronization voice, constituted by the shortest notes which occurs, at each time, in the score (Sundberg & al., 1987).

In addition, a threshold of 50 msec was applied for a note's minimum duration. The duration of a note must surpass a minimum duration in order to be perceptible. This rule is similar to certain duration rules used in speech synthesis (Carlson & al., forthcoming). Also, an amplitude and duration normalization procedure was used keeping the average sound level and the total duration of the piece constant.

These rules were tested by the authors on various pieces of contemporary music. For reasons to be discussed later, the quantities by which the rules affected the performance had to be increased as compared to what normally applies to traditional music. The quantity of each rule was chosen depending, among other things, on the character of the piece. When the resulting performances seemed musically appropriate, a formal listening experiment was carried out, in which five subjects, all working professionally with this kind of music, were asked for their preference.

Listening experiment

As mentioned above, we used two different kinds of music; music composed for piano and computer-generated random music. Seven examples were selected (see Table I).

P. Boulez: First eight measures of Piano sonata.

A. Webern: Third piano variation op 27.

I. Xenakis: Extract from Herma.

WN+Q: White Noise frequency variation with +Quantized durations.

WN-Q: White Noise frequency variation with non-Quantized durations.

PN+Q: Pink Noise frequency variation with + Quantized durations.
PN-Q: Pink Noise frequency variation with non -Quantized durations.

Table I. Music excerpts selected for the listening experiment.

Random algorithms were used for composing the last four pieces in the Table. In the compositions with quantized durations only three different durations were permitted. In the compositions with non-quantized durations a total of 27 different durations were permitted and sequences of more than two notes of the same duration were not allowed. Models for the composing algorithms were taken from a survey by Dodge and Jerse (1985).

Three different setups for the quantity of each rule were used; one for the Webern piece, one for the pieces by Boulez and Xenakis, and one for the computer generated melodies. Most of the quantities of each rule were the same in all the rule setups. They were about 2 or 4 times their normal values previously used for traditional music. The largest difference between the setups occurred in rule 3 (the shorter the shorter) for the Webern music, where the quantity was increased eight times as compared to the other setups.

The music examples were arranged in pairs on a digital test tape. In each pair the same excerpt was played one time with and the other time without application of the rules. The order of pairs and the order within pairs was randomly selected. The task of the listeners was to select which performance they preferred in each pair. The subjects were asked to pay attention to the performance as a whole and to disregard single "bad" notes.

For the piano music examples it seemed essential to use a realistic sound that was easily associated with a human performer, while for the non-piano examples it appeared more logical to use a purely synthetic sound that was not associated with any conventional instrument. Therefore, the sound used for the first five excerpts was that of a sampled piano as produced by a Casio FZ1 sampler. For the last four examples a purely synthetic sound on a Yamaha FB01 synthesizer was used.

The pair-wise comparison was preferred, as it was considered essential not to raise too high demands on the subjects; a test where the musical examples appeared in a totally random order and where the subjects assessed the musical quality of the performance, would be more demanding.

The subjects were two professional pianists specialized on performance of contemporary music, and three composers of electro-acoustic music. They found the task fairly simple since, in most cases, they readily heard a difference between the two performances.

	<u>piano sound</u>	<u>with rules</u>	<u>without rules</u>
Results	Webern	4	1
	Boulez	4	1
	Xenakis	4	1
	WN-Q	5	0
	PN-Q	5	0
	<u>synth sound</u>		
	WN-Q	5	0
	PN-Q	4	1
	WN+Q	5	0
	PN+Q	5	0

Table II. The number of subjects preferring the versions played with or without application of the rules.

As can be seen from Table II, there is a clear preference for the rule-generated performances in all cases. The preference is slightly more evident for the random melodies. This is surprising in view of the fact that the rules were developed for music played on conventional instruments.

Discussion and Conclusions

The present investigation showed that performance rules developed for traditional tonal music improved the performance also of contemporary, atonal music. Only few rules could not be applied for various reasons. The revisions of the rules were small but probably important.

In atonal music, there are no chords in a traditional sense. As a consequence, the notions of melodic charge and harmonic charge become meaningless. In an early stage, we tried performances where the melodic and harmonic charge were simply omitted. These performances seemed to suffer from a lack of long-term events, and the music gave the impression of an aimless wandering. As soon as the chromatic charge was introduced, this disease was cured. We find it interesting that music performance seems to need events embracing, and thus marking, the formation of greater blocks.

In random generated music wide leaps are very likely to occur, while long sequences of small melodic intervals are rare. In traditional music, on the other hand, melodic movements along the scale are frequent. The new rule that increased the sound level in sequences of small intervals may serve the purpose of marking emphasis, which would be appropriate; in all types of communication it seems important to emphasize the unexpected elements (Carlson & al., 1988).

Another modification required for this application of the rules was that the quantities needed adjustment, as mentioned above. Most of the rules had to be exaggerated in order to produce the desired effect. Also, in some cases the amount of a rule had to be altered between different pieces. The reason for this is not known. However, the following explanations seem likely.

First, we used merely six rules while the complete rule system contains 14 main rules. When introducing a new rule in the system we have often observed that a reduction of the quantities for the existing rules became necessary in order to avoid exaggerated effects. A small number of rules would then entail the need of increasing the quantities for these rules.

Second, it is possible that contemporary music calls for a more overt marking of musical events than traditional music. For reasons of style, it would be more difficult to detect the musical structure in contemporary music than in traditional music. For example it would be more difficult to predict the continuation of a series of tones. It is possible that this leads to a need for increased quantities for the rules.

One of the reasons why the rules had a better effect on the random melodies may be that the pieces composed by a human composer already contains more of the special interesting quality that performance may enhance. In any event, a dead-pan performance of the humanly composed pieces seemed more acceptable than when the same type of performance was applied to the randomly composed pieces.

The present listening experiment was carried out with experts only, composers and pia-

nists. A possible question is to what extent the results hold true also for a typical audience at contemporary music concerts. Our assumption is that these two groups of listeners do not disagree to any greater extent; for example, it seems impossible that a successful pianist could maintain ideals not shared by his audience. The reason why we chose experts as listeners was to avoid the greater number of subjects.

In the experiment we have shown that the rules improved the performance. On the other hand our method does not allow us to conclude that our rules are optimal; for instance, we cannot exclude the possibility that a random distribution of expressive variations would not produce a comparable improvement of the performance. On the other hand, at an early stage during the development of the rules for marking chromatic charge we tried to make crescendos where the average chromatic charge increased. The musical effect of this was unsatisfactory. This speaks against the assumption that random distribution of crescendos is a realistic alternative.

It is an interesting finding that the same rules were successful in contemporary music as in traditional music. This supports the assumption that these rules are not style dependent, but may work even for other types of music. This is not to say neither that no other rules are needed for improving the rule system, nor that there are no style-dependent rules. The fact that the rules worked also in contemporary music suggests that they can be assumed to introduce qualities to the performance that have a more general value. However, the underlying mechanism is not well understood. It is possible that the effects of the rules introduce sound patterns that allude to extramusical frameworks of reference, leaning on the listeners experience of speech, motion etc.

References

- Carlson, R., Friberg, A., Frydén, L., Granström, B. and Sundberg, J. (1988, forthcoming). Music and speech performance: parallels and contrasts.
- Dodge, C. and Jerse, T.A. (1985). *Computer Music*, Schirmer Books.
- Sundberg, J. and Frydén, l. (1984). *Teaching a computer to play melodies musically*, Analytica, Festschrift for Ingemar Bengtsson, Publications issued by the Royal Swedish Academy of Music, Nr 47, 67-76.
- Sundberg, J., Friberg, A. and Frydén, L. (1987). *Rules for automatised performance of music*, Talk at the Second Conference on Science and Music, London, will be printed in proceedings in the Contemporary Music Review.
- Thompson, W. F., Friberg, A., Frydén, L. and Sundberg, J. (1986) *Evaluating Rules for the Synthetic Performance of Melodies*, Speech Transmission Laboratory Quarterly Progress and Status Report, 2-3/86, 27-44.

UN SOFTWARE PER LA GENERAZIONE AUTOMATICA E LO STUDIO DELLA MUSICA JAZZ

F. Giomi, M. Ligabue

Divisione Musicologica CNUCE/C.N.R. - Conservatorio di Musica "L. Cherubini"
P.zza delle Belle Arti 2 - Firenze

Il progetto "Jazz", sviluppatosi a partire dal 1985 in seno alla Divisione Musicologica del CNUCE (Istituto del C.N.R.), prevede lo sviluppo di un software tool per lo studio dell'improvvisazione musicale.

Gli obiettivi del package di programmi sono essenzialmente due: la generazione automatica di improvvisazioni armonico-melodiche e la creazione di una struttura atta a facilitare lo studio delle procedure sottostanti all'invenzione musicale estemporanea in campo jazzistico.

La metodologia d'indagine trova i suoi presupposti in campo semiotico-musicale utilizzando anche alcuni principi della linguistica trasformazionale. Si assume quindi come concetto base che la musica, nel nostro caso quella improvvisata, possa essere intesa come una sorta di linguaggio con proprie regole sintattiche e stilistiche comprensibili ed interpretabili da qualsiasi individuo acculturato ad un certo sistema musicale.

Abbiamo considerato la struttura del blues in 12 misure che ci è sembrata come una delle più rappresentative: è attestata nella tradizione jazzistica dalle sue origini fino ad oggi e presenta caratteristiche ben precise nel suo linguaggio armonico e melodico che permettono di individuarla linguisticamente e stilisticamente in maniera certa. Ulteriormente si è assunto per la nostra ricerca lo stile Be-Bop in cui il blues conosce oltre che particolare fortuna anche un notevole sviluppo armonico e melodico.

Il lavoro di spoglio dei testi ha permesso di determinare un particolare idioletto consentendo così di operare su di una situazione omogenea. L'indagine testuale, condotta a più livelli, ha inoltre consentito di elaborare una serie di regole sintattico-grammaticali che sono state poi formalizzate in strutture ad albero di carattere derivazionale. Tali strutture sono state implementate in programmi e sottoprogrammi.

Una volta determinato l'oggetto di indagine il progetto è stato suddiviso in due grandi sezioni, quella di generazione e quella di studio assistito interattivo a molteplici livelli.

Il lavoro di analisi ha permesso di elaborare un certo numero di strutture di 4 battute. Sintatticamente le varie possibilità di combinazione rilevate sono state controllate per mezzo di alberi derivazionali di carattere probabilistico.

L'opzione, in questa fase, per strutture di 4 battute è dovuta a motivi squisitamente stilistici; con questo metodo infatti si sono potute salvaguardare alcune caratteristiche dell'accompagnamento quali la ricorsività di certe figurazioni, la loro variazione, lo spostamento di accenti, tutti elementi caratteristici dello stile jazzistico.

Riguarda al metro l'analisi ha rilevato la netta predominanza del 4/4 con qualche attestazione saltuaria del 3/4. Rispetto all'armonia il blues si attesta storicamente come una progressione di accordi che conosce i suoi capisaldi nella successione I-IV-V spesso con cadenza plagale di ritorno sul primo. Questi elementi vengono elaborati, soprattutto dal be-bop in poi, attraverso un gioco di inserimenti e sostituzione che, rispettando i punti nodali del I, IV e V grado, tendono ad arricchire le capacità espressive del genere.

Durante lo spoglio dei testi sono state riscontrate tutta una serie di configurazioni possibili che, analizzate ed inventariate hanno permesso la costruzione di una struttura ad albero

che sintetizza, battuta per battuta, le varie soluzioni; di ogni accordo possibile è stata rilevata anche la sua incidenza probabilistica.

La fase seguente del progetto si articola sull'analisi della parte melodica che si presenta come una serie di altezze organizzate secondo uno schema ritmico; i primi due aspetti da prendere in considerazione sono quindi stati evidentemente proprio questi: ritmo e altezze.

Riguardo al primo il lavoro di analisi si è basato su un processo di segmentazione al fine di rilevare l'incidenza probabilistica delle varie figurazioni in modo analogo a come si era proceduto nella parte armonica.

Nel passaggio alla parte di programma la generazione della struttura ritmica sfrutta una sottodivisione in tre sezioni: la prima genera il frammento iniziale, la seconda il nucleo centrale e la terza il frammento finale.

Tutta la sezione presenta un livello di interazione con la parte che genera la struttura ritmica dell'armonia permettendo così di ottenere strutture omogenee e di rispettare l'iter creativo dell'improvvisatore, stimolato durante l'esecuzione dagli elementi sottostanti. Così il programma, al momento di generare la struttura ritmica della sezione melodica, è in grado di leggere la generazione armonica e adatta la seconda alla prima evitando procedimenti non corretti e stilisticamente non verificati.

Il processo creativo melodico si articola a tre livelli che chiameremo, per comodità, principali, anche se sono dati, ovviamente, tutti gli stadi intermedi. Il primo vede un'improvvisazione che si sviluppa dall'inizio alla fine della progressione armonica su di un'unica scala; il secondo livello il passaggio attraverso le varie funzioni armoniche presenti nella progressione; il terzo, senz'altro il più evoluto, basa la costruzione dell'assolo sull'intero percorso armonico facendo corrispondere ad ogni accordo un'appropriata famiglia di scale.

Il processo di generazione è stato messo in grado di scegliere fra ognuno dei tre livelli descritti.

L'interazione fra le due parti del programma, armonica e melodica, è estesa: la seconda legge, una volta generato, tutto il percorso della prima e solo dopo procede su base probabilistica nella determinazione dell'andamento melodico. A questo punto per ogni valore della generazione ritmica viene assegnata un'altezza, completando così il lavoro di creazione melodica.

Tutto il lavoro di analisi, come in parte accennato nell'introduzione, è stato quindi finalizzato alla creazione di una competenza di base per la generazione automatica, intesa quest'ultima come mezzo di verifica della correttezza dei presupposti teorici.

Tale bagaglio di competenze viene quindi reso disponibile al possibile fruitore per mezzo della seconda sezione del progetto, quella che per comodità abbiamo chiamato di studio.

L'utente trova innanzi a sé, comunque un unico "oggetto": deve solo stabilire se delegargli completamente le scelte, svolgendo quindi una funzione passiva, o se interagire con la macchina. Al pari di un qualunque improvvisatore è in grado di definire l'andamento del brano, il metro e la tonalità; può anche determinare il numero di progressioni che desidera vengano eseguite e decidere per una generazione completa, sia armonica che melodica, o solo parziale.

La parte più interessante è però quella che consente tramite un ridotto e pratico set di comandi di determinare, parzialmente o in toto, tanto il percorso armonico che quello melodico. È permesso operare con elementi precostituiti, per comprensibili motivi di praticità, come organizzare a piacimento strutture proprie.

Il software tool è realizzato nel linguaggio di programmazione VS-FORTRAN (normalmente implementato su hardware IBM).

È strutturalmente diviso in due sezioni principali (Armonica e Melodica), che a loro volta sono organizzate in sezioni secondarie (Ritmico-Armonica, Armonica, Variazioni-Armoniche, Ritmico-Melodica, Melodica, ecc.).

In accordo al sistema di regole e ai presupposti teorici che stanno alla base della generazione, le scelte richieste in ogni sezione vengono fatte in funzione di alberi n-ari asimmetrici di probabilità che, come visto, rispettano i criteri di derivazionalità imposti dall'analisi del problema.

Sezione Ritmico-Armonica.

Provvede alla costruzione dell'intero percorso ritmico attraverso la scelta tra gruppi ritmici di 4 battute, immagazzinati e organizzati per mezzo di una apposita subroutine.

Considerando anche l'ipotesi di un eventuale inizio in levare dell'improvvisazione melodica una pausa di una battuta vuota viene anteposta alla prima progressione della parte armonica di ogni brano.

La sezione ritmico-armonica produce, una volta terminata, una sequenza ordinata di durate e influenza la successiva generazione del ritmo relativo alla parte melodica.

Sezione Armonica.

Per ognuna delle "n" progressioni viene costruita, seguendo l'omologo albero generatore, una sequenza diversa di 24 accordi in modo da far corrispondere due accordi per ogni battuta, in accordo alla figurazione ritmica prodotta per tale battuta.

Per consentire la generazione della cadenza finale vengono aggiunti alla ennesima progressione uno o due appositi accordi.

Tutti gli accordi prodotti da questa sezione sfruttano 4 delle 8 voci a disposizione.

Sezione Variazioni-Armoniche.

La sezione variazioni-armoniche permette la modifica della sequenza ordinata di accordi generata dalla sezione immediatamente precedente.

Si tratta di una sezione opzionale.

Essa è strettamente connessa alle specifiche definite dall'utente per modificare il percorso armonico attraverso il sottocomando "PATH". Come caso massimo avremo, secondo la sintassi, la specifica di un nuovo accordo per ogni semibattuta.

Sezione Ritmico-Melodica.

Costruisce l'intelaiatura ritmica necessaria per il conseguente sviluppo della melodia vera e propria. Interagisce con la sezione ritmico-armonica per consentire la dovuta omogeneità della generazione.

Tra i suoi parametri principali di ingresso c'è il valore metronomico: esiste una sostanziale differenziazione ritmica in funzione della velocità scelta.

La sezione ritmico-melodica opera su una suddivisione del pezzo in tre parti: inizio, corpo e finale.

Ognuna di queste parti ha il suo svolgimento su appositi alberi di scelta con chiara espansione o contrazione della parte centrale per l'adeguamento al numero di progressioni.

La scelta delle singole unità ritmiche avviene in un range di 128 diverse possibilità catalogate in una apposita subroutine-database.

Una volta terminata la sezione dovrà fornire alla successiva una sorta di mappa concer-

nente la disposizione di note, pause, gruppi, ecc. lungo tutto l'arco del pezzo, in modo da consentire la necessaria interrelazione tra ritmo e improvvisazione.

Sezione Melodica.

Costituisce la parte principale del programma: costruisce interamente lo sviluppo melodico dell'improvvisazione.

L'interazione è presente con tutte le sezioni precedenti, proprio per la sua collocazione al di sopra delle altre.

La scelta iniziale (effettuata all'inizio dell'esecuzione) viene fatta tra due possibilità:

1) Lettura dell'accordo relativo di ogni battuta attraverso una catalogazione costituita da due campi di cui uno contenente il grado dell'accordo e l'altro contenente il tipo dell'accordo. Conseguentemente avviene l'identificazione delle scale possibili che viene fatta in funzione dei due numeri letti e di un piccolo albero probabilistico di scelta.

2) Generazione delle note su scale appropriate (TTJZSM) con eventuale possibilità di spaziare sulle "funzioni" relative di ogni battuta. Questo significa che per ogni battuta avrò a disposizione alcune scale sempre corrette che potranno, all'occorrenza, essere trasposte per adeguarle alla funzione in corso nella battuta.

Per ogni battuta si ha così la definizione casuale-controllata di una scala, cioè di un certo numero di "note legali", sopra le quali potrà muoversi la linea melodica.

Di nota in nota viene scelto il tipo e l'andamento del percorso. Sono previsti: nota ribattuta, salto superiore, salto inferiore, scala superiore e scala inferiore.

Variazioni Melodiche.

Al contrario della sezione variazioni-armoniche, questa non corregge a posteriori il contenuto melodico ma è totalmente integrata nella sezione melodica in modo che per ogni nota possa essere considerato il caso, se presente, in cui è stata specificata una scala particolare per quella battuta. La nuova scala prevale su quella scelta immediatamente prima dal programma.

Attraverso il suo sottocomando "SCALE" si possono assegnare fino a un massimo di una scala per battuta (scelte tra le presenti costruite con la scala cromatica).

Output.

L'output può essere di due tipi: musicale con file di tipo Midi o grafico con tabulati facilmente traducibili in partiture tradizionali.

Si hanno diverse possibilità di interazione che vanno dalla semplice richiesta di una generazione completamente automatica fino all'esecuzione di sequenze totalmente assistite.

Riassumendo, quindi, l'obiettivo generale del progetto si pone nel campo di studio della musica improvvisata cercando di fornire un mezzo che sia il più adeguato possibile alle finalità richieste dall'apprendimento.

Ci sembra di poter affermare che il sistema presenta buone doti di praticità e specificità per lo studio dell'idioma jazzistico consentendo anche in questo settore lo sfruttamento delle vaste possibilità offerte dall'elaboratore.

Bibliografia.

BARONI M., JACOBONI C. — *Computer generation of melodies: further proposals*. Computer and the Humanities, 17, 1983.

CAMILLERI L. — *Un sistema di regole per la generazione di semplici melodie tonali*. Rapporto interno c84-11. Pisa: CNUCE/C.N.R., 1984.

CAMILLERI L. — *Linguistic Theory and music*. Proceedings of the International Symposium of Music Reason and Emotion, Gent: University of Gent, Communication and Cognition, 1985.

DE ROSE N. — *Tecnica dell'improvvisazione jazzistica*. Melodi, 1979.

DE ROSE N. — *Armonia e fraseggio jazz*. Milano: Melodi, 1982.

FRY C. — *Computer improvisation*. Computer Music Journal, 4, n. 3, 1980.

HODEIR A. — *Uomini e problemi del jazz*. Milano: Longanesi, 1958.

JONES K. — *Compositional applications of stochastic processes*. Computer Music Journal, 5, n. 2, 1981.

LERDAHL F., JACKENDOFF R. — *A generative theory of tonal music*. Cambridge Mass.: MIT Press, 1983.

LIGABUE M. — *A system of rules for computer improvisation*. Proceedings of ICMC 1986. The Hague: Royal Conservatory, 1986.

LUCCIO F. — *La struttura degli algoritmi*. Torino: Boringhieri, 1984.

MEHEGAN J. — *Jazz rhythm and the improvised line*. New York: Watson-Guption, 1962.

MEHEGAN J. — *Swing and early progressive piano styles*. New York: Watson-Guption, 1968.

MEHEGAN J. — *Tonal and rhythmic principles*. New York: Watson-Guption, 1984.

NATTIEZ J.J. — *Fondements d'une semiologie de la musique*. Paris: Union Générale d'Éditions, 1975.

NELSON O. — *Blues and the abstract truth*. Mark Music, (s.d.).

ZANO A. — *Mechanics of modern music*. Trad. it. Ancona: Berben, 1973.

FORMALIZZAZIONE DI UN PROCESSO COMPOSITIVO
Lo Studie 1 di Karlheinz Stockhausen

P. Furlani, P. Zavagna

Conservatorio "B. Marcello" di Venezia - C.S.C. dell'Università di Padova
Per gentile concessione di Campanotto Editore - Pasian di Prato - Udine

Il nostro lavoro consiste nella realizzazione all'elaboratore dello *Studie 1* di Karlheinz Stockhausen. Realizzazione che non si limita alla semplice ri-esecuzione del pezzo con l'ausilio della tecnologia informatica, bensì estende il processo di esecuzione alla fase di composizione vera e propria dell'opera. Si sono infatti introdotte nell'elaboratore le regole musicali fissate da Stockhausen generando così automaticamente anche la partitura. La scelta di quest'opera deriva da più motivazioni. Innanzitutto è la prima composizione di musica elettronica ad essere analizzata in dettaglio dall'autore e tale analisi è un'utile traccia per la stesura da immettere nell'elaboratore. Essendo uno studio, poi, non raggiunge elevati gradi di complessità compositiva e quindi risulta adatto per un primo contatto con il mondo della musica elettronica. Infine i materiali sonori elementari sono molto semplici (tutto il lavoro si basa su soli suoni sinusoidali), ma il processo di composizione è estremamente interessante sul piano didattico in quanto consente di estendere la medesima concezione strutturale dal microlivello sonoro al macrolivello formale.

Lo Studie 1 si colloca proprio all'inizio delle ricerche nel campo della musica elettronica. "A partire dal 1950 si cominciò a mettere in discussione tutto ciò che costituisce la musica europea [...], anche il materiale sonoro finora usato, i suoni stessi". Da questo presupposto nasceva la necessità di rivedere e scomporre il suono nelle sue componenti più elementari, per poi disporre di esse a piacimento, soprattutto per poterle inglobare in un processo compositivo, che tenesse conto di tutti, o quasi, i parametri che definiscono il suono. Si poté così inserire in quell'iter storico-compositivo, che, attraverso Schönberg e la dodecafonia, aveva portato allo strutturalismo integrale, anche il materiale "grezzo" per produrre i suoni: l'onda sinusoidale. È proprio attraverso la manipolazione di suoni sinusoidali prodotti da oscillatori che nasceranno i primi componimenti di musica elettronica, non solo di Stockhausen.

L'idea fondamentale che ha guidato l'autore nel suo lavoro è stata quella di serialità integrale, che porta ad una composizione rigorosamente strutturale di "tutti" i parametri che identificano il suono.

Il primo intervento "sul" suono che viene effettuato è quello di costruire un suono complesso, che non sia il prodotto di suoni armonici naturali. Avremo così gruppi di frequenze semplici (onde sinusoidali prodotte da oscillatori) moltiplicate fra loro secondo intervalli artificiali, imposti dal compositore. Da una frequenza originaria potremo così ottenere serie di gruppi di frequenze.

La serie che propone Stockhausen è la seguente:

(1) $5/12$, $5/4$, $5/8$, $12/5$, $4/5$.

"Ogni frequenza ottenuta mediante moltiplicazione è a sua volta il punto di partenza di una nuova serie". Studiare acustica "sarà indispensabile per tutti quei compositori che non si accontentano di accettare i fenomeni sonori come dati, ma che invece si oppongono alla dittatura del materiale e che vogliono, per quanto possibile, imporre le loro proprie concezioni formali ai suoni per arrivare ad una nuova corrispondenza tra materiale e forma: tra microstruttura acustica e macrostruttura musicale".

Il raggruppamento delle frequenze inizia la composizione; esso avviene estraendo progressivamente dalla tabella il numero delle frequenze ricavato dalla seguente serie:

(2) 4, 5, 3, 6, 2, 1.

A questo punto il materiale base della composizione, per quanto riguarda le altezze, è preparato; abbiamo cioè tutte le misture (i singoli "suoni", per così dire) che compongono il pezzo.

Il passo successivo riguarda le intensità da assegnare ad ogni frequenza. A questo proposito viene fatta una prima distinzione per quanto concerne i gruppi di suoni, le cui frequenze originarie "devono avere tutte la stessa importanza. Per questo hanno la stessa intensità". Posto il valore dell'intensità da assegnare alle suddette frequenze uguale a n , si procede sottraendo 4 dB ad ogni frequenza che segue finché non si è raggiunto il gruppo successivo.

La seconda distinzione da operare è quella che riguarda le misture, che sommano al valore di ampiezza precedentemente assegnato (assoluto) il valore corrispondente alla serie

(3) 3, 4, 2, 1, 6, 5,

partendo, per far questo, dalla frequenza iniziale di ogni mistura. I risultati da noi ottenuti seguono rigorosamente le istruzioni date da Stockhausen nella sua analisi; differiscono tuttavia dalle tabelle presenti nei testi che trattano dello *Studie I*. Siamo portati a credere che errori di stampa presenti nell'originale tedesco abbiano avuto come conseguenza alcuni malintesi. In realtà il semplice spostamento di un gruppo o di una mistura o addirittura di una frequenza può creare un errore difficilmente rilevabile a causa di altri errori conseguenti.

L'ultimo elemento da tener presente è infine il variare d'intensità ad ogni ripetizione della tabella. Poiché "nel corso del pezzo l'intera successione delle frequenze [...] viene utilizzata sei volte", ad ogni ripetizione la prima componente del gruppo avrà un valore diverso d'ampiezza. Per completare la composizione ci risulta invece che la tabella viene utilizzata più di sei volte; tuttavia, non esistendo la partitura completa del brano, ma avvalendoci solo dello schema formale, non ci è stato possibile verificare altrimenti i risultati ottenuti.

La serie

(4) 4, 2, 3, 5, 6, 1,

degli involuppi decide il tempo d'attacco e di decadimento di ogni mistura.

I linguaggi di compilazione di partiture in tempo differito come il MUSIC 5 o quello da noi adottato per l'*Apple* consentono di controllare l'involuppo tramite il tempo d'attacco, il tempo di decadimento e il modo di involuppo da associare ad ogni suono (nel nostro caso ad ogni mistura). Abbiamo cercato di rendere le indicazioni di Stockhausen nella maniera che ci avrebbe permesso di ottenere un risultato il più possibile simile alle intenzioni del compositore. All'ascolto si noterà immediatamente come le due versioni del brano differiscono: la nostra "esecuzione" spicca per la caratteristica "metallicità" e "asetticità" del suono.

Il problema di assegnare ad ogni frequenza una durata ha fatto sortire la necessità di controllare la partitura da un linguaggio specifico di compilazione come il MUSIC 5.

Nel nostro caso si tratta, da un lato, di dare alle frequenze delle durate "inversamente proporzionali alla loro altezza", dall'altro di trovare dei giusti rapporti tra la durata del suono e quella della pausa. Il primo viene risolto da Stockhausen prendendo il valore della frequenza in Hz e dividendolo per la velocità del nastro moltiplicata per dieci: il risultato sarà il valore in secondi della durata della frequenza; il secondo ricorrendo ancora ad una serie, che determinerà questa volta il rapporto intercorrente fra durata del suono e durata della pausa all'interno di ogni mistura. Si tenga presente che la mistura è un singolo suono (le frequenze sono sempre raggruppate verticalmente in essa), resta quindi da decidere quale dev'es-

sere la “durata superiore” (suono più pausa) di ogni mistura. Questa viene determinata dalla frequenza di ampiezza maggiore presente nella mistura. Avremo così, ad esempio, che la prima mistura, composta dalle quattro frequenze di valore 1920, 800, 1000, 625, durerà complessivamente 1920/762 secondi.

La serie

(5) 2/6, 4/6, 6/6, 3/6, 5/6, 1/6,

delle “durate” (rapporto suono/pausa) determina soltanto l’ordine nel quale le misture si susseguono, non determina invece se il suono deve precedere o seguire la pausa.

Per manipolare i dati ci siamo serviti di un file nel quale scrivere tutte le note (la partitura) della composizione. L’operazione viene svolta direttamente all’interno del programma, rendendo molto più semplice la gestione e il trattamento dei dati.

Per rendere più controllabili le misture e per poterle scrivere facilmente nel file abbiamo utilizzato una matrice nella quale riportare i dati di ogni struttura. Leggendo poi dalla matrice le caratteristiche delle varie sequenze risulta molto più agevole raggrupparle secondo le indicazioni date da Stockhausen per quanto riguarda il carattere delle strutture.

Il raggruppamento delle misture in sequenze, così come quello delle sequenze in struttura, segue i valori della serie (2).

Restano da decidere tre parametri fondamentali: il tipo di raggruppamento delle sequenze all’interno della struttura; il suo “carattere”, poiché la durata delle sequenze presenti in una struttura varia, si dovrà decidere anche come sarà l’inizio o il termine delle sequenze raggruppate verticalmente; infine la disposizione del rapporto suono/pausa all’interno delle misture. La serie del “carattere delle strutture” è uguale alla (4) degli involuppi.

L’ “eco di sequenza” è stato il penultimo gradino che abbiamo dovuto superare prima di affrontare l’ “assetto formale” nella sua completezza. La serie

(6) 2/6, 4/6, 6/6, 3/6, 5/6, 1/6,

decide la quantità della sequenza che deve venir ripetuta nell’eco, che, in quanto tale, ripete gli ultimi 2/6, 4/6, ecc., della sequenza stessa al termine della sua durata complessiva.

“Tutto quanto si è fin qui detto non rispecchiava che a metà la concezione del pezzo. Vale a dire che l’approssimazione dei valori limite era avvenuta solo in una direzione, verso lo zero. Ciò dipende dal fatto che le frequenze venivano prodotte solo verso il grave e che a esse corrispondono proporzionalmente durate e intensità.

La stessa cosa doveva avvenire anche verso l’acuto, verso l’ “infinito”. A questo scopo si procede ad un rispecchiamento di tutte le proporzioni: ogni struttura viene contrapposta a sé stessa in una trasposizione”. Gli intervalli di trasposizione seguono le proporzioni frequenziali originarie creando la seguente serie:

(7) 5/12, 5/4, 5/8, 12/5, 4/5, 8/5,

più 5 permutazioni di questa serie con i suoi stessi valori proporzionali. Dopo aver generato le prime 6 strutture principali e le loro trasposizioni, il programma genera altre 6 nuove strutture principali e, dopo averle permutate seguendo gli intervalli della serie (7), le loro trasposizioni. Questo per 6 volte (la prima più le 5 permutazioni) per un totale di 72 strutture (36 principali e 36 loro trasposizioni).

I vantaggi della ricostruzione all’elaboratore emergono quando, intervenendo direttamente sui parametri (sui dati, manipolabili facilmente nei programmi), possiamo, con lo stesso schema formale, la stessa rigorosa struttura usata da Stockhausen, creare altri “nuovi”

Studie 1.

L'utilità didattica, l'inserimento nell'ambito dell'analisi musicologica di una fetta della musica - quella elettronica - normalmente trascurata da questa disciplina e la ricerca di nuovi mezzi compositivi attraverso le forme del passato (anche se recente), sono fra i risultati che volevamo ottenere da questo lavoro, sperando di essere riusciti, almeno in parte, nel nostro intento.

Bibliografia.

FURLANI P., ZAVAGNA P. — *Uno studio musicale all'elaboratore*. Campanotto Editore, Pesian di Prato, Udine, 1988, (da cui è stato tratto l'intervento sopra riportato)..

STOCKHAUSEN — *Komposition 1953 Nr. 2*. In: Aa. Vv., *La musica elettronica*, Feltrinelli, Milano, 1976.

INFORMATICA E MUSICOLOGIA: UN SISTEMA PER IL RECUPERO DI "FONDI" MUSICALI

L. Finarelli - ENIDATA SpA
G. Haus - LIM-DSI - Università di Milano

Introduzione.

Presso le biblioteche dei conservatori e di altri enti giacciono moltissimi manoscritti musicali solo in parte schedati e in condizioni di conservazione spesso precarie, i quali, dopo un'opportuna selezione, possono essere recuperati e valorizzati.

Nella maggior parte dei casi, oltre alle difficoltà di consultazione, essi sono di difficile lettura ed esecuzione, sia a causa della notazione, sia perché esistono solo le parti staccate della musica polifonica.

Per il recupero di questi giacimenti musicali si è progettato, nell'ambito di un progetto pilota dell'ENIDATA SpA per il *Recupero e valorizzazione di giacimenti culturali in campo musicale*, un sistema con le seguenti funzioni:

- caricamento e archiviazione di singole parti;
- formattazione assistita da elaboratore della partitura;
- stampa ed edizione della partitura;
- stampa ed edizione delle singole parti;
- esecuzione MIDI (per verifica).

Il primo obiettivo di questo progetto è la realizzazione di una workstation per musicologi che lavorano su questi antichi manoscritti non pubblicati e necessitano di un sistema computerizzato, veloce e flessibile, per le loro ricerche e pubblicazioni. Questo sarebbe molto significativo considerando l'enorme lavoro necessario per realizzare studi musicologici e revisioni critiche; inoltre non bisogna dimenticare la competenza specialistica dei tipografi e degli incisori.

Descrizione del sistema.

Il sistema è stato sviluppato lavorando, insieme al M° Guido Salvetti, sul materiale presente nel "Fondo Nosedà", una ricca raccolta di testi musicali che si trova custodita all'interno della biblioteca del Conservatorio "G. Verdi" di Milano. In questa raccolta ci sono diverse migliaia di brani, molti dei quali sono interessanti pezzi antichi non pubblicati dei quali esistono solamente le singole parti manoscritte e quindi mai visti in partitura e mai eseguiti in tempi recenti.

Tra questi sono stati scelti alcuni brani significativi di compositori famosi, quali G. Paisiello o D. Cimarosa. La scelta è stata fatta prendendo in considerazione una prima ricostruzione parziale delle partiture selezionate, realizzata con il nostro sistema nel modo preview e stampata in bozza. Dopo questa prima valutazione delle partiture sono stati scelti i brani da ricostruire totalmente; questi sono stati completamente inseriti dalle singole parti manoscritte, lette e interpretate dal gruppo musicologico del Conservatorio guidato dal M° Salvetti, e se ne è costruita la partitura orchestrale rendendone così possibile lo studio, la pubblicazione

e l'esecuzione.

Attualmente il sistema è disegnato per brani del periodo 1750-1850 per piccola orchestra ma potrà facilmente essere adattato a qualsiasi tipo di musica scritta in notazione comune (su pentagramma).

Chiaramente questo sistema può essere utile per molti altri scopi, e non solo per uso musicologico, cioè in tutti quei casi in cui ci sia una pagina di musica da stampare o da visualizzare.

Il sistema che abbiamo realizzato cerca di soddisfare sia le richieste dell'editoria musicale, sia le esigenze di interazione tra la macchina e il musicista che si occupa dell'input e della elaborazione delle partiture. Per fare ciò dapprima ci siamo confrontati con la letteratura scientifica (1, 2, 3, 4, 5) e con i prodotti commerciali; poi abbiamo studiato come sviluppare l'architettura del sistema affinché fosse orientato ad attività musicologiche ed editoriali e aperto a integrazioni ed estensioni successive.

Inoltre noi vorremmo proporre questo progetto come uno degli anelli della catena di sistemi connessi da reti che lavorano nell'ambito dell'informatica musicale. Abbiamo infatti già realizzato un primo accordo preliminare con il Center for Computer Assisted Research in the Humanities (CCARH) per rendere compatibile il loro database (l'opera omnia di J.S. Bach e di A. Corelli) con il nostro sistema (6, 7). Se poi pensiamo a una codifica standard (ad esempio la codifica ANSI attualmente in via di definizione), o a una possibile trascodifica tra codifiche differenti, la nostra workstation si propone come quella specializzata nel trattamento degli aspetti editoriali e grafici dell'elaborazione dei testi musicali; mantenendo ben salda la notazione tradizionale, s'intende.

In molti casi, infatti, la notazione della musica su carta pentagrammata è ancora la più utile e significativa per molti musicisti, siano essi professionisti o meno, come esecutori, studenti, musicologi, ecc; comunque, la notazione su pentagramma è la più diffusa.

Interfaccia Utente.

Il lavoro musicologico viene fatto, in passaggi successivi, da operatori con differenti competenze. Questa è una delle principali ragioni che hanno influenzato la nostra scelta di strutturare l'interfaccia utente in diversi modi per le varie funzioni necessarie: inserimento, verifica, correzione, aggiustamenti tipografici, analisi musicologica, ecc..

Il Software gira su personal computer Apple Macintosh II e SE interfacciato con una tastiera MIDI. L'interfaccia utente è divisa in due ambienti:

- *input*, per l'inserimento delle singole voci;
- *tipografico*, per la definizione grafica della partitura completa.

Le funzioni dell'ambiente di input sono utilizzabili per l'inserimento di nuovi brani; le funzioni disponibili nell'ambiente tipografico, invece, sono necessarie nei successivi passaggi e sono quindi indispensabili anche considerando l'utilizzo del sistema in connessione con database preesistenti.

L'ambiente di inserimento, studiato per un input naturale, ergonomico e intuitivo, anche per il musicista non familiarizzato con i computer, propone diversi modi di interazione tra loro concorrenti:

- tastiera musicale;
- pedali;
- tastiera alfanumerica;

— mouse.

Ciascuno di essi ha una funzione specifica se utilizzato assieme agli altri, però può essere ignorato utilizzando solamente i modi più generali.

Entrando nello specifico possiamo vedere che:

- l'altezza delle note è inserita attraverso la tastiera musicale (soluzione molto semplice e veloce per i musicisti);
- le durate, le dinamiche e gli altri segni di uso comune (programmabili con alcuni tasti funzionali) sono inseriti con la tastiera alfanumerica;
- i segni meno usati come gli abbellimenti, le indicazioni agogiche, ecc. sono disponibili da menu con il mouse.

Così la maggioranza delle operazioni di inserimento può essere fatta tenendo ferme le due mani: ad esempio, una impostata sulla tastiera musicale e l'altra sulla tastiera del computer, o sul mouse. Ma se necessario per il genere di musica o per la disosizione dell'operatore, è possibile realizzare le operazioni di codifica senza la tastiera musicale, solo con la tastiera alfanumerica e il mouse.

A seconda della complessità del testo musicale si può inserire la singola parte in successivi passaggi, ritornando sulle stesse battute per aggiungere le informazioni mancanti.

La tastiera musicale attualmente utilizzata è una tastiera MIDI (Yamaha DX7 II) con cuffia o amplificazione per il controllo audio e con due pedali (footswitch). I pedali hanno le stesse funzioni che nel pianoforte:

- il destro per le legature di frase,
- il sinistro per i modi d'attacco (staccato, appoggiato, ecc.) come definito da menu.

Nell'altro ambiente, quello tipografico, si opera sulle parti (o sulla parte) precedentemente codificate, o trasferite e trascodificate da altri database.

I criteri tipografici adottati contengono già una certa conoscenza sulla stampa della musica e sulla composizione delle pagine (9, 10, 11, 12), ma la caratteristica più importante di questa workstation è la possibilità di modificare gli stessi criteri grafici e soprattutto di aggiustare a mano (con il mouse sullo schermo) la disposizione dei segni.

È quindi l'esperienza e la competenza dell'operatore, che considera il bilanciamento complessivo della pagina e il suo senso musicale, che definisce l'immagine tipografica di ciascuna pagina e dell'intera edizione.

L'editing grafico è diviso in due livelli:

- *preview*, che è la realizzazione di default della partitura dalle singole parti (solo la sincronizzazione verticale) che gira nella workstation di input basata sul Macintosh SE;
- *definitivo*, che ha tutte le possibilità grafiche e che gira su Macintosh II con Megascreen.

Strutture dati.

I dati sono organizzati in due forme:

- *testo*, per il nucleo musicale (core);
- *gerarchica*, per la descrizione completa della partitura (musicale e grafica).

I file testo sono costituiti da informazione musicale che rispetta la codifica definita ed adottata al L.I.M., che è un'evoluzione della codifica utilizzata presso la Divisione Musicologica del CNUCE-CNR (Pisa) e del DARMS.

Il file testo è la forma che può essere utilizzata per le analisi musicologiche e per lo scambio tra differenti sistemi (con trascodifica) e eseguito via MIDI. L'esecuzione MIDI (utilizzabile per la correzione e l'analisi, oltre che per la fruizione) è realizzata trasformando il core musicale in un file MIDI eseguibile secondo formato Opcode.

L'altra forma, gerarchica, è la descrizione del brano con l'informazione grafica completa della partitura. Un brano è organizzato gerarchicamente come segue:

- * un *brano* è una lista di pagine;
- * una *pagina* è una lista di accollature;
- * un'*accollatura* è una lista di righi;
- * un *rigo* è una lista di simboli musicali (note, pause, ecc.).

L'accollatura è l'insieme dei righi sincroni, può coincidere o meno con la pagina ed è costituita da un numero variabile di righi.

La struttura dati del brano è un nodo secondo la definizione (in Pascal) riportata in Tabella A nella pagina seguente.

TABELLA A: definizione Pascal della struttura dati.

```

TYPE
  elemtype=(fixsimb,fixmus,polisimb,var2simb,var4simb,
            testo,rigo,accoll,pagina);

  listpnter^listelem;

  listelem=record
    prox:listpnter;
    pos:point;
    bordo:rect;
    case tag:elemtype of
      fixsimb:(nfsim:integer);
      fixmus:(nmsim:integer;
              musinfo:longint);
      polisimb:(npsim:integer;
                musdata:longint;
                ptrsimb:listpnter);
      var2simb:(n2vsim:integer;
                v2pt:point;
                datamus:longint);
      var4simb:(n4vsim:integer;
                pt1,pt2,pt3:point);
      testo:(dimfont:integer;
              numfont:integer;
              stylefont:set of styleitem;
              words:^str255);
      rigo:(numrigo:integer;
            lung:integer;
            primosimb:listpnter);
      accoll:(numaccoll:integer;
              accsimb:listpnter;
              firstrigo:listpnter);
      pagina:(numpag:integer;
              page:rect;
              firstacc:listpnter;
              pagsimb:listpnter)
    end;

  VAR radice:listpnter;

```

Memorizzazione.

La workstation più piccola (Macintosh SE) memorizza circa 80 pagine su di un floppy disk da 3,5”.

La workstation basata sul Macintosh II registra le partiture complete sull’hard disk: circa 1.000 pagine in un disco da 40 Mbytes. In prospettiva è semplice pensare ai CD-ROM (14) come supporti molto adatti e convenienti per immagazzinare una grandissima quantità di partiture sia in forma stampabile sia in forma di testo. Così da poter utilizzare questo archivio sia per operazioni di stampa, analisi, ricerca, sia per realizzare versioni editoriali particolari per le proprie esigenze.

Il “Tipografo Musicale”.

Come già accennato un obiettivo primario del nostro progetto è la stampa della musica e quindi l’editoria musicale. La scelta più importante è quella di non porre vincoli meccanici o logici al revisore e all’editore ma di fornire loro l’aiuto che un sistema di calcolo specializzato può dare.

E, in confronto con il modo tradizionale di scrivere e incidere la musica, il computer permette successive elaborazioni e differenti versioni provvisorie dello stesso lavoro.

In questo modo siamo arrivati a formulare il concetto di “Tipografo Musicale”: la caratteristica più saliente è la possibilità di aggiustare e posizionare ciascun segno della pagina indipendentemente. Questa, unita all’impostazione automatica della pagina con i criteri grafici impliciti, porta alla miglior cooperazione tra la regolarità e velocità della macchina e la sensibilità ed esperienza dell’uomo.

Una volta che le parti da mettere in partitura sono state selezionate, il sistema calcola lo “spine” per ciascuna accollatura e le dispone nella pagina. Lo spine (spina dorsale) è la struttura ritmica dell’intera partitura che consente il corretto incolonnamento sincronizzato delle singole parti. La pagina proposta può essere poi modificata spostando ciascun segno all’interno del livello più ampio che lo comprende, ad esempio: le note all’interno del rigo, il rigo all’interno dell’accollatura, ecc..

Ogni modifica sarà riallineata dal sistema, ma sempre rispettando la priorità della decisione dell’operatore. Così che è il gesto dell’operatore, o la decisione del tipografo a caratterizzare l’immagine della pagina.

Evidentemente è semplicissimo stampare le singole parti creando delle accollature di singole voci.

Chiaramente sono disponibili diversi set di fonti. Attualmente è implementata la fonte PostScript “Sonata” della Adobe, ma comunque qualunque altra può essere disegnata o acquisita (mediante scanner a 72 e 300 dpi) per le edizioni richieste. Tra le fonti del nostro sistema ci sono segni che cambiano dinamicamente forma e dimensione in riferimento al contesto in cui sono posti. Questi segni (ad esempio: graffe, legature, stanghette, ecc.) sono calcolati parametricamente per mezzo di opportuni polinomi (splines, curve di Bezier). I parametri sono basati su di un insieme di punti che l’utente può selezionare e modificare con il mouse sullo schermo.

L’output ha due modi di stampa:

— bozza, con una qualità di 75 dpi, per il controllo e il preview (Apple Imagewriter);

— laser, con una qualità di 300 dpi, per la stampa finale (Apple Laserwriter).

Inoltre in prospettiva c'è lo sviluppo di un progetto per guidare una fotocompositrice così da raggiungere una qualità di stampa adeguata alle esigenze editoriali professionali.

Prospettive.

Una prospettiva immediata, dopo l'edizione delle partiture recuperate, è la loro diffusione e valorizzazione. L'esecuzione dei brani sarà curata dal gruppo strumentale "Musica Insieme" di Cremona diretto dal M^o Pietro Antonini; questa, insieme alla registrazione, costituirà un primo consistente incentivo alla diffusione di analoghe, e più vaste, operazioni di recupero di antichi brani musicali.

Vista la ricchezza della storia musicale del nostro paese, è auspicabile che in questo settore si realizzi un fruttuoso intervento dell'informatica a favore della cultura musicale.

Ringraziamenti.

Il nostro particolare ringraziamento va a G. Salvetti (docente di Storia della Musica al Conservatorio G. Verdi di Milano) che ha guidato tutta la parte musicologica del presente progetto.

Desideriamo inoltre ringraziare: M. Abbado (direttore del Conservatorio G. Verdi di Milano), G. Acciai (editoria musicale, Suvini Zerboni), P. Antonini (direttore del gruppo strumentale "Musica Insieme"), S. Bellini (ricercatore del L.I.M.), M. Brancaleoni (ricercatore del L.I.M.), N. Gargiulo (responsabile di progetto, ENIDATA), A.Z. Laterza (bibliotecaria del Conservatorio G. Verdi di Milano), A. Rodriguez (direttore artistico del L.I.M.), G. Zanarini (comitato tecnico-scientifico, ENIDATA).

Il sistema software è stato disegnato da L. Finarelli e G. Haus. Lo sviluppo del software è stato realizzato da S. Bellini e M. Brancaleoni.

Questo progetto è finanziato da ENIDATA SpA e supportato tecnicamente da Apple Computer SpA.

Bibliografia.

1) A.R. BRINKMAN — *Representing Musical Score for Computer Analysis*. Journal of Music Theory, 30/2, 1986.

2) D. BYRD — *User Interfaces in Music-Notation Systems*. ICMC Proceedings, The Hague, CMA Publ., 1986.

3) C. ROADS — *An Overview of Musical Representation*. In Musical Grammars and Computer Analysis, Baroni/Callegari Ed., Olschki Publ., Firenze, 1984.

4) C. YAVELow — *A Report of Workshop for Music Notation by Computer*. Computer Music Journal, 11/2, MIT Press, 1987.

5) A. GOMBERG — *A Computer Oriented System for Music Printing*. PhD Thesis, Washington University, 1975.

- 6) K.B. HEWLETT, E. SELFRIDGE-FIELD — *Directory of Computer Assisted Research in Musicology*. CCARH, Menlo Park, CA, 1986.
- 7) K.B. HEWLETT, E. SELFRIDE-FIELD — *Directory of Computer Assisted Research in Musicology*. CCARH, Menlo Park, CA, 1987.
- 8) S. BELLINI — *Un sistema per l'analisi/sintesi di testi musicali*. Tesi di Laurea in Fisica, A.A. 85-86, Università degli Studi, Milano.
- 9) L. BERGOGNONI, M. CROCE, G. HAUS, S. MASSAZZA — *Un sistema editoriale per la composizione elettronica di testi musicali*. Atti del Convegno sui "Text Processing", A.I.C.A., Milano, 1985.
- 10) S. MASSAZZA — *Tecniche per l'impaginazione e la stampa di partiture musicali mediante elaboratore elettronico*. Tesi di Laurea in Fisica, A.A. 83-84, Università degli Studi, Milano.
- 11) S. DUNNE, H. JURGENSEN — *Foundation for a General Mark-Setting System with Applications to Musical Score Formatting*. Report n. 171, Univ. Western Ontario, 1987.
- 12) J. RASKIN — *A Hardware Independent Computer Drawing System using List Structured Modeling: the Quickdraw Graphics System*. MS Thesis, Pennsylvania State University, 1967.
- 13) M. BRANCALEONI — *Un sistema elettronico per la tipografia musicale*. Tesi di Laurea in Scienze dell'Informazione, A.A. 86-87, Università degli Studi, Milano.
- 14) L. FINARELLI, G. HAUS — *Un data base musicale su CD-ROM*. Atti del Convegno sui "CD-ROM", A.I.C.A., Milano, 1986.

SESSIONE COMPOSIZIONE MUSICALE

IMPORTANZA DELLA DIDATTICA PER L'EVOLUZIONE DELL'INFORMATICA MUSICALE

S. Sapis, A. Vidolin

C.S.C. - Centro di Sonologia Computazionale dell'Università di Padova

1. Introduzione.

La produzione musicale contemporanea fa riferimento a due filoni principali: uno, industriale e commerciale, che è rivolto al gran pubblico della cosiddetta musica leggera (canzoni, pubblicità, colonne sonore di film e programmi televisivi) e l'altro, di ricerca, che riguarda un'area d'ascolto più ristretta perpetuando la tradizione musicale colta. Entrambi questi filoni utilizzano le nuove tecnologie informatiche e richiedono la nascita di nuove figure professionali e un aggiornamento delle professioni già esistenti. In alcuni casi, i mezzi tecnologici sono utilizzati per abbassare i costi di produzione cercando di mantenere inalterato il risultato musicale. Normalmente si attua una drastica riduzione del numero degli esecutori che spesso porta alla produzione autonoma della musica da parte del compositore. In altri casi lo strumento informatico diventa il vero significato della musica, e ciò giustifica un continuo aggiornamento sul piano tecnologico che serve come illusione per fornire un'immagine di avanguardia verso l'esterno. In altri casi ancora il mezzo informatico diventa effettivamente uno strumento di ricerca e di sperimentazione finalizzato all'evoluzione del pensiero musicale e pertanto richiede lunghi tempi di assimilazione. Queste distinzioni possono essere sfumate perché tali aspettative si possono trovare in entrambi i filoni.

Le figure professionali necessarie all'utilizzazione di tali mezzi possono essere divise in tre categorie, che riguardano rispettivamente la produzione musicale, la liuteria elettronica e la didattica. Per quanto concerne la produzione musicale dobbiamo distinguere i due momenti di realizzazione di un lavoro: la fase di creazione da quella di esecuzione. Il compositore che tradizionalmente è l'unico artefice della fase creativa viene oggi affiancato per la parte operativa da un esperto tecnico-musicale. Anche nell'esecuzione l'interprete (direttore d'orchestra, strumentista) viene affiancato o sostituito dall'esecutore tecnico che può suonare uno strumento elettronico oppure curare la regia del suono.

Nel campo della liuteria elettronica possiamo distinguere la fase della ricerca da quella della produzione industriale. La ricerca si articola in un momento più teorico che fonde gli aspetti dell'informatica, dell'acustica, della psicologia e della musica in un modello globale che in un secondo momento può essere sperimentato allo stato prototipale e successivamente passare ad eventuali applicazioni industriali. Di conseguenza incontriamo la figura del ricercatore di estrazione universitaria in grado di lavorare in equipe interdisciplinari e quella più operativa del tecnico che sviluppa i prototipi.

Nel campo della didattica musicale si possono distinguere due livelli di intervento: uno professionale e l'altro che riguarda la cultura musicale generale e che ci sembra necessario arricchire oggi del contributo di idee che nascono dall'informatica musicale. Così, mentre per la cultura generale è sufficiente un semplice aggiornamento degli insegnanti, ci sembra più difficile individuare un profilo dell'insegnante in grado di coprire da solo le esigenze di tutte le figure professionali sopra elencate.

Lo scopo di questo articolo è quello di analizzare le problematiche relative alla didattica per la formazione professionale di queste nuove figure e di fornire delle indicazioni.

2.1 Aggiornamento del musicista tradizionale.

L'interprete tradizionale viene sempre più spesso a contatto con sistemi di amplificazione, elaborazione, registrazione del suono. Saper suonare davanti un microfono è pertanto indispensabile. Nel caso del *Live Electronics*, in particolare, l'esecutore deve saper interagire con l'intero sistema elettroacustico che si può considerare un'estensione dello strumento tradizionale. Esistono inoltre molte composizioni per strumenti e nastro che richiedono all'interprete particolari abilità. È quindi necessario saper leggere una partitura in cui la parte memorizzata su nastro è indicata solo in termini evocativi con riferimenti temporali assoluti espressi in secondi, oppure saper seguire il tempo fissato nel nastro e riconoscere gli eventi sonori chiave per la sincronizzazione.

Il direttore d'orchestra oltre a ciò è spesso messo a confronto con i problemi di sincronismo fra esecutori che seguono metronomi diversi la cui scansione temporale è fissata su nastro magnetico per l'ascolto in cuffia o la segnalazione viene data attraverso spie luminose. Nel *Live Elettronics* il direttore deve saper differenziare e equilibrare i suoni acustici da quelli elettroacustici anche in presenza di effetti quali trasposizione, ritardo, accumulazione, ecc., che possono presentarsi anche in forme molto leggere e quindi ambigue. Inoltre con il diffondersi di sale ad acustica variabile il direttore deve essere in grado di trovare la miglior configurazione acustica della sala in funzione delle musiche in programma.

2.2 Nuovi musicisti.

Prima di entrare nel dettaglio delle nuove figure professionali che nascono con l'impiego dei mezzi informatici è necessario analizzare gli strumenti e le tecniche di realizzazione di un lavoro musicale. Come si è già accennato, la prassi di produzione corrente prevede una prima fase di ideazione, scelta e preparazione dei materiali sonori e compositivi e una seconda di realizzazione (esecuzione) del lavoro. L'elaboratore può essere utilizzato in entrambe le fasi grazie all'intervento di particolari tecnici-musicisti.

Nella prima fase l'elaboratore viene utilizzato come aiuto alla composizione e tale utilizzo può avvenire a diversi livelli. Si può passare dalla composizione automatica dell'intero brano alla generazione di singole parti o sezioni "montate" poi a mano in un secondo momento. Si può passare dalla generazione di una partitura simbolica, come ad esempio quella tradizionale su pentagramma, a una più operativa per la sintesi del suono mediante l'elaboratore stesso. I tutti questi casi devono essere definiti e programmati insieme di regole che rappresentano le scelte compositive; queste regole, poi, operano su materiali simbolici o sonori portando alla realizzazione informatica del lavoro. È difficile che un compositore abbia le conoscenze necessarie per realizzare autonomamente tutto il lavoro, pertanto diventa obbligatorio l'intervento di uno specialista che sappia esplicitare le esigenze intuitive del musicista in formalismi e programmi. Il suo ruolo, quindi, è simile a quello dell'analista informatico con una spiccata preparazione nel campo della teoria musicale e nella elaborazione e sintesi dei suoni.

Nella fase di esecuzione dal vivo il materiale sonoro può provenire da un nastro magnetico oppure da un processore audio-numerico funzionante in tempo reale. Nel primo caso è possibile l'intervento di un esecutore che "interpreti" il nastro operando alla regia del suono secondo le specifiche indicate in partitura (Stroppa, 85) ad esempio spazializzazione del suono, andamenti dinamici, sincronizzazioni con eventuali strumentisti, ecc.. Il ruolo di questo

esecutore è spesso sottovalutato in quanto erroneamente si presume che il nastro non richieda ulteriori interventi.

Nel secondo caso l'interprete dal vivo svolge un ruolo che può andare da quello di un'esecutore tradizionale a quello di un direttore d'orchestra. Infatti un sistema in tempo reale può presentarsi attraverso differenti ambienti esecutivi (Sapir, Vidolin, 85) che vanno dal singolo strumento a insiemi più complessi.

L'uso dei processori audio-numeriche in tempo reale, oltre alla tradizionale sintesi dei suoni, consente di elaborare materiali elettroacustici dal vivo e di reagire come un'interprete sintetico a sollecitazioni sia acustiche che gestuali dell'*interprete informatico*. Quest'ultimo deve sapersi adattare ai vari ambienti esecutivi che possono prevedere come dispositivi gestuali di ingresso sia tastiere tradizionali che nuovi strumenti. Anche nel caso di tastiere tradizionali, comunque, il significato di ciascun tasto può essere differente da quello convenzionale e ciò comporta una pratica strumentale completamente rinnovata.

2.3 Ricerca e Liuteria.

Come si accennava in precedenza anche nel campo dell'informatica musicale esiste la logica di produzione che parte dalla ricerca di base, prevalentemente teorica, a quella applicativa per portare infine alla produzione industriale di nuovi strumenti musicali. Val la pena sottolineare come nel nostro secolo la liuteria acustica ha prodotto pochissimi nuovi strumenti ma ha concentrato l'attività di ricerca nel perfezionamento degli esistenti e nell'industrializzazione della produzione. Al contrario, con l'avvento dell'elettricità, c'è stata una proliferazione di idee, invenzioni e realizzazioni che assieme alle nuove esigenze musicali hanno aperto la strada ad una ricerca supportata dalla metodologia scientifica. Il nostro secolo ha prodotto un'infinità di strumenti musicali elettronici e soprattutto negli ultimi anni, con l'avvento delle tecniche digitali, c'è stata una vera esplosione della produzione. Tale produzione si discosta notevolmente da quella tradizionale acustica in quanto lo strumento elettronico è un mezzo per inventare "Nuovi strumenti musicali" intesi sia in senso tradizionale che più astratto.

Purtroppo la completa libertà teorica che caratterizza lo strumento elettronico e che lo rende adattabile alle svariate esigenze del compositore viene spesso ristretta per ragioni commerciali dall'industria. Di conseguenza esiste una buona produzione di simulatori elettronici di strumenti tradizionali ma non vengono prodotti industrialmente strumenti elettronici in grado di sfruttare a pieno le possibilità compositive offerte dalla nuova tecnologia. Questa carenza è dovuta sia a scelte commerciali ma anche alla mancanza di strumenti teorici informatici in grado di manipolare globalmente l'informazione musicale dal macrolivello formale al microlivello sonoro.

È quindi necessaria una attività di ricerca che spinga l'industria verso prodotti più aperti e che si studino per il futuro nuovi strumenti e nuove tecniche che tengano conto delle esigenze del linguaggio musicale contemporaneo. Sono quindi necessari due tipi di ricercatori, uno di livello universitario per ricerche incrociate tecnologico-musicologiche e uno più operativo per lo sviluppo di prototipi e la produzione su scala industriale.

È ovvio che scompare la figura del liutaio artigiano per essere sostituita da quella di un'industria che vive in stretto contatto con la ricerca scientifica interdisciplinare.

2.4 Compositore.

La figura del compositore ha un ruolo chiave in questo contesto in quanto è l'utilizzatore

dei prodotti ma nello stesso tempo è lo stimolo per il loro sviluppo. Di conseguenza egli deve conoscere i nuovi concetti della musica informatica restando però svincolato dal singolo strumento il quale è soggetto alla continua trasformazione industriale. Deve piuttosto assimilare le potenzialità dei nuovi mezzi per poterli utilizzare sia in senso personale nella composizione che in proposte e stimoli verso il mondo musicale e scientifico. Purtroppo la formazione del compositore d'oggi non tiene assolutamente conto di queste esigenze e soprattutto in questo caso la carenza didattica diventa particolarmente grave. Mentre nel settore scientifico la struttura universitaria supplisce in parte alla formazione aggiornata di ricercatori e tecnici (anche se non specifici del campo musicale) e analogamente per la formazione degli esecutori è possibile integrare nuove materie negli studi tradizionali, per la formazione del compositore, invece, è necessario riformulare in termini più ampi la didattica in quanto è cambiata la figura stessa del compositore ed egli richiede l'assimilazione in età di formazione dei nuovi concetti.

3. Organizzazione della didattica.

Viste le difficoltà di pensare una nuova didattica le inevitabili richieste di aggiornamento vengono spesso soddisfatte attraverso attività che si possono inquadrare come di divulgazione o informazione più che di formazione. Ad esempio negli ultimi anni si sono moltiplicati seminari, corsi settimanali, workshop, mostre, programmi educativi alla radio o alla televisione, lezioni-concerto, libri, riviste specializzate e articoli su riviste di grande diffusione, programmi didattici su personal computer, ecc. che per la loro brevità o sinteticità non consentono l'acquisizione dei concetti di base ma solo la ricezione degli aspetti più immediati, spesso spettacolari o superficiali.

Sul piano della formazione non esistono nemmeno all'estero modelli precisi cui fare riferimento. Come è noto in Italia i due luoghi deputati sono il conservatorio e l'università con la differenza che il primo può considerarsi più vicino ad una scuola di indirizzo professionale che ad un istituto di diffusione di cultura e conoscenza basato sulla ricerca. Pertanto l'istituzione in cui si studia la musica manca di quelle attività di stimolo e supporto che ne garantiscono una crescita e viene ancora lasciato al singolo musicista lo sviluppo del pensiero musicale. Di conseguenza la scuola non è più a contatto con la realtà del mondo musicale che la circonda. La ricerca musicale che viene svolta a livello universitario è prevalentemente a carattere musicologico nel settore storico in quanto è inserita nella facoltà delle scienze umanistiche.

All'estero troviamo alcune analogie prevalentemente nei paesi europei per il fatto che in essi esiste ancora la separazione fra università e conservatorio mentre è completamente diversa la situazione nell'America del nord. In ogni caso riscontriamo una sensibilità ai problemi sopra elencati molto maggiore di quella che si riscontra in Italia. Ad esempio in Francia grazie ad una diversa legislazione gli studi musicali a livello universitario consentono di completare e integrare la formazione del musicista tradizionale come pure di provvedere alla formazione delle varie figure necessarie alla vita musicale quali pedagoghi (dalla scuola di base al livello universitario); operatori culturali (educatori nei centri culturali e sociali); organizzatori per il mondo dello spettacolo (enti lirici, teatri, istituzioni concertistiche); ricercatori e musicologi; critici musicali, ecc.. Nel Nordamerica la musica si studia all'università a diretto contatto con le discipline scientifiche e pertanto risultano privilegiati gli indirizzi tecnologici come l'informatica musicale.

Esistono, comunque, anche in Italia delle eccezioni quali il DAMS dell'università di Bologna per gli aspetti musicologi o il DEI dell'Università di Padova per quelli tecnici. Manca comunque una attività didattica che fonda questi due aspetti.

4. Conclusioni.

In base a quanto analizzato sopra possiamo individuare i tre indirizzi di formazione seguenti. La formazione musicale, che è rivolta a compositori, interpreti e teorici; quella tecnica, che riguarda gli ingegneri del suono, gli assistenti musicali, i ricercatori per la ricerca di base e per il mondo industriale; quella pedagogica, che si occupa di formare gli insegnanti a diversi livelli. Vista la stretta dipendenza che lega il mondo musicale a quello tecnologico è necessario far sì che questi tre indirizzi abbiano un tronco didattico comune che fornisca le basi musicali e scientifiche.

La nostra esperienza didattica si basa prevalentemente sui seguenti insegnamenti: i Corsi di informatica musicale del Centro di Sonologia Computazionale (C.S.C.) dell'Università di Padova che si realizzano prevalentemente nel periodo estivo a partire dal 1983; il Corso di Musica all'elaboratore elettronico che si svolge presso la Facoltà di Ingegneria dell'Università di Padova per la laurea in elettronica; il Corso di Musica Elettronica al Conservatorio "B. Marcello" di Venezia. Questi tre insegnamenti rispondono ad altrettante esigenze diverse di conoscenza e ognuno di questi si rivolge a studenti aventi una specifica preparazione di base.

Da questa esperienza si possono trarre alcune considerazioni che ci permettono di dare alcune indicazioni per impostare la didattica dell'informatica musicale in termini più organici di quanto si faccia oggi.

Per quanto riguarda la parte musicale è necessaria una buona conoscenza della teoria musicale e degli elementi di composizione e di analisi sul repertorio generale e più in particolare della musica contemporanea. Per l'aspetto pratico oltre all'educazione tradizionale dell'orecchio si deve sviluppare la sensibilità all'ascolto dei suoni elettronici e delle manipolazioni elettroacustiche di quelli concreti, impadronendosi delle principali tecniche della recente storia della musica elettroacustica. Per quanto riguarda la parte scientifica è necessaria una preparazione di base a livello liceo scientifico almeno per le discipline quali matematica e fisica. A ciò va aggiunto uno studio specifico sull'informatica che riguardi l'architettura degli elaboratori, le tecniche di programmazione, la conoscenza di almeno un linguaggio formale.

A queste materie di base vanno aggiunte nuove discipline che permettono di collegare il mondo tecnologico-scientifico con quello musicale. Fra queste, la psicoacustica che riguarda gli aspetti della percezione del suono e dei meccanismi di organizzazione dei suoni in fatti musicali; le tecniche di analisi, sintesi ed elaborazione numerica del suono, lo studio dei principali sistemi informatici per la musica dagli strumenti commerciali funzionanti in tempo reale (sintetizzatori, campionatori, sistemi MIDI) fino a sistemi più sofisticati e complessi, linguaggi di aiuto alla composizione e processi di composizione automatica.

Questo studio avrà diversi gradi di approfondimento a seconda dell'indirizzo prescelto che si può realizzare attraverso un approccio con ottiche diverse. Vediamo di chiarire questo concetto attraverso un esempio. Prendiamo spunto dalle tecniche di manipolazione del suono mediante filtraggio: il ricercatore deve saper progettare un filtro numerico in funzione delle esigenze musicali, il compositore deve saper fornire al ricercatore le specifiche di tale filtro e deve poi sfruttarlo nella composizione, l'interprete deve saperlo usare.

Le materie sopra elencate consentono di fornire all'allievo gli strumenti essenziali per lavorare nel campo della informatica musicale; è compito del singolo allievo saper scegliere la strada personale di specializzazione che può realizzarsi attraverso ricerche ed esperienze autonome ovvero operando in collaborazione con istituti di ricerca.

Per quanto concerne i luoghi di formazione, come abbiamo già visto non esiste attual-

mente una istituzione che possa assolvere da sola al compito della didattica dell'informatica musicale. Questo sembra irrealizzabile anche in futuro e non solo per ragioni pratiche in quanto sarebbe estraneo al sistema didattico generale. Ci sembra quindi più ragionevole proporre un collegamento fra istituzioni musicali e scientifiche che consenta piani di studio diagonali in grado di soddisfare questa esigenza di interdisciplinarietà. Oltre a ciò ci sembra necessario affiancare ad una buona preparazione teorica una altrettanto valida esperienza sul piano pratico anche se questo aspetto è più difficile da realizzarsi in quanto gli strumenti si evolvono con la tecnologia e sono lontani da una standardizzazione analoga a quella degli strumenti tradizionali.

Sicuramente questo progetto potrà realizzarsi in maniera più facile se la musica diventa una disciplina universitaria e viene studiata e insegnata in tutti i suoi aspetti. La presenza di un'istituzione che abbia come obiettivo primario lo studio teorico e la ricerca sulla musica porta ad un innalzamento del livello qualitativo anche nella dimensione più artigianale della pratica musicale. Inoltre, per quanto riguarda la musica informatica, vengono sicuramente favoriti quegli scambi di natura interdisciplinare che contribuiscono al suo sviluppo.

Bibliografia.

AA.VV. — *Directory of educational programs. Education committee of the AES*. AES, 1984.

AA.VV. — *Quel enseignement musical pour demain?* Atti del 1° colloquio di pedagogia musicale, Cannes, MIDEM classique, 1985-1986.

SAPIR S., VIDOLIN A. — *Interazioni fra tempo e gesto*. In "Quaderno LIMB 5", La Biennale di Venezia, 1985.

SAPIR S. — *Informatique musicale ed pedagogie*. In "Processeurs audio-numeriques temps reel et informatique musicale: mise en oeuvre d'un systeme d'exploitation et utilisation pour la recherche, la production et la pedagogie musicales". These de doctorat d'etat, vol. 2, pp. 186-215, Université d'Aix-Marseille II, 1987.

STROPPA M. — *Traiettorie... deviata*. Ed. Ricordi, Milano, 1985.

VIDOLIN A. — *Elettronica e Conservatori*. In *Laboratorio Musica*, anno II, n. 16, settembre, 1980.

UN SISTEMA DI AIUTO ALLA COMPOSIZIONE: MUSIC ASSISTANT

F. Villa

C.S.C. - Centro di Sonologia Computazionale - Università di Padova

1. Considerazioni iniziali.

Prima di descrivere Music Assistant desidero premettere alcune considerazioni di carattere generale: oltre che rispecchiare il mio attuale approccio compositivo, esse infatti hanno informato la concezione e la progettazione di Music Assistant.

Il processo del comporre implica un'attività di preparazione e di realizzazione che è lunga dal poter essere formalizzata: molte scelte si determinano durante la fase di stesura. Queste a loro volta generano nuove strategie in un continuo processo di osmosi fra i vari aspetti.

Uno dei possibili modi di essere compositore è perciò quello di comporre musica senza preoccuparsi troppo di comprendere tutte le proprie procedure compositive e volerle a tutti i costi modellizzare. Mi riferisco alle procedure che operano al cosiddetto livello macroformale, ma anche e soprattutto alle funzioni che si instaurano in maniera complessa e inconscia fra i diversi livelli formali, dove, frequentemente, le categorie descrittive e discriminative risultano essere fecondamente inutili. Personalmente non riesco nemmeno a concepire la possibilità di formalizzare completamente il processo compositivo e infatti i risultati dell'automazione totale o quasi non si discostano da una buona musica per ambiente.

Sono inoltre convinto che sia pure pressoché impossibile elaborare un programma di aiuto alla composizione universale, in grado cioè di adattarsi alle esigenze e al pensiero di tutti i compositori: qualsiasi sistema di aiuto alla composizione, anche il più asettico viene costruito in base ad un insieme di assiomi compositivi. Nei limiti di quanto precede, una delle caratteristiche fondamentali di un sistema di aiuto alla composizione deve essere quello di poter essere configurato in modo elastico.

2. Music Assistant.

Music Assistant consiste in un insieme di programmi di aiuto alla composizione utilizzabili su personal computer Apple II che permettono di generare partiture operative MG. Play (1) o Music V (2).

MG. Play è un sistema sviluppato da Mauro Graziani che compila partiture simili a quelle Music V per il controllo di un sintetizzatore Mountain Hardware collegabile all'Apple. È quindi possibile ottenere, associando MG. Play e Music Assistant, una piccola stazione di lavoro domestica, pur nei limiti dei 16 oscillatori in sintesi additiva simulati dal sintetizzatore. Il noto programma Music V di Max Mathews viene invece implementato su elaboratori di maggior potenza, per cui è necessario trasferire la partitura generata con Music Assistant dall'Apple all'elaboratore sul quale gira Music V.

Per mezzo di Music Assistant il compositore definisce in modo interattivo dapprima le caratteristiche generali degli eventi che intende generare e poi redige la partitura.

Le principali fasi di lavoro sono tre (Fig. 1):

1) Definizione delle variabili che il compositore intende tenere sotto il proprio diretto con-

trollo nella successiva fase di generazione della partitura.

Tali variabili possono operare a vari livelli della struttura musicale.

- 2) Definizione dei sottoprogrammi di aiuto alla composizione che dovranno essere impiegati. Alcuni sottoprogrammi speciali di aiuto alla composizione sono già presenti in Music Assistant e possono essere richiamati con istruzioni molto semplici, mentre altri possono essere definiti ex novo (in applesoft) e memorizzati. Fra i sottoprogrammi speciali di aiuto alla composizione ve ne sono di quelli che operano a partire da funzioni a segmenti di retta definite dal compositore per il controllo di parametri vari, mentre altri lavorano su vettori con sequenze di numeri da impiegare ad esempio in procedure iterative.
- 3) Redazione della partitura.

Durante questa fase il compositore assegna i valori definitivi (valori di redazione) alle variabili che controllano la generazione delle istruzioni relative a ciascun evento o ad un insieme di eventi. Ciascun valore di redazione può essere copiato direttamente in un campo dell'istruzione di nota in uscita oppure può essere elaborato in base ai sottoprogrammi di aiuto alla composizione (Fig. 2).

Le variabili, le funzioni, i vettori e i sottoprogrammi di aiuto alla composizione (o le istruzioni di richiamo) vengono memorizzati in files separati. È possibile perciò configurare a piacere i diversi files preparatori a seconda delle esigenze. I sottoprogrammi di aiuto alla composizione vengono integrati all'interno del programma di redazione della partitura (diventano di fatto un unico programma di generazione della partitura, e questo è l'aspetto forse più interessante), mentre il contenuto degli altri files viene letto e memorizzato in vettori interni al programma stesso (Fig. 3).

Le variabili e i sottoprogrammi di aiuto alla composizione possono agire ai diversi livelli gerarchici in cui è possibile scomporre una partitura. Tipicamente al primo livello vengono definiti genericamente i singoli eventi sonori mentre al secondo viene definita la loro collocazione e la loro variazione (timbrica, ritmica, dinamica, ecc.) nel tempo. Ad esempio, in una partitura in sintesi additiva gli stessi operatori possono essere applicati al livello della singola parziale, insieme di parziali (nota), insieme di note, ecc..

Una volta generata la partitura, è possibile elaborarla mediante un programma che opera su aree di lavoro di 100 istruzioni al massimo ciascuna. Tale programma, particolarmente adatto per modifiche o correzioni, contiene un repertorio fisso di sottoprogrammi di aiuto alla composizione.

3. Conclusioni.

Dopo circa due anni di impiego Music Assistant si è rivelato uno strumento assai versatile con il quale è possibile redigere partiture secondo approcci molto diversi fra loro. L'uso di questo sistema è relativamente facile (3) e trova applicazione sia nell'ambito della composizione che in quello della psicoacustica e della didattica (in quest'ultima impiegando in particolare la piccola stazione di lavoro sull'Apple).

Personalmente lavoro con Music Assistant per effettuare piccole prove a casa con il sistema MG. Play, al fine di generare in seguito le partiture Music V che, trasformate presso il Centro di Sonologia Computazionale di Padova in files sonori parziali della composizione, miscelo infine con il sistema ICMS sviluppato da Graziano Tisato (4).

Fra l'altro la generazione di partiture Music V mediante Music Assistant, permettendo di eludere l'uso dei sottoprogrammi di aiuto alla composizione presenti in Music V (tutte le PLF e gran parte delle PLS), fornisce al compositore una partitura per così dire precompilata in grado di evocare in maniera significativa il risultato sonoro e che, per tale ragione, risulta essere un buon supporto durante la composizione.

I limiti principali di Music Assistant sono determinati dalla (relativa) lentezza e poca memoria dell'Apple: il tempo di generazione di una partitura è a volte piuttosto lungo, e la mancanza di ulteriore spazio di memoria non ha reso possibile l'elaborazione di supporti grafici adeguati.

Il naturale sviluppo di Music Assistant è perciò quello di adattarlo ad elaboratori più potenti e di dotarlo di una buona interfaccia con l'utente, mantenendo le caratteristiche di adattabilità compositiva ed elasticità di configurazione.

Bibliografia.

- 1) GRAZIANI M. — *Sistema Musica per Mountain Hardware*. Dattiloscritto, Verona, 1986.
- 2) MATHEWS M. — *The technology of computer music*. MIT Press, 1969.
- 3) VILLA F. — *Music Assistant 1.0*. Dattiloscritto, Venezia, 1987.
- 4) TISATO G. — *Sistema Musica - Manuale Operativo*. Centro di Sonologia Computazionale, Università di Padova, Rapporto interno, 1984.

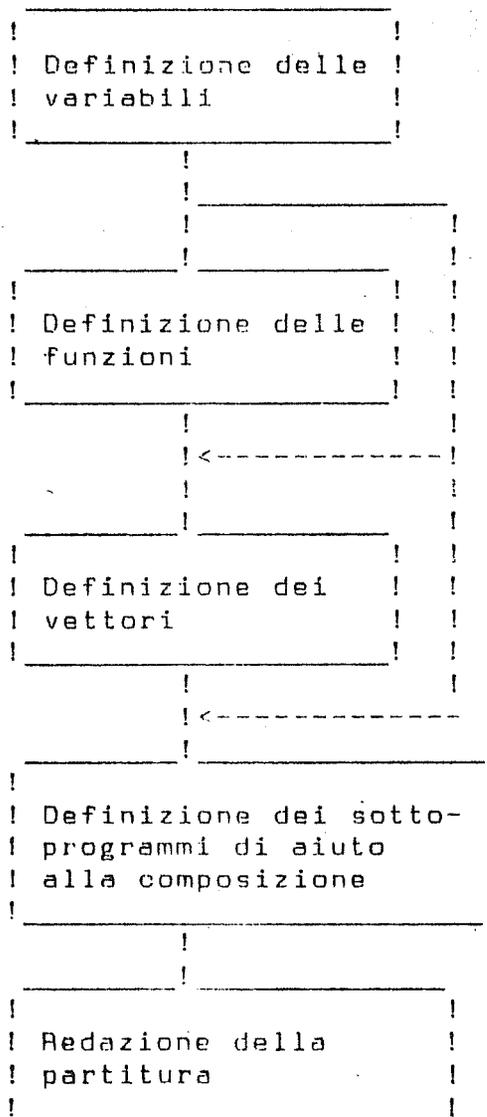


Fig. 1 - Diagramma di flusso del lavoro con Music Assistant.

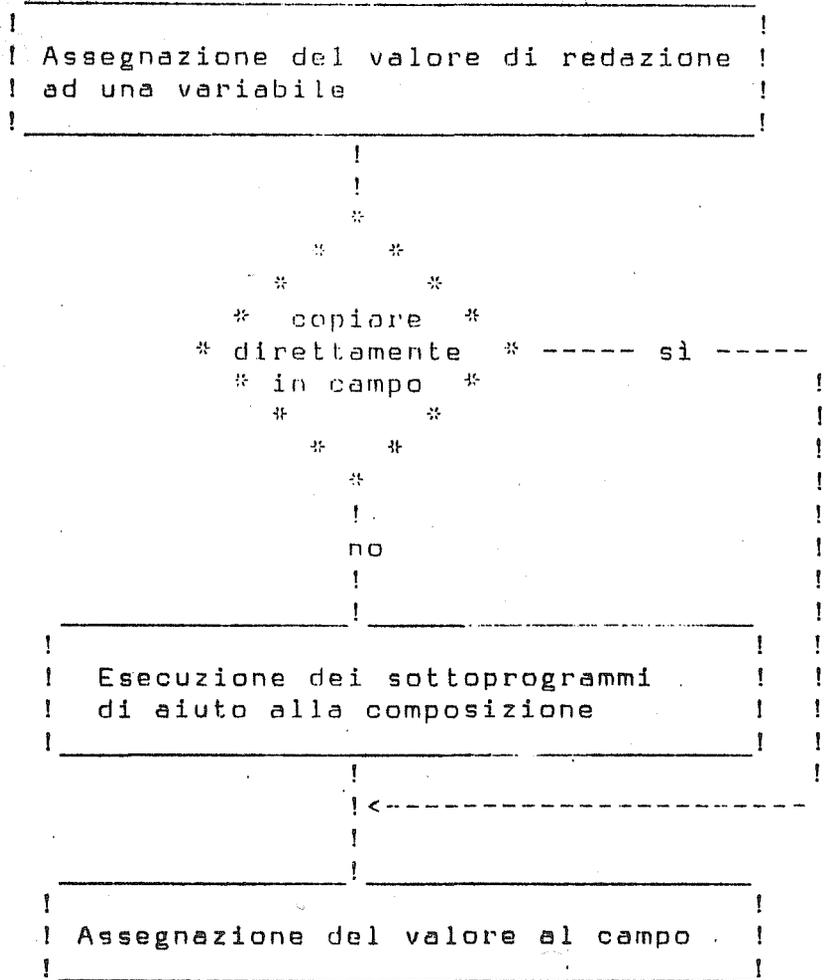


Fig. 2 - Assegnazione del valore di redazione.

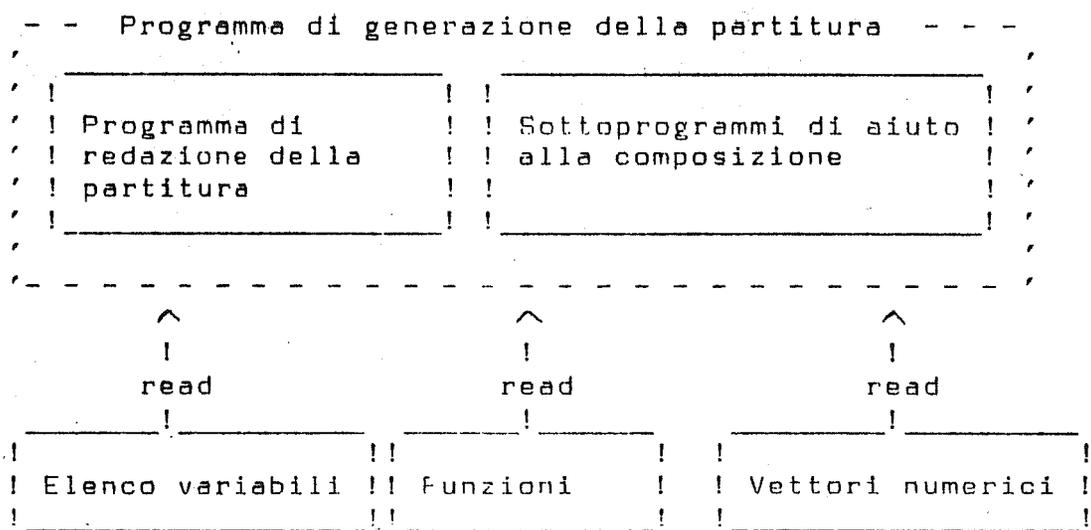


Fig. 3 - Configurazione delle componenti in Music Assistant.

ANALISI DELLA COMPOSIZIONE "AQUAM FLARE IN MEDIA LABIA TUA"

M. Graziani, W. Prati
C.S.C. Padova - Tecnomusica Milano

AQUAM FLARE IN MEDIA LABIA TUA, per trombone e 4I utilizzata dal vivo, è stata composta nel 1987 presso il Centro di Sonologia Computazionale dell'Università di Padova.

La ragione prima che ci ha spinto a percorrere la strada dell'interazione in tempo reale di computer e strumento tradizionale è quella di una verifica personale delle possibilità di utilizzo "live" di una macchina come la 4I che, nella configurazione attuale, va, forse, pensata più come uno strumento di ricerca che come un vero sistema musicale.

Finora, infatti, il ruolo di questa macchina, anche nei pezzi pensati per una esecuzione dal vivo, si è limitato alla generazione di suoni sintetici. La nostra composizione è la prima ad utilizzare la 4I per il trattamento dei suoni di uno strumento acustico, effettuato rigidamente in tempo reale senza campionamenti preconfezionati.

Questa scelta è stata dettata da considerazioni di tipo compositivo. Si desiderava poter disporre, per l'elaborazione, di un materiale estremamente mobile, che potesse mutare nel corso dell'evoluzione del pezzo e legare strettamente con il suono emesso dal solista.

L'idea musicale, infatti, è quella di creare un percorso evolutivo tra il "solo" tradizionale, alla cui scrittura normalmente ci si riferisce in questi casi, e la creazione di un tessuto sonoro molto complesso e mutevole sia sotto il profilo polifonico che timbrico.

Inizialmente si alternano, così, veloci e brevissime acciaccature a note più lunghe; queste ultime vengono acquisite dall'elaboratore e prolungate ("loop" in tempo reale) dando modo al trombone di effettuare altri interventi.

In questa prima sotto-sezione, la 4I trattiene soltanto note singole a cui il solista sovrappone suoni a distanza variabile all'interno del tono, eseguiti con leggere modulazioni di coulisse nell'ambito di micro-intervalli.

Le acciaccature tendono, in seguito, a divenire vere e proprie linee solistiche, mentre lo spessore dei suoni tenuti cresce, in quanto l'esecutore al computer inizia ad accumulare il materiale via via acquisito, producendo un tappeto di frequenze i cui intervalli, sempre molto stretti, tendono a creare variazioni timbriche e battimenti di diversa frequenza.

Il climax così stabilito raggiunge il suo apice con un glissato eseguito con la macchina sul materiale acquisito, da cui si alza una linea solistica decisamente definita. Si tratta di un segnale tecnico/musicale di ciò che caratterizzerà la seconda fase, basata, per il trombone, su un materiale sviluppato da tale linea fino a creare una struttura strettamente legata alla figura contrappuntistica, mentre la macchina si impadronisce dei suoni dello strumento, che utilizza per creare linee di glissati incrociati che si allargano progressivamente con passi che vanno dal quarto di tono fino ad intervalli di 22/4 di tono.

La conclusione di questa ideale prima sezione è affidata ad una graduale concentrazione del solista e del computer verso i glissati.

La seconda parte, invece, si basa su un utilizzo più "tradizionale" della 4I.

L'uso del programma NOT4I, di S. Sapir, permette di introdurre una vera e propria partitura che utilizza suoni sintetizzati che derivano da analisi effettuate sui suoni di trombone.

La politimbricità e la possibilità di utilizzare più voci, hanno reso possibile l'elaborazione di una struttura polifonica con una gamma di timbri che va da quelli più caratteristici del trombone, fino a suoni vicini ad una voce di soprano.

L'analisi tecnica, tuttavia, si concentra sui programmi messi a punto per la prima parte.

Il trattamento del suono nella 4I.

In questa macchina, il meccanismo di acquisizione si basa su un ADC a 16 bit lineari e 16 KHz., che, per ogni ciclo, scarica un campione in una locazione predefinita della data memory.

Per il trattamento, sono stati realizzati due strumenti, pensati in modo da poter coesistere nella 4I. Il primo è in grado di acquisire 1 sec. di suono e di metterlo in un loop infinito; il secondo realizza vari tipi di trasposizioni e delays.

In figura 1 si può osservare lo schema del primo strumento. Il suono viene memorizzato in una linea di ritardo di 16K, con due dispositivi di acquisizione: il primo (a sin.) legge semplicemente 1 sec. di suono e lo memorizza nella RAM, ricavata dalla wave-table memory della 4I, cancellando ciò che la memoria conteneva. Si tratta, quindi, di acquisizione senza accumulazione.

Il secondo, nel corso di ogni acquisizione, rilegge il contenuto della memoria e lo somma al segnale in ingresso, dopo aver moltiplicato entrambi per fattori di guadagno regolabili. In questo caso si ha l'accumulazione di diversi suoni in memoria. I fattori di guadagno consentono di regolare i livelli del nuovo segnale rispetto ai vecchi.

I due meccanismi di acquisizione sono pilotati da triggers che l'esecutore può azionare da tastiera alfanumerica. Lo scatto del trigger provoca la lettura di 1 sec. di suono con la modalità prescelta (cancellazione o accumulazione).

Dato che la lettura è cieca, cioè non presta attenzione alle discontinuità fra campione iniziale e finale, il problema del looping viene risolto nel modo seguente: il segnale memorizzato viene letto da due puntatori fra loro sfasati di 90° . Ai due segnali vengono applicate, separatamente, due versioni dello stesso involuppo trapezoidale sincronizzate in modo tale che il punto di zero sia corrispondente al momento in cui il puntatore di lettura raggiunge la fine della tabella e inizia un nuovo ciclo (punto di quasi certa discontinuità). Dato che i segnali originali sono sfasati, anche gli involuppi risultano fra loro sfasati in modo tale che il sustain dell'uno copre il buco dell'altro. Questo ingegnoso sistema (oppure, secondo altri punti di vista, questo sporco trucco), permette di ottenere, in tempo reale, loops assolutamente privi di clicks. Naturalmente, esso funziona, come tale, solo con suoni tenuti; nel caso di serie di note, invece, si avrà una eco periodica a 1/2 secondo. Questo strumento è attivato/disattivato mediante uno switch di on/off, anch'esso controllato da tastiera, il che permette anche di immagazzinare materiale e utilizzarlo in un secondo tempo.

Il secondo strumento (fig. 2) è concettualmente più semplice. Al segnale acquisito, trattato come in un normale delay, vengono applicate 4 linee di lettura con trasposizione. La prima traspone all'ottava inferiore, alla quinta ed all'ottava superiori ed è usata per rinforzare il suono del trombone. La seconda effettua tre trasposizioni la cui entità è controllata mediante potenziometri. Ovviamente, la precisione di tali dispositivi è scarsa, ma questa linea è ottima per produrre glissati di estensione predefinita. La terza non effettua trasposizioni, ma solo sfasamento, anche variabile (effetti stile flanger e phaser). L'ultima crea due trasposizioni variabili, controllabili esattamente da tastiera, scegliendo in una serie definita in precedenza.

Si noti che, a differenza dei normali delays, il materiale è inviato in feedback dopo il trattamento, non prima. Questo, soprattutto nel caso delle trasposizioni, aumenta notevol-

mente la complessità del materiale in uscita e, se l'intervallo di trasposizione è piccolo, permette di creare degli pseudo-glissati.

L'unione di questi due programmi che, come già detto, possono coesistere nella 4I, configura la macchina come un potente strumento di trattamento del suono acquisito in tempo reale. Gli autori desiderano ringraziare lo staff del CSC, in particolare G. Tisato e S. Sapir, nonché G. Schiaffini, esecutore alla prima e paziente collaboratore nel corso delle prove.

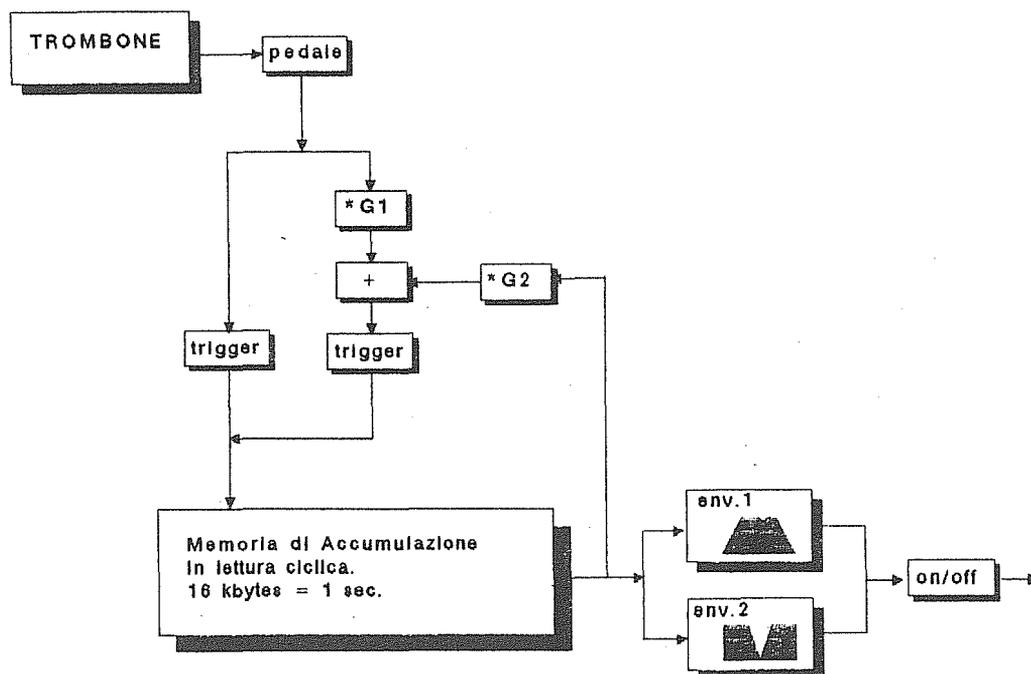


Fig. 1 - Trattamento del suono del Trombone. Strumento numero 1.

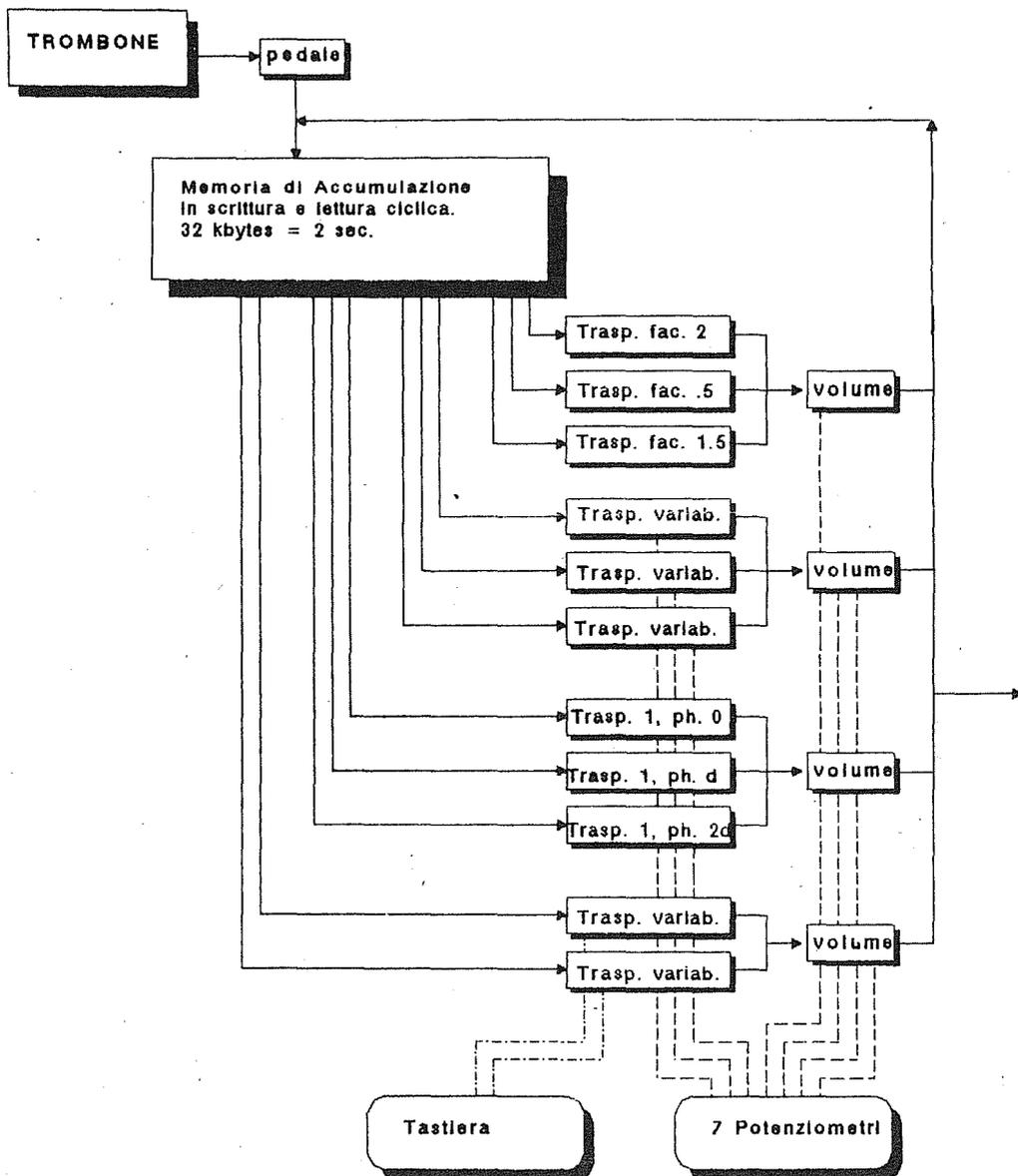


Fig. 2 - Trattamento del suono del Trombone. Strumento numero 2.

PROCESSI COGNITIVI COME PARAMETRI COMPOSITIVI

R. Doati

C.S.C. - Università di Padova - Via Gradenigo 6/A - PD

1.

Negli ultimi vent'anni, a partire cioè da quando grazie all'elaboratore fu possibile compiere analisi dinamiche di suoni naturali, molto lavoro è stato fatto da ricercatori e compositori in merito alla realizzazione di suoni che fossero percettivamente "interessanti", che possedessero un carattere "naturale", superando in tal modo uno dei principali handicap della musica elettronica degli anni '50 e '60 (per una rassegna esaustiva, Risset e Wessel 1982). Oggi, non solo è possibile ascoltare senz'alcun affaticamento percettivo opere di musica informatica, ma il nostro sistema uditivo, affinato da nuovi stimoli sonori, è in grado di cogliere differenze timbriche un tempo inapprezzabili. In un linguaggio complesso quale quello musicale però, oltre alla semplice rilevazione che qualcosa esiste, sono in gioco altri processi. La percezione è solo il primo di questi, cui segue quello in cui l'essere umano assegna un significato a ciò che sente. Per fare questo noi mettiamo in moto meccanismi ancora in gran parte sconosciuti; e la scoperta di questi meccanismi è importante quanto la scoperta dei meccanismi percettivi nel passato, per costruire non solo NUOVI SUONI, ma anche NUOVI SIGNIFICATI.

Se poi consideriamo che nuovi suoni "significanti" rappresenterebbero solo un lessico e che quindi sarebbe necessario dare vita a NUOVE REGOLE, a una nuova sintassi (a meno di non rifarsi a sintassi "storiche", ma allora che senso avrebbe usare un mezzo nuovo quale l'elaboratore?), ci rendiamo conto di quanto cammino debba ancora fare oggi la musica informatica per raggiungere anche un solo livello di confronto con la musica strumentale.

Il compositore deve essere attento e sensibile ai risultati della ricerca psicologica, perché questa gli fornisce NUOVI STRUMENTI concettuali. Delle scoperte della psicoacustica (settore uditivo della psicofisica, una branca della psicologia sperimentale) ha particolarmente beneficiato il compositore alle prese con l'elaboratore, avendo la possibilità di acquisire alcuni importanti principi di composizione DEL suono (autentica rivoluzione della musica elettronica). Ma quanto detto vale anche per la musica acustica, dove la conoscenza dei modi di comportamento del suono (dal punto di vista fisico e dal quello percettivo) è applicata sia alla costruzione di timbri che all'organizzazione formale (si vedano in particolare i lavori di Dufourt, Grisey, Murail) (Dufourt, 1987 (in stampa); Grisey, 1984; Murail, 1982).

Studi recenti (Grey, 1975; Wessel, 1979; McAdams, 1982) hanno rivelato che il nostro sistema uditivo è più sensibile a qualità globali del suono, quali brillantezza, dinamicità, fusione, piuttosto che ai singoli parametri. (Oggi, ad esempio, non ha più senso effettuare un esperimento sulla percezione dell'altezza utilizzando suoni sinusoidali, poiché sappiamo che l'altezza non è una qualità percettiva legata univocamente ed esclusivamente alla frequenza, ma è la somma di numerosi elementi concorrenti, primo fra tutti la distribuzione dell'energia spettrale).

Purtroppo, però, tutti questi studi sono condotti in condizioni sperimentali (ripetuti ascolti in ambienti appositamente predisposti, con la richiesta di una quantità di attenzione tale da stremare anche il più assiduo ascoltatore di musica) e (a parte rare ed episodiche eccezioni come Grey, 1978) su singoli suoni isolati, (situazione decisamente poco frequente in un'opera musicale). Scarsa attenzione è stata rivolta, per esempio, al conflitto fra modi di organizza-

zione percettiva diversi (Bregman e Pinker, 1978; McAdams, 1984), fenomeno interessante per la musica dove al variare di uno o più parametri è possibile far prevalere percettivamente un timbro (organizzazione simultanea) su una melodia (organizzazione sequenziale) o viceversa.

2.

Uno degli approcci più recenti alla psicologia della musica è quello cognitivista, e scopriamo un interessante dato di fatto che unisce questo orientamento teorico all'informatica. La scienza, e in particolare la psicologia, cognitivista che a partire dagli anni Sessanta cominciava a emergere da un incrocio di varie discipline, ha ricevuto un notevole impulso dall'avvento dell'elaboratore, in quanto quest'ultimo fornisce un modello di elaborazione delle informazioni obiettivo. Concetti quali codificazione e trasformazione delle informazioni, provenienti dalla scienza informatica, vengono assunti dagli psicologi e inseriti in modelli di processi cognitivi.

Con il termine processi cognitivi sono indicate quelle attività mentali note come percezione, attenzione, ricordo, linguaggio e soluzione di problemi. Il più importante obiettivo della psicologia cognitiva è quindi la comprensione della natura e del modo di organizzazione di tali processi (Moates e Schumacher, 1980).

Un modello del sistema di elaborazione delle informazioni che ci giungono dall'ambiente circostante è costituito dalle seguenti componenti:

- 1) recettori sensoriali, ossia gli organi (orecchio, occhio, ecc.) che per primi rispondono all'informazione in "ingresso", e rappresentano quindi l'elaborazione percettiva a basso livello;
- 2) registri sensoriali, sistemi che per darci la possibilità di compiere un'analisi più approfondita sugli stimoli e di derivarne un significato, trattengono per breve tempo una rappresentazione abbastanza completa dell'informazione ricevuta (il registro uditivo viene chiamato memoria ecoica);
- 3) memoria permanente, in cui si trova il nostro personale repertorio di conoscenze del mondo che viene consultato per dare un senso a quanto percepiamo;
- 4) processi di riconoscimento di configurazioni, che scattano per trasformare e analizzare gli stimoli ricevuti (questi ultimi raramente hanno la stessa forma della conoscenza memorizzata) in modo da poterli confrontare con il contenuto della memoria permanente;
- 5) attenzione, processo di scelta, fra le innumerevoli stimolazioni ambientali, della parte che verrà analizzata ed elaborata;
- 6) memoria di servizio (comunemente detta consapevolezza o coscienza), risorsa strettamente legata all'attenzione che entra in gioco quando ci troviamo di fronte a nuove combinazioni di eventi, consentendoci di controllare e modificare elaborazioni in atto.

A questo punto il titolo del mio intervento potrà sembrare provocatorio poiché, si dirà, da sempre il compositore percepisce, si concentra su particolari elementi, ricorda, apprende, usa un linguaggio, risolve problemi. Ma, se è possibile fare un parallelo, il fatto che già Bach utilizzasse tecniche di scrittura come la polifonia fittizia, nulla toglie all'importanza che la scoperta e lo studio di meccanismi di organizzazione percettiva quali la *stream segregation* (Bregman, 1971) (su cui si basa tale tecnica), hanno per il compositore contemporaneo, perché grazie a ricerche come questa è ora possibile vedere tutto ciò non tanto come artificio ma come principio compositivo con regole e variabili ben definite, il che consente una tra-

sformazione creativa e cosciente di tali regole. Inoltre va detto che mentre nel passato la maggior parte delle tecniche interpretative del compositore¹ traeva origine dall'articolazione propria dei diversi strumenti (nell'esempio precedente: gli strumenti ad arco, il flauto), oggi l'uso dell'elaboratore, orchestra virtuale, non può avere la stessa funzione, a meno che non si utilizzino programmi compositivi (sia a livello del suono che della forma) scritti da altri. In poche parole con l'elaboratore² è il compositore stesso a decidere i limiti e le possibilità dello "strumento" che si costruisce.

I processi cognitivi sono in larga misura privati, così nel contesto di cui ci occupiamo, quelli usati dal compositore non coincideranno con quelli di chi ascolta. Ma la musica è costruita su leggi che, se conosciute dall'ascoltatore, possono aiutarlo a decodificare il suono in significato; più approfondita sarà questa conoscenza, più corretta sarà la decodificazione. Nello stesso tempo, più vicine queste leggi saranno a quelle coinvolte nei processi e nelle strutture mentali (simboliche) che entrano in gioco nella determinazione del significato (qualunque sia il tipo di informazione ricevuta: sonora, visiva, olfattiva, ec.), più semplice sarà il processo interpretativo.

La figura che qui si propone è quindi quella di un compositore che, acquisita una buona conoscenza dei meccanismi cognitivi, la utilizzi come strumento concettuale per la costruzione di un discorso musicale decodificabile dall'ascoltatore, senza dover per questo rinunciare ai propri significati³.

In un recente ed esaustivo saggio sull'importanza delle rappresentazioni mentali di dimensioni e strutture musicali alla luce della psicologia cognitivista, McAdams (1987) formula una teoria dei processi di organizzazione nell'ascolto che si propone come nuovo approccio per la comprensione dell'esperienza musicale. Senza entrare nel merito dell'enunciato, ritengo utile indicare le principali aree incluse in tale teoria:

- 1) "lettura" della superficie acustica;
- 2) organizzazione dell'informazione acustica in immagini uditive coerenti⁴;
- 3) segmentazione⁵ ed estrazione di un lessico musicale;
- 4) costruzione di relazioni strutturali;
- 5) conseguimento di un discorso musicale.

Volendo fare un bilancio del lavoro di indagine svolto in queste cinque aree diremmo che mentre le prime due vantano una considerevole quantità di ricerche e la terza inizia a ricoprire un certo interesse per gli studiosi (Lerdahl e Jackendoff, 1983), le rimanenti sono state finora affrontate raramente e in termini più speculativi che pratici. Va comunque segnalato che anche la prima area, e parte della seconda, pur essendo oggetto della ricerca psicoacustica da circa vent'anni, andrebbero riviste sotto una nuova luce, dal momento che l'approccio cognitivista ridefinisce la percezione come il processo della *determinazione del significato* dei suoni che udiamo. In questo senso va letta la ricerca di McAdams (1984a, b), la cui teoria postula un gruppo di regole per la creazione di immagini uditive (organizzazione simultanea), e uno per l'organizzazione sequenziale.

Nella maggior parte degli studi condotti per conoscere la natura e il modo di funzionamento dei processi cognitivi inerenti la musica (in particolare Lerdahl e Jackendoff, 1983) viene presa in considerazione la musica tonale (talvolta anche quella prodotta casualmente dal ricercatore, mai una musica atonale o anche aleatoria ma generata con fini creativi). Come può tale approccio rivestire importanza per il compositore contemporaneo? Credo che

una risposta possa essere data facendo un confronto con quanto avvenuto in campo musicale a seguito delle ricerche di psicoacustica (si veda figura 1). Se attualmente siamo in grado di sintetizzare suoni che, pur nuovi all'orecchio umano, percettivamente risultano "natural", è soprattutto grazie a quegli studi (in particolare Risset, 1969) che verso la fine degli anni '60 si sono occupati della simulazione di strumenti acustici, tradizionali. Ciò poteva non essere chiaro a quel tempo dal momento che, si diceva, il compositore necessita di timbri nuovi e non di caricature di precedenti. Oggi sappiamo che i criteri utilizzati nella composizione di un suono si basano sulla conoscenza di quelle regole "natural" rivelatrici proprio dal lavoro di analisi e sintesi, attuato sui suoni di strumenti tradizionali. Parimenti, l'obiettivo di poter creare nuove leggi per la costruzione di un discorso musicale (composizione COL suono) è raggiungibile solo attraverso la scoperta dei principi che sottendono un sistema musicale stabilizzato quale quello tonale⁶.

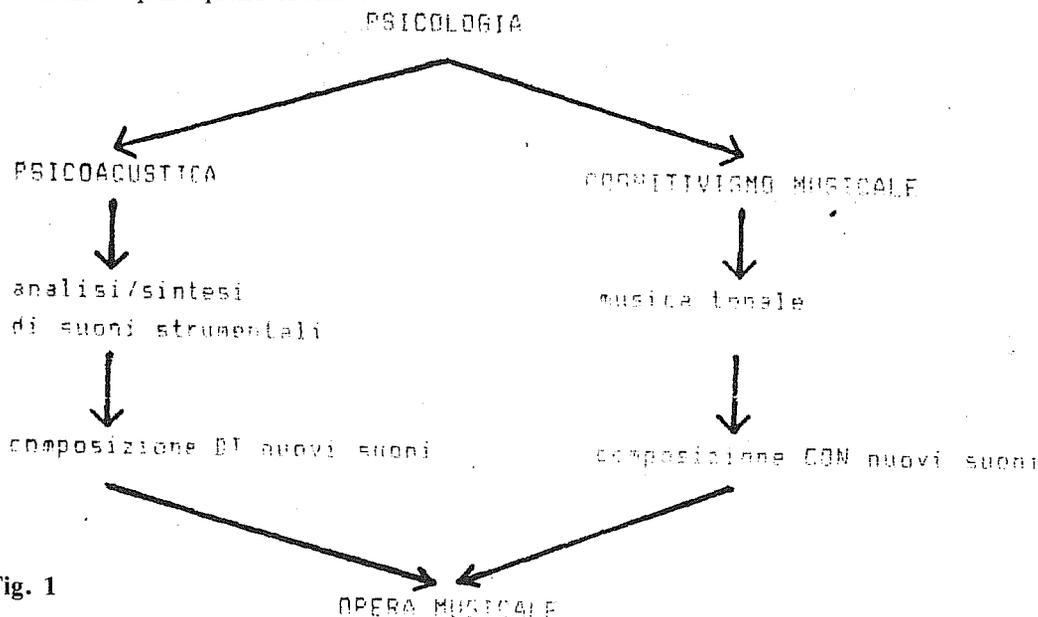


Fig. 1

Pur tuttavia qualche critica a un simile approccio è giustificato muoverla. Per esempio: nello studio del rapporto fra memoria e forma musicale, si ritiene che un ruolo molto importante possa essere svolto da un'organizzazione strutturale di tipo gerarchico dal momento che questa, essendo facilmente memorizzabile, conduce a una migliore percezione della struttura musicale. È risaputo che il parametro di gran lunga più interessante per il compositore in questo secolo è il timbro (Erickson, 1975), eppure oggetto principale di tali studi sono le gerarchie di altezza e ritmo (fa eccezione un recente intervento di Lerdahl (1987)). Se poi si considera la conoscenza che oggi abbiamo della natura multidimensionale del timbro (Grey, 1975) e le enormi potenzialità offerte dall'elaboratore per la costruzione del materiale sonoro, è lecito chiedersi perché in uno studio sulle relazioni strutturali nella percezione dell'altezza musicale quale quello di Cross, Howell e West (1985), si guardi il circolo delle quinte o quello delle qualità tonali e non anche, per esempio, l'inviluppo spettrale.

3.

Per la composizione mediante elaboratore credo si possa parlare di una fase, quella at-

tuale, pretonale. A differenza del recente passato, in cui si cercava la libertà da sistemi percettivi gerarchici (che creavano, per esempio, l'attesa di un particolare grado della scala), oggi il compositore si interessa alla *costruzione* di un tale sistema. Questa volta, però, non si tratta di costruire un sistema di altezze, bensì un sistema di timbri, una rete di relazioni funzionali⁷ fra le diverse dimensioni del timbro (una delle quali è l'altezza). Per far questo il compositore dovrà tenere conto delle funzioni cognitive in gioco nell'ascolto.

Così nella costruzione di quella che è stata definita "superficie acustica", si dovrà considerare la complessa catena che dal suono conduce al senso attraverso attenzione, memorizzazione, apprendimento e ricordo. Non si vogliono qui dare criteri compositivi "universali" perché, oltre all'impossibilità oggettiva di una simile operazione, si ritiene la loro formulazione parte integrante del lavoro creativo dell'autore e quindi essenzialmente personale. Intendo piuttosto fornire alcune indicazioni sui principi che sottendono una mia recente composizione.

Deve essere tenuto lontano da fonti di luce (1985/86) fa uso di un lessico costituito da pochi elementi semplici:

- 1) suoni singoli (che, ispirandomi a Paul Klee (1925), il "musicista" che più influenza il mio lavoro, definisco punti passivi);
- 2) strutture (superfici attive e/o passive);
- 3) glissandi (linee attive);
- 4) suoni concreti⁸. L'adozione di un vocabolario limitato, comune a tutta la musica realizzata esclusivamente con suoni sintetici e che perciò può essere definita musica da camera⁹, aiuta il processo di segmentazione del discorso musicale.

La caratterizzazione di un suono è il problema chiave da affrontare se si vuole rendere riconoscibili (ricordiamo l'importanza cognitiva del riconoscimento) eventi sonori anche quando trasformati e/o in contesti diversi. Per costruire il materiale sonoro dell'opera ho quindi fatto uso di due diversi modelli geometrici di spazi timbrici¹⁰ lungo le cui dimensioni sono ordinati diversi parametri.

In figura 2 è rappresentato lo spazio che riunisce i parametri a basso livello: numero di parziali, ritardi di entrata, durate fisiche relative, involuppi d'ampiezza, involuppi spettrali, tempi d'attacco, deviazioni di frequenza e di ampiezza, ecc.. I diversi valori di ognuno di essi formano i gradi di una scala; ad esempio, le configurazioni (determinate da funzioni grafiche) assunte dagli istanti di inizio delle parziali procedono dal semplice (tutte le componenti iniziano nello stesso istante) al complesso (ogni componente ha un istante d'inizio diverso) e lo stesso vale, con configurazioni diverse, per le durate fisiche. Ancora: da un estremo all'altro gli involuppi spettrali distribuiscono l'energia su una bassa regione dello spettro (timbro scuro) e su diverse regioni lungo tutto lo spettro (timbro chiaro).

A livello superiore è collocato uno spazio timbrico a due dimensioni (figura 3) che controlla principalmente l'armonicità degli spettri secondo due parametri: espansione-compressione e traslazione. Essi determinano uno dei fattori indicati da McAdams nel suo studio sulla formazione di immagini uditive (McAdams, 1984a, b).

L'ordinamento in scala di parametri che danno origine al timbro (elemento fondamentale nella percezione della struttura di un'opera) rappresenta la possibilità di poter individuare categorie percettive. La distinzione di queste categorie sarà in funzione, sia della capacità del compositore di pensare (o meglio rappresentarsi mentalmente) e realizzare i propri oggetti sonori in termini più globali di quanto comunemente avviene (per esempio brillantezza, fusione, dinamicità timbrica, qualità dell'attacco¹¹), sia dell'esperienza che l'ascoltatore ha di

nuove categorie percettive. Riferendoci al modello del sistema di elaborazione delle informazioni presentato, diremo che per poter assegnare un significato a nuovi timbri, occorre possedere nuovi repertori nella memoria permanente, cosa che si ottiene con l'esperienza di un ascolto attivo.

Nella composizione che stiamo descrivendo, l'organizzazione timbrica risulta dalla sovrapposizione dei due spazi di figura 2 e 3, e le sue dimensioni sono, se così si può dire, lette, rivelate nel corso della composizione in maniera discreta dalle strutture e dai suoni singoli, in modo continuo dai glissandi ¹².

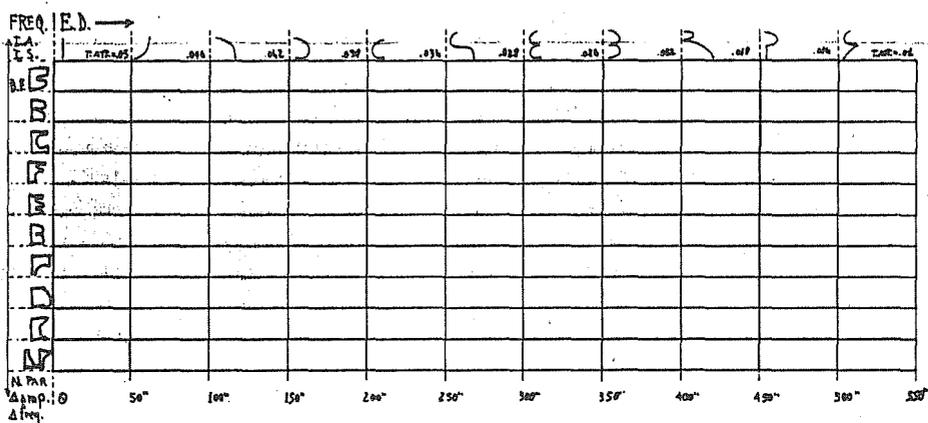


Fig. 2 - Spazio parametrico.

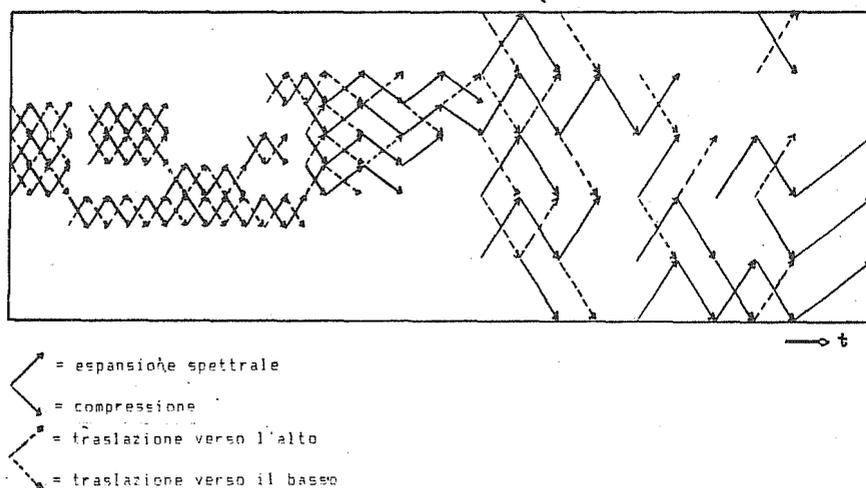


Fig. 3 - Spazio timbrico.

Il materiale sonoro si trova disposto in strutture complesse secondo principi di organizzazione che traggono origine da esperienze personali fatte con le leggi della Gestalt (Doati, 1985). Il fenomeno figura-sfondo e concetti quali buona continuazione, chiusura, somiglianza, sono strumenti tuttora usati per comprendere l'organizzazione percettiva e fanno parte anche dell'approccio cognitivista (per es. Watkins e Dyson, 1985).

In considerazione del fatto che l'ambiguità svolge un'importante funzione cognitiva, ho posto in conflitto timbro e ritmo; secondo Deliege (1985), infatti, i meccanismi che effettuano una segmentazione del discorso musicale operano su:

- 1) continuità spettrale (che è direttamente collegata a cambiamenti in altezza, timbro e dinamica);
- 2) fattori temporali come cambiamento in durata, articolazione e pause fra gruppi e note (cfr, McAdams, 1987).

Inoltre, carattere fondamentale di una struttura in quanto organizzazione di elementi significanti è l'autoregolazione (la struttura controlla e regola sé stessa, diviene fenomeno naturale) (Piaget, 1968) e il ritmo assicura la propria autoregolazione con i mezzi più semplici quali simmetrie e ripetizioni. Da ciò deriva l'uso di strutture ritmiche semplici e regolari con un profondo carattere di ripetizione. Il reticolo timbrico sopra descritto sottolinea, e più spesso altera questa autoregolazione ponendo, come detto, in conflitto organizzazione ritmica e organizzazione timbrica.

La figura 4 (un estratto dalla partitura grafica) rappresenta un esempio di questo tipo di conflitto fra la rigidità ritmica della struttura e una certa variabilità timbrica associata a essa (indicata in basso come variazioni dello spettro).

Lo spazio compositivo rappresentato in figura 5, presenta una forma "a episodi", a sottolineare il modo in cui percepiamo la forma musicale. Estruendo le caratteristiche salienti della massa sonora presentata e attraverso regole di preferenza (influenzate sia dal modo di organizzazione dei suoni¹³ che dalla personale estetica), l'ascoltatore ricava veri e propri episodi che memorizza per poi confrontarli e dar luogo alla percezione dell'intera forma. (È forse inutile precisare che tale comportamento si verifica anche quando nell'opera in ascolto non vi siano pause).

A questo proposito i pochi e brevi rumori isolati presenti nella composizione hanno funzione di nodi semantici, poiché rompendo la ripetitività, l'omogeneità strutturale, provocano un improvviso cambio di contesto che rende più facile la memorizzazione delle strutture.

L'uso di suoni concreti, rappresenta infine la relazione esistente (o provocata) fra mondo sintetico e mondo reale: quale dei due è "più reale"?

Conclusione.

Helmholtz considerava le leggi che svolgono una funzione naturale nel nostro orecchio come "... the building stones with which the edifice of our musical system has been erected..." (Helmholtz, 1863). Ma da quando il grande fisico e fisiologo tedesco scrisse queste parole è passato più di un secolo e nel frattempo si sono scoperte "nuove" leggi naturali. Perché non edificare un nuovo sistema musicale? La psicologia cognitivista, che sta dando il suo grande contributo in quest'opera di disvelamento, può aiutare il compositore a costruire un lessico e una sintassi adeguati all'uso delle nuove tecnologie.

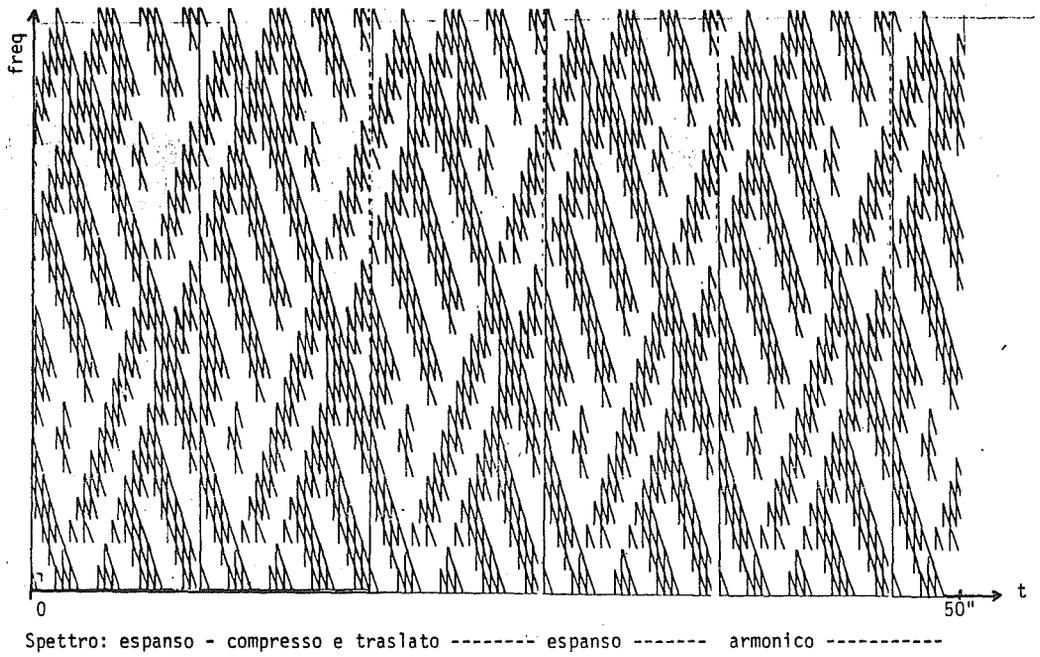


Fig. 4 - Estratto della partitura grafica.

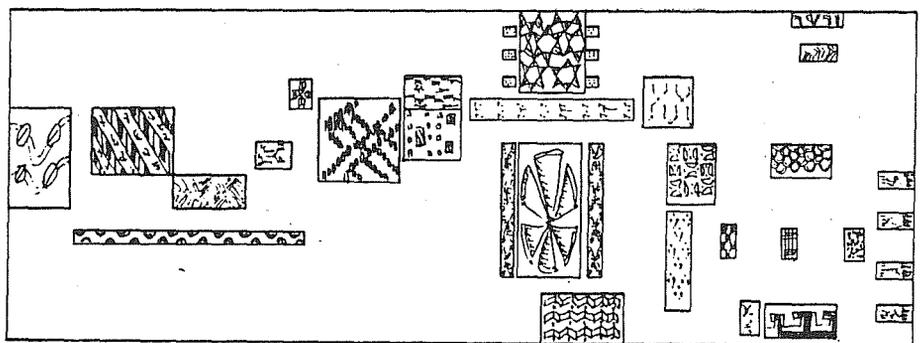


Fig. 5 - Spazio formale.

NOTE.

1) La funzione interpretativa a cui si allude è quella svolta dal compositore nell'interpretare la propria idea musicale per la sua realizzazione. È il passaggio da una fase puramente simbolica a una operativa.

2) Mi riferisco naturalmente a sistemi di una certa potenza, che siano cioè in grado di fornire la più ampia libertà per quanto riguarda la definizione del materiale sonoro.

3) L'essere umano dà un senso a ogni cosa; davanti a un sistema non verbale come quello musicale, e in particolare quello elettronico o informatico, l'ascolto sarà di tipo referenziale. Per cui accade che l'ascoltatore possa sentire un tema, un accordo, uno sviluppo armonico, laddove il compositore non ha impiegato alcuna intenzionalità, travisando così il significato dell'opera. Quanto detto non vuole essere un divieto a chi ascolta di fare riferimento a leggi storiche (tonalità, armonia, ecc.), ma solamente un invito a prendere in considerazione l'utilizzo, da parte del compositore, di principi diversi da quelli che pur possono avere, per ragioni culturali comuni, legami con leggi del passato.

4) L' "immagine uditiva" è una metafora elaborata da McAdams (1984b) e sta a indicare la rappresentazione psicologica di un'entità sonora che nel proprio "comportamento acustico" presenta una coerenza interna.

5) In linguistica tale termine indica la suddivisione del discorso parlato nelle unità componenti.

6) Si dirà giustamente che la musica non risulta dall'applicazione lineare di una serie di regole teoriche. Ma il compositore può violare le leggi solo dopo averle acquisite, solo se in possesso di una grammatica consolidata, e nella composizione musicale mediante elaborazione tale grammatica è ancora tutta da costruire.

7) In un sistema ordinato la funzione cognitiva della relazione fra due elementi sarà diversa da quella fra altri due elementi. Nel sistema tonale, per esempio, il senso dato a un intervallo di quinta è di altro tipo rispetto a un intervallo di settima. Sull'argomento si vedano i criteri scelti da McAdams e Saariaho (1985) per gli elementi che conducono a una forma musicale.

8) Forse a questo breve elenco andrebbe aggiunto un altro elemento: la pausa. Componente musicale altamente espressiva che pare bandita dalla musica informatica.

9) Penso che ciò sia dovuto al fatto che nella musica informatica non si è ancora in grado di costruire un numero elevato di piani "orchestrali" ben differenti.

10) Per la formulazione del concetto di spazio timbrico si vedano Grey (1975) e Wessel (1979).

11) Indispensabile per il passaggio dalla rappresentazione mentale alla realizzazione di un pensiero musicale è la familiarità che il compositore deve avere con le diverse rappresentazioni del suono (temporale, frequenziale, ecc.).

12) Obiettivo ideale, secondo quanto detto finora, sarebbe quello di realizzare un'organizzazione timbrica tale da poter, nell'ascolto, anticipare mentalmente (pur con esiti diversi) il timbro che seguirà, proprio come avviene per una scala di altezze.

13) Meglio organizzata è l'informazione, più informazione siamo in grado di memorizzare per stabilire relazioni.

AA.VV. — *Auditory Processing of Complex Sounds*. A cura di W. Yost e C. Watson, Hillsdale, London, Lawrence Erlbaum Associates, 1987.

BREGMAN A., CAMPBELL J. — *Primary auditory stream segregation and the perception of order in rapid sequences of tones*. *Journal of Experimental Psychology*, 89, 1971.

BREGMAN A., PINKER S. — *Auditory streaming and the building of timbre*. *Canadian Journal of Psychology*, 32, 1978.

CROSS I., HOWELL P., WEST R. — *Structural relationships in the perception of musical pitch*. In: Howell P., Cross I. e West R., *Musical Structure and Cognition*, London, Academic Press, 1985.

DELIÈGE I. — *Les règles préférentielles de groupement dans la perception musicale*, tesi, Bruxelles, Università Libre de Bruxelles, 1985.

DOATI R. — *Simmetria, regolarità, direzione, velocità*. *Bollettino LIMB*, 5, 1985.

DOWLING J., HARWOOD D. — *Music Cognition*. New York, Academic Press, 1986.

DUFOURT H. — *Timbre et espace*. In: Atti del Convegno "Spazio reale e spazio sintetico", Torino, 1987, (in stampa).

ERICKSON R. — *Sound Structure in Music*. Berkeley, University of California Press, 1975.

GREY J. — *An Exploration of Musical Timbre*. Tesi, Stanford, Stanford University, 1975.

GREY J. — *Timbre discrimination in musical patterns*. Journal of the Acoustic Society of America, 64, 1978.

GRISEY G. — *La musique: le devenir des sons*. Darmstadter Beitrage, XIX, 1984.

HELMHOLTZ H. VON — *Die Lehre von den Tonempfindungen als physiologische Grundlage für die Theorie der Musik*. Braunschweig, Vieweg, 1863, (traduzione inglese: *On the sensations of tone as a physiological basis for the theory of music*, New York, Dover, 1954).

KLEE P. — *Pädagogisches Skizzenbuch*, 1925, (traduzione: *Quaderno di schizzi pedagogici*, Firenze, Vallecchi, 1979).

LERDAHL F. — *Timbral Hierarchies*. Contemporary Music Review, 2, 1987.

LERDAHL F., JACKENDOFF R. — *A Generative Theory of Tonal Music*, Cambridge, MIT Press, 1983.

McADAMS S. — *Spectral fusion and the creation of auditory images*. In: Clynes M. (a cura di), *Music, Mind, and Brain. The Neuropsychology of Music*, New York, Plenum Press, 1982, (traduzione: *Fusione spettrale e la creazione di immagini uditive*, Bollettino LIMB, 2, 1982).

McADAMS S. — *The auditory image: A metaphor for musical and psychological research on auditory organization*. In: Crozier W. e Chapman A. (a cura di), *Cognitive Processes in the Perception of Art*, Amsterdam, North Holland, 1984(a).

McADAMS S. — *Spectral Fusion, Spectral Parsing and the Formation of Auditory Images*, tesi, Stanford, Stanford University, 1984(b).

McADAMS S. — *Music: a science of the mind?*. Contemporary Music Review, 2, 1987.

McADAMS S., BREGMAN A. — *Hearing musical streams*. Computer Music Journal, 3, 4, 1979.

McADAMS S., SAARIAHO K. — *Qualities and functions of musical timbre*. Proceedings of the 1985 International Computer Music Conference, Vancouver, Berkeley, Computer Music Association, 1985.

MOATES D., SCHUMACHER G. — *An introduction to Cognitive Psychology*. Belmont, Wadsworth, 1980, (traduzione: *Psicologia dei processi cognitivi*, Bologna, Il Mulino, 1983).

MURAIL T. — *La revolution des sons complexes*. Darmstadter Beitrage, XIX, 1982.

PIAGET J. — *Le structuralisme*. Paris, Presses Universitaires de France, 1968, (traduzione: *Lo strutturalismo*, Milano, Il Saggiatore, 1968).

RISSET J.C. — *An Introductory Catalogue of Computer Synthesized Sounds*. Bell Laboratories, Murray Hill, New Jersey, 1969.

RISSET J.C., WESSEL D. — *Exploration of timbre by analysis and synthesis*. In: Deutsch D. (a cura di), *The Psychology of Music*, New York, Academic Press, 1982, (traduzione: *Indagine sul timbro mediante analisi e sintesi*, Bollettino LIMB, 2, 1982).

SLOBODA J. — *The Musical Mind: The Cognitive Psychology of Music*. Oxford, Clarendon Press, 1985.

WATKINS A., DYSON M. — *On the perceptual organisation of tone sequences*. In: Howell P., Cross I e West R, *Musical Structure and Cognition*, London, Academic Press, 1985.

WESSEL D. — *Timbre space as a musical control structure*. *Computer Music Journal*, 3, 2, 1979, (traduzione: Lo spazio timbrico come struttura di controllo musicale, in Alvise Vidolin, *Musica e elaboratore*, Firenze, Vallecchi-La Biennale di Venezia, 1980).

SESSIONE RAPPORTI DI ATTIVITA'

IL LABORATORIO DI TRATTAMENTO DEL SEGNALE AUDIO
DELL'ISTITUTO DI ACUSTICA "O.M. CORBINO" DEL C.N.R.

V. Asta, P.E. Giua - Istituto di Acustica "O.M. Corbino" - C.N.R. - Via Cassia 1216 - Roma
L.M. Del Duca - Società di Informatica Musicale - via F. Grazioli Lante 30 - Roma
e Istituto di Acustica "O.M. Corbino" - C.N.R.

In seno al Reparto AAE (Acustica Ambientale ed Elettroacustica) dell'Istituto di Acustica "O.M. Corbino", si è recentemente costituito un gruppo di ricerca denominato L.A.S.T. (Laboratorio di Trattamento del Segnale Audio); le sue principali aree di interesse sono il trattamento del segnale vocale (analisi/sintesi della voce da testo scritto e da concetti) e la sintesi musicale elettronica, sia dal punto di vista delle metodologie software che delle architetture hardware dedicate.

Il Laboratorio è strettamente connesso col sistema di calcolo dell'Istituto, d'altronde gestito da alcuni membri del L.A.S.T., che comprende attualmente:

- Un calcolatore microVAX II, con 13 Mbytes di memoria centrale, 590 Mbytes di memoria di massa, 2 unità nastro, 16 linee seriali in DMA, interfaccia Ethernet, sistema operativo ULTRIX (Berkeley UNIX).
- Circa 20 calcolatori IBM-PC (XT ed AT) o compatibili, con disco fisso di capacità dai 20 agli 80 Mbytes, 640 Kbytes di memoria centrale, sistema operativo MS-DOS, connessi per lo più in rete locale Ethernet (protocollo DECnet) tra di loro e con il microVAX.
- Una serie di periferiche connesse in rete (e quindi accessibili indifferentemente dal microVAX o dai PC), comprendente fra l'altro:
 - una stampante Laser,
 - una stampante di linea veloce,
 - due modems 300/1200 bauds (uno in ricezione automatica ed uno in chiamata automatica),
 - due plotters (uno in formato A3 ed uno in A4).
- Un calcolatore Force STANDARD-3U, su bus VME, con unità centrale Motorola MC68010, 2 Mbytes di memoria centrale e 80 Mbytes di memoria di massa, 13 linee seriali, sistema operativo UNIX SYSTEM V, connesso al microVAX tramite un'interfaccia parallela in DMA ad alta velocità (oltre che con links seriali sotto protocollo UUCP).

Prossimamente (si spera entro l'anno) sarà acquistato un Array Processor Mercury ZIP3232+ da 16 MFLOPS, interfacciato col microVAX. È inoltre in via di realizzazione una connessione, tramite linea dedicata ed interfaccia sincrona, con la rete DECnet e quindi SNA (BITNET - EARN), con ulteriore accesso alla rete nazionale ITAPAC a commutazione di pacchetto (protocollo X.25). Ciò renderà tra l'altro più agevole la connessione (peraltro già attiva) con il CINECA di Bologna, dove abbiamo a disposizione tempo macchina sul calcolatore vettoriale CRAY-XMP. Un'ulteriore connessione in fase di attuazione è l'accesso alla rete nazionale e mondiale UUCP (EUNET - USENET).

Le parti più specifiche al Laboratorio sono il sistema Force e tre PC, oltre che un laboratorio di elettronica. I calcolatori sono essenzialmente dedicati all'acquisizione, restituzione e trattamento in tempo reale (tramite controllo di hardware dedicato) del segnale audio: il sistema Force è prevalentemente orientato al trattamento dei segnali musicali, e i PC al tratta-

mento del segnale vocale.

Il sistema Force funge principalmente da periferica intelligente del microVAX: ad esso è interfacciata, tramite un controller di interfaccia parallela in DMA, una scheda (in parte di nostra costruzione) comprendente due convertitori Analogico-Digitale e due Digitale-Analogico a 16 bits (dinamica: 96 dB) optoisolati e con i necessari filtri d'ingresso e di uscita, con frequenza di campionamento fissa a 44.1 KHz (standard PAL/SECAM); la scheda comprende inoltre un microprocessore di controllo dedicato, ed un buffer di memoria (gestito dal microprocessore) per i campioni in ingresso e in uscita.

I PC fungono invece da workstations dedicate per il trattamento del segnale in tempo reale, oltre che da sistemi di sviluppo per realizzazioni hardware a microprocessore. Ad essi sono collegate le seguenti periferiche principali:

- Tre sistemi OROS AU-20, ciascuno dei quali consiste di due schede installate nel bus del PC e comprende un convertitore Analogico-Digitale ed uno Digitale-Analogico, entrambi a 16 bits, anch'essi con i relativi filtri e con frequenza di campionamento programmabile via software, da 250 Hz a 50 KHz; ed un chip di trattamento del segnale in tempo reale TMS32020, con tempo di ciclo di 200 nanosecondi, comunicante con i convertitori tramite un bus interno specializzato.
- Un sistema SPRINT+ per la programmazione di memorie PROM ed EPROM e di dispositivi logici programmabili (PAL, PLA etc.).
- Un emulatore Nicolet NICE-88 di microprocessore Intel 8088.
- Un emulatore ZAX di microprocessore Motorola 68000.
- Tutto il materiale descritto è ovviamente accompagnato dal relativo software di controllo, in particolare per quanto riguarda l'acquisizione e la restituzione dei segnali audio, da files su disco e viceversa; è inoltre disponibile una vasta biblioteca di microprogrammi per il TMS32020.

Infine, sul microVAX si dispone di una gran quantità di software applicativo, scritto localmente o importato da altri centri di ricerca con cui il Laboratorio collabora, soprattutto per l'analisi automatica o interattiva del segnale vocale e per la sintesi della voce e musicale.

Lo staff del Laboratorio è attualmente composto da 6 persone (due fisici, un ingegnere, un linguista, un tecnico elettronico ed un musicista), oltre ad alcuni laureandi in fisica che collaborano nel quadro dello svolgimento della loro tesi di laurea.

Il Laboratorio ha rapporti di collaborazione, già attivi o in via di formalizzazione, con i seguenti istituti o centri di ricerca:

- IRCAM - Institut de Recherche et Coordination Acoustique-Musique, Parigi;
- LAFORIA - Laboratoire des Formes et Intelligence Artificielle - Université P. et M. Curie, Parigi;
- ITR - Istituto Tempo Reale, Firenze;
- SIM - Società di Informatica Musicale, Roma;
- CSC - Centro di Sonologia Computazionale, Padova;
- LIMSI - Laboratoire d'Informatique pour la Mécanique et les Sciences de l'Ingenieur - CNRS, Orsay;
- Dipartimento di Fisica dell'Università di Roma - La Sapienza;

- Dipartimento INFOCOM dell'Università di Roma - La Sapienza;
- ILC - Istituto di Linguistica Computazionale - CNR, Pisa;
- Dipartimento di Linguistica dell'Università di Pisa;
- Istituto di Linguistica dell'Università di Venezia;
- Dipartimento di Automatica e Informatica del Politecnico di Torino.

Il Laboratorio è inserito nel progetto comunitario Prometheus (parte del progetto Eureka), sottoprogetti PRO-ART e PRO-CHIP, con un programma di ricerca settennale per la realizzazione di un sistema di dialogo orale tra guidatore ed autoveicolo; il programma comprende tra l'altro la concezione, il progetto e la realizzazione di una famiglia di circuiti VLSI semicustom, per un'implementazione integrata su silicio del sistema totale (che verrà inizialmente realizzato con prototipi hardware a componenti discreti).

Il Laboratorio partecipa inoltre a due Progetti Finalizzati del CNR ("Robotica" e "Sistemi informatici e calcolo parallelo"), con programmi di ricerca quinquennali su sistemi e componenti (hardware e software) di trattamento numerico del segnale in tempo reale, per applicazioni musicali e non.

Per quanto riguarda gli aspetti più musicali dell'attività del Laboratorio, l'attenzione è rivolta attualmente su due campi principali:

- concezione e realizzazione di firmware di controllo in tempo reale di circuiti dedicati per sintesi musicale elettronica (il lavoro implica tra l'altro una ricerca su opportune strutture di dati e adeguate strategie di controllo, nonché di protocolli di comunicazione tra coprocessori);
- tecniche e metodologie di organizzazione e controllo di "software tools" per ricerca e produzione musicale.

Inoltre è attualmente in via di definizione, tra l'altro, un accordo con l'ITR di Firenze per un piano di ricerca comune, basato per ora sulle seguenti linee:

- strategie di controllo spaziale del suono con sistemi a molti canali;
- algoritmi di analisi spettrale a breve termine di suoni concreti;
- algoritmi di filtraggio dinamico di suoni naturali sulla base di uno o più modelli vocalici.

RECENT WORK IN AI & MUSIC WITH LINES FOR THE FUTURE

P. Beyls

Artificial Intelligence Lab - Vrije Universiteit Brussel Pleinlaan 2 - Brussels

Survey of activities.

Much decision making requiring human intelligence can now be handled by a machine in many problem domains, including music. Tools and ideas from current work in artificial intelligence (AI) have proven to be of extreme value for music in two ways: from the engineering - and from the cognitive points of view. Four sub-fields can be observed: composition, performance, problems of music theory and digital sound processing.

Our lab does fundamental research in the following topics: knowledge engineering, complex dynamics and massive parallelism, autonomous agents and knowledge representation. In addition, our media group explores the potential of symbolic computation for musical as well as visual (computer graphics) applications.

It is my belief that such a scientific environment can be a productive and stimulating one for an artist. AI aims, amongst other things, to build models of mental activity in order to study various modes of thinking, the artist might use these models to produce works of art. In fact, the artist delegates aesthetic responsibilities to a machine. AI offers excellent tools to probe into those aspects of aesthetic decision making that remain elusive or not fully understood. AI provides methodological strategies and as well as a technological infrastructure to explore ideas and "play with concepts" in a tangible way (Beyls 88).

Types of musical activity at our lab include: composition (Steels), intelligent tutoring (Cuypers) and performance (Beyls). Some introductory comments are given for all sub-fields but performance oriented work is described in a little more detail.

Steels is primarily interested in machine learning related to problems of musical intelligence. (Steels 86b) reports on the acquisition of compositional skills and the application of machine learning techniques to derive heuristic rules from a set of constraints which express deep musical knowledge. Actual learning experiments are in the field of classical tonal harmony.

Also, Steels investigates methods for understanding and simulating the various processes of reasoning that takes place while someone is composing music. This objective resulted in a program, written in KRS (Steels 86a), which aims to represent both composition processes and actual musical fragments from the musical idiom expressed by Mozart. In "Mozart Studies", a few musical themata from Don Giovanni and Requiem were taken as basic material. Mozart's own compositional techniques were used to further elaborate and actually compose a new and original piece out of this basic material.

The borrowed sequences were expressed as facts, while Mozart's composition techniques were represented as procedural knowledge. Remarkably, some typically Mozart-sounding-sequences were generated although there was no explicit indication in the program to do so (musical example one).

This work was carried out on a Symbolics Lisp machine controlling a TX-816 through a Hinton midi interface.

Cuypers, who also designed a music editor on the Lisp machine (Cuypers 86), is currently starting research in the field of intelligent musical tutoring.

The objective here, is to design a system which acts as a good teacher would to a student for the specific problem domain of music. This system exhibits intelligence in the sense that it is adaptive both towards the individual characteristics of the student and towards the evolution of the learning process itself. In other words, a solution is searched for to communicate knowledge on an individual basis. Cuypers adopted to tackle the problem of teaching musical harmony. This research is fully documented in (Cuypers 87).

Performance.

Beyls' current research has its roots in performance, i.e. the study and subsequent modelling of mind/body activity in the process of musical performance. However, this work includes research in the following related areas: the creation of responsive instruments, the analysis, classification and exploration of musical structures generated both by humans and machines and, the generative modelling of creativity-driven in musical behaviour.

In order to study man-machine interaction and the potential of augmented complexity in musical processes in which responsibilities are shared by both the performer and the program, the author designed two virtual musical personalities. These musical characters are largely modelled after knowledge and strategic behaviour borrowed from myself. It would be correct to say that they are highly interactive and conversational and real-time expert systems. The expertise - i.e. the musical knowledge required to solve a structural problem or to react in a consistent way given a local musical context - is borrowed from examples provided by a human performer, in our case the person who designed the system.

In other words, characters are modelled - among other things - after knowledge, attitudes, stylistic tendencies and belief systems of what we assume constitutes exciting musical behaviour.

The way musical knowledge is represented in the system is a most crucial design consideration because the whole must function in real-time. Both factual (i.e. for example a scale or modus) and procedural knowledge (i.e. for instance building chords from basic notes or interpolating between structures in time) are available in the system. Furthermore, because of temporal constraints, we have adopted a dual memory organization, that is, both in terms of long (LTM) and short term (STM) memory.

LTM is general in nature, exhibits approximative definition and is self-organizing in the sense that it knows what to do if memory overflow occurs. STM has absolute fidelity, is very small but holds details about recent history in the behavioural data stream. STM is not self-organizing but simply keeps the most extreme stimuli collected so far and as decided on the spot.

A key problem is to design a concise model of intelligent man-machine interaction in the problem domain of music. It is clear that a traditional stimulus-response model is of no use here. Actually, our work develops along a path of increasing complexity which is somehow reflected in the following list:

- designing responsive instrument/program relationships;
- the design of an adaptive system sensitive and reactive to changing contexts;
- building a system which exhibits musical intelligence and a "personal drive" in relation to (or the absence of) external stimuli;

— the ultimate goal: building a learning system, that is, the design of a personality that learns from examples provided by humans or machines and capable of the perception and interpretation of its own output. An other obvious idea is to implement learning based on observation and discovery; the program classifies perceived stimuli and tries to build new concepts in doing so.

Another problem is the design of physical sensors that relate directly to the topology of the human body and the programs that extract relevant information from incoming - both vocal and tactile - sensory data.

The performer and the machine are in a continuous loop of information interchange: they talk and listen to eachother in an a-priori abstract way. Consider the following question/answer model:

- perceive and store data if available and useful;
- if nothing happens, try to invent a problem/activity by yourself;
- do feature extraction from patterns if available;
- be allert to changing contexts and keep track of these changes;
- apply rules for interpretation of collected information;
- explore all available data sets at all times, report if anything interesting is detected;
- formulate an answer which is relevant to the question - in other words, formulate a statement which bears coherence to the current state of affairs.

It is clear that the same process is taking place at both sides of the screen though, the programs are only on the edge of being intelligent in the sense that they are certainly responsive and largely adaptive. Also, we are learning a lot from the very nature of implementing ideas in programs in an interactive computational environment. Machine intelligence can be studied in a modular, well structured way because activities can be observed as both isolated and context dependent phenomena.

On the other hand, human musical intelligence is much harder to study as it involves too many processes we hardly understand and space/time variables that we know too little about. The creation and observation of artificial characters allows for systematic and progressive refinement of AI models for musical activity. However, this type of work will make knowledge available of know human apply musical knowledge in real time performance and composition which is still the inevitable bottle neck of current work in this field.

Future work.

Up to the present, many different procedures for collecting, analyzing and inventing musical activity have been tested and implemented in OSCAR (OScillator ARtist) (musical example). Time has come for full implementation of the most promising models - those which represent some musical character in a unique way.

Such a model can be inspired on more general behavioural models and should be flexible enough to tailor highly personal musical individuals. The seeming conflict between general open-endedness and precise characterisation raises many important and difficult problems while designing such programs.

A great many examples of models which describe intelligent activity in living systems

- natural and artificial - are found in the work of both Morris and Capra. Models from both sociology and physics provide hooks to describe musical activity. For instance, Morris suggests many ideas for stimulus evaluation, goal oriented and exploratory behaviour, and discovery while playing games. Ideas from this field have already influenced the design of OSCAR heavily.

Another idea is to have a number of "musical individuals" performing together - possibly in a network with several processors.

Still another idea is to model a composer-performer, that is to add a sort of "architectural module" which does planning in time. The mutual relationship between longterm structuring and "decision making on the spot" during actual performance is an exciting problem. The program LOUISE is an 8 - handed performer playing the piano. The idea is to use message passing between the hands - so hands can exchange intervals or - in more general terms - hands can borrow stylistic information from each other. LOUISE is partially operational and will be able to behave as fully independent musician.

Gestural control structures:

Recent years have witnessed a boost in research on alternative control strategies as well as new physical control devices. These new controllers are tailored to the topology of the body and the associated software maps bodily activity onto some musical process. As such, the music is guided by the expression of physical energy by the performer. The development of complex and extremely subtle control mechanisms become not only possible, but a domain of research in itself. The intimate relationship between sound and gesture - obvious and available for virtuoso exploration within the world of acoustic instruments - opens fascinating venues for the performing artist using digital electronics.

Experimentation at the lab proves that adaptation of control mechanisms borrowed from conventional, acoustic instruments yields interesting results. The "modified violin" is one such an instrument. However, also more exotic alternatives are investigated. These include the use of picture recognition techniques with a real-time video digitizer to trace the human body (Beyls) and the use of sub-sonic waves to locate an "intelligent conductor's baton" in 3-d space (Cuypers). These projects involve learning techniques since the interpreting program has to get to know the expressive range of the performer and his/her abilities to deal with the basic problem of controlling many musical parameters through the interactive specification of higher-level parameters.

References.

STEELS — *The KRS concept system*. Luc Steels, VUB-AI Lab Technical Report, 1986.

STEELS — *Learning the craft of musical composition*. Luc Steels, Proceedings of the International Computer Music Conference, Den Haag, 1986.

BEYLS — *Discovery Through Interaction*. Peter Beyls, paper presented at the Intelligent Images Forum, IMAGINA, Monte Carlo, 1988.

CUYPERS — *Een Grafische Musiekinterface*. Ludo Cuypers, Licentiaatsthesis, Vrije Universiteit, Brussel, 1986.

CUYPERS — *Intelligence Tutoring Systemen per Computer*. Research proposal for INOWL, 1987.

TELETAU

UN SOFTWARE PACKAGE PER L'INFORMATICA MUSICALE

L. Camilleri, F. Giomi, P. Grossi, M. Ligabue, G. Nencini
Divisione Musicologica CNUCE/C.N.R. - Conservatorio di Musica "L. Cherubini"
Piazza delle Belle Arti 2 - Firenze

Introduzione.

Dal 1980, la Divisione Musicologica del CNUCE, Istituto del CNR, ha iniziato una serie di studi sulla telematica musicale, con l'obiettivo di mettere a disposizione di utenti locali e remoti il suo patrimonio di software e il suo archivio digitale di brani musicali.

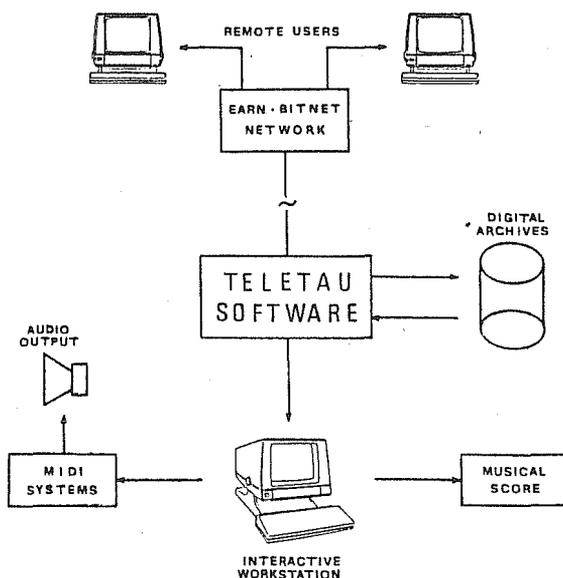
Il TELETAU è strutturato in modo tale da rendere facile l'espansione dei suoi comandi specialmente quelli che riguardano la generazione di strutture musicali e l'analisi musicale.

Le sue possibilità operative riguardano i seguenti campi di applicazione:

- codifica di brani musicali;
- generazione automatica di strutture musicali;
- elaborazione di brani musicali;
- gestione dell'archivio di brani musicali;
- analisi musicale.

Le sue modalità operative prevedono due differenti modi di accesso:

- 1) accesso locale,
- 2) accesso remoto.



Accesso locale.

L'accesso locale consente di utilizzare il sistema TELETAU in modo interattivo.

Il software lavora con file di particolare tipo. Il tipo SOURCE identifica un file che contiene un brano musicale scritto nel sistema di codifica del TELETAU. Il tipo OBJECT può identificare sia la traduzione numerica di un file di tipo SOURCE che un testo creato con gli algoritmi di generazione automatica. Il tipo RUN contiene una lista di comandi che verranno eseguiti sequenzialmente.

Un tipo di lavoro interattivo prevede l'uso di un insieme di comandi per la lettura, memorizzazione, elaborazione, ed esecuzione di testi musicali, per la generazione automatica e l'analisi musicale.

Molti comandi prevedono delle opzioni che restringono l'applicazione delle trasformazioni in sezioni del brano, su singole voci o su singoli parametri.

I comandi possono essere suddivisi nelle seguenti categorie:

- elaborazione di pezzi musicali (DUPLICA, GOBACK, INVERT, MODIFY, REMAKE, SCALE, SHUFFLE, STACCATO, TEMPER);
- esecuzione (PLAY);
- gestione dell'archivio (CHAIN, EXCHANGE, INSERT, LOAD, SAVE, SEARCH, SOURCE, TYPE);
- analisi musicale (ANL1, ANL2, ANL3, ANL4);
- generazione automatica (CREATE, JAZZ, WORK);
- controllo dei parametri di sintesi del suono (GENERATE, SENDMIDI, SETMIDI).

Accesso remoto.

Questo tipo di utilizzo permette ad ogni utente delle reti BITNET-EARN (che coprono l'Europa Occidentale, l'America del Nord, Israele e alcuni paesi asiatici) di accedere a questo servizio di Informatica Musicale. Il TELETAU è attivo sulle macchine virtuali CHERU@IFIIDG e MUSIC3@ICNUCEVM.

La sua principale caratteristica è il suo ambiente "Free-Running", al quale si può accedere per mezzo di una particolare procedura. Una volta in tale ambiente, il TELETAU è in grado di accettare, sotto forma di file, comandi provenienti da qualsiasi utente della rete. Il TELETAU esegue i comandi ricevuti ed invia automaticamente all'utente remoto i risultati delle elaborazioni. L'utente remoto ha anche la possibilità di consultare ed utilizzare l'archivio digitale di brani musicali.

Una volta in questo ambiente si attende l'arrivo dei file provenienti dagli utenti della rete. Il sistema esamina il tipo dei file inviati e la loro provenienza. Se il file è di tipo OBJECT o SOURCE viene caricato su disco e aggiunto all'archivio di brani musicali. Se si tratta invece di un file di tipo RUN (tipo che specifica un file contenente una lista di comandi), vengono eseguiti i comandi presenti e l'eventuale risultato viene mandato all'utente remoto.

Gestione di sistemi MIDI.

Il TELETAU non è vincolato ad uno specifico sistema di sintesi del suono. I file conte-

nenti brani musicali o particolari elaborazioni devono essere tradotti nel codice del sistema di sintesi del suono usato dall'utente remoto.

Per quanto riguarda il lavoro locale, è stata realizzata una speciale interfaccia Midi pilotabile dall'uscita seriale di un personal computer. Attraverso questo sistema si può pilotare fino a sedici voci indipendenti in qualsiasi parametro del suono.

Inoltre possono essere inviati in maniera completa i codici relativi al sistema esclusivo di qualsiasi sintetizzatore Midi.

La versione presente presso la Divisione Musicologica del CNUCE al Conservatorio di Musica L. Cherubini di Firenze, utilizza un collegamento remoto via linea dedicata con un mainframe operante in VM/370. Un personal computer IBM viene usato come terminale intelligente permettendo sia il colloquio con l'host che la gestione dei sistemi di sintesi Yamaha (2 TX81Z, CX5M, DX7) mediante la speciale interfaccia Midi.

Conclusioni.

Il TELETAU presenta caratteristiche operative che possono interessare gli aspetti compositivi, la didattica musicale e la musicologia.

A questo proposito è stato utilizzato nel Corso di Informatica Musicale del Conservatorio di Musica L. Cherubini di Firenze.

Inoltre il TELETAU viene utilizzato per la realizzazione di composizioni assistite strumentali, composizioni elettroniche e installazioni.

Bibliografia.

CAMILLERI L. et al. — *A Software Tool for Music Analysis*. INTERFACE, 16.

GIOMI F., LIGABUE M. — *An Interactive System for Musical Improvisation*. (In corso di stampa).

NENCINI G. et al. — *TELETAU - A Computer Music Permanent Service in Proceeding of 1986 International Computer Music Conference*. P. Berg (ed.), San Francisco, Computer Music Association.

NENCINI G. — *Modalità Operative del TELETAU (release 2)*. Pisa, CNUCE, 1988.

ELABORAZIONE DEI SUONI NATURALI E COMPOSIZIONE CON IL SISTEMA 4I

S. Sapir

C.S.C. - Centro di Sonologia Computazionale - Padova

INTRODUZIONE.

Il sistema 4I disponibile presso il C.S.C. dell'Università di Padova ha ormai superato la sua prima fase di sviluppo (AA.VV. 1984, Azzolini Sapir 1984, Sapir 1985-1987, Sapir De Poli 1983, Sapir Vidolin 1985). È stato integrato nell'insieme delle risorse del centro, e viene usato per la produzione musicale e la didattica dell'informatica musicale. Gli ultimi sviluppi del sistema, che verranno presentati in questa relazione, riguardano per prima cosa la realizzazione, da G. Di Giugno, di due canali di acquisizione di segnali analogici, che potenziano le possibilità di elaborazione sonora del processore 4I; e in secondo luogo il software applicativo del sistema che è stato arricchito di due programmi per l'aiuto alla composizione, utilizzabili essenzialmente a scopi didattici, che completano le possibilità già numerose dell'intero sistema.

TRATTAMENTO DEL SUONO DAL VIVO CON IL SISTEMA 4I.

L'elaborazione numerica dal vivo dei suoni naturali procede in modo ben diverso da quella dei suoni sintetici. Si tratta di lavorare con dei segnali complessi, già costruiti, e di cui non si conosce né la struttura interna, né il principio di sintesi, al contrario dei suoni sintetici interamente determinati dal loro algoritmo generatore. La manipolazione dei suoni naturali coinvolgerà in genere i loro parametri percettivi; ciò implica degli interventi che dipendono notevolmente dal principio di causalità. Ad esempio è impossibile riprodurre in tempo reale una frase musicale a rovescio, visto che non se ne conosce ancora la fine.

Negli studi elettro-acustici, il trattamento elettronico del suono può essere diviso in tre categorie: trasformazione, selezione, regolazione (Reibel e Pacquetteau 1972, Haller 1984). Vedremo, secondo questo schema, quali operazioni sono state implementate sotto forma di moduli nella biblioteca del sistema 4I e qualche esempio di strumento costruito per la realizzazione dei pezzi musicali di M. Graziani e W. Prati e di M. Sambin.

Trasformazione.

Le trasformazioni del segnale sonoro permettono di modificare alcune delle sue caratteristiche percettive. La modificazione dello spettro è ottenibile tramite modulazione ad anello o tramite filtraggio; la modifica delle durate tramite riverberazione o uso del loop; la stratificazione delle voci in un insieme polifonico tramite accumulazione, accumulazione e ritardo, effetto coro; la variazione delle dinamiche tramite mixaggio o spazializzazione; la trasposizione in altezza tramite variazione della velocità di lettura del suono. I moduli base che consentono queste manipolazioni sono i moltiplicatori, gli oscillatori, le linee di ritardo ed i sommatore.

Esistono nelle biblioteche del sistema 4I dei moduli complessi che consentono di effet-

tuare delle trasformazioni elementari sul segnale; verranno brevemente elencati qui di seguito:

- MIXE: unità di mixaggio a 24 ingressi.
- SPCE: unità di spazializzazione quadrifonica.
- OINT: oscillatore ad interpolazione.
- DELY: linea di ritardo (4 secondi al massimo).
- LPFO: filtro LP del primo ordine.
- FLT2: filtro universale del secondo ordine.
- CMBV, CMBZ, ALPV,: filtri “Comb” e All Pass.

Selezione.

Le operazioni di selezione consentono di estrarre dal segnale elementi significativi per il controllo dell'elaborazione. Necessitano dei meccanismi di analisi spettrale o temporale. I moduli utilizzati sono in genere dei banchi di filtri “passa-banda”, dei rilevatori di altezza, dei rilevatori di inviluppo, dei rilevatori di soglia. Abbiamo realizzato i moduli seguenti.

- DISC: discriminatore in funzione di una soglia.
- DDIS: doppio discriminatore.
- ENVF: rilevatore di inviluppo di ampiezza.
- PITD: rilevatore di altezza.

Regolazione.

La regolazione riguarda i molteplici controlli delle trasformazioni elencate sopra, in funzione di informazioni rilevate dagli operatori di selezione. Si tratta quindi di utilizzare delle porte sensibili all'ampiezza (gate), di moduli logici per la decisione (ricerca di un massimo o di un minimo, sample & hold ecc.), di meccanismi automatici di controllo spaziale. Sono disponibili i moduli seguenti:

- MAXO: massimo fra due ingressi.
- MINO: minimo fra due ingressi.
- NGAT: noise gate con trigger di schmidt.
- AUTO: controllo automatico di guadagno.
- SWIT: commutatore fra due segnali.
- SAHO: sample & hold.
- STSP: regolazione automatica dello spazializzatore SPCE.

Questi moduli sono spiegati nel manuale operativo del programma 4I disponibile presso il C.S.C. e in Sapir 1987.

Esempi di algoritmi per l'elaborazione di suoni naturali.

Sono stati riportati sulle figure 1 e 2 due esempi di algoritmi che sono stati implementati

sul processore 4I per elaborare suoni strumentali. Il processore 4I dispone di un numero limitato di risorse, questo spiega la possibilità di disporre di solo 3 oscillatori ad interpolazione, o la possibilità di campionare al massimo 4 secondi di suono. Per ulteriori dettagli vedere Graziani 1988.

AMBIENTI DI AIUTO ALLA COMPOSIZIONE.

La possibilità di sintetizzare dei suoni in tempo reale può anche essere sfruttata per l'aiuto alla composizione. Il musicista può ascoltare direttamente il risultato sonoro di partiture elaborate in tempo differito da programmi compositivi. Ciò equivale quasi al ruolo del pianoforte o di un altro strumento polifonico, come supporto sonoro nella fase di elaborazione della composizione.

Abbiamo quindi adattato per il sistema 4I, due programmi di aiuto alla composizione: POD di Barry Truax, Truax 1973-1977-1978-1985, e PROJECT 1 di Gottfried Michael Koenig, Koenig 1970, 1983. In entrambi i casi, si è trattato di trasportare sul PDP-11 i programmi di composizione, di adattarli al sistema operativo RT11SJ, e di realizzare un'interfaccia per il processore 4I, che consenta di eseguire direttamente le partiture elaborate con l'aiuto di questi programmi.

Pod.

POD è un programma dedicato alla sintesi dei suoni e alla composizione musicale.

Esistono più versioni di questo programma; abbiamo scelto di installare la versione POD6X, già implementata per il processore audio-numeric tempo reale: il DMX-1000, Walraff 1979. Il lavoro è stato realizzato in collaborazione con F. Cappello e G. De Poli ed è stato presentato alla mostra di musica elettronica organizzata dalla Biennale di Venezia: "La Nuova Atlantide", nel Novembre 1986 a Venezia, Cappello e De Poli 1986.

Pod è un programma interattivo che consente di controllare nei loro dettagli i parametri della sintesi, e di dare delle direttive generali per lo svolgimento della composizione che sfruttano dei processi pseudo-aleatori. La scelta di questi algoritmi e in particolare la distribuzione di Poisson (POD = POisson Distribution) deriva da esigenze teoriche ed estetiche dell'autore. La sintesi sonora è effettuata dal processore 4I. Quest'ultimo è programmato per disporre di più voci "strumentali" FM indipendenti (6 per la versione equivalente al DMX-1000, e 16 per la versione definitiva in corso di realizzazione).

POD consente di elaborare la composizione lavorando nel modo seguente:

- definire degli oggetti sonori secondo le modalità previste dalla tecnica FM (involuppo di ampiezza, ratio C:M, indice massimo di modulazione, numero dell'involuppo associato all'indice di modulazione);
- selezionare alcuni oggetti sonori;
- attivare il calcolo della distribuzione di Poisson;
- eseguire un estratto della composizione secondo alcune variabili di esecuzione quali velocità, senso dell'esecuzione, riscaldamento degli involuppi ecc..

Tutte queste operazioni sono interattive e possono essere accompagnate da visualizzazioni grafiche o numeriche. Durante la fase di definizione degli oggetti sonori, è possibile aiutarci nella scelta dei parametri "suonando" direttamente con il processore 4I grazie alla tastiera del terminale. Inoltre le partiture create possono essere manipolate da altri programmi (editing e mixaggio) per creare delle partiture più complesse che possono eventualmente essere sintetizzate in tempo differito.

Il lavoro più grosso è stato l'adattamento dello scheduler alla logica di funzionamento del sistema 4I. Questa parte è stata scritta in MACRO-ASSEMBLER, e gestisce in tempo reale l'insieme dei dati strutturati in linked list.

L'esecuzione della partitura deve rispettare i tempi e l'evoluzione interna dei suoni (involuppi). La polifonia è gestita come indicato dall'algoritmo seguente:

QUANDO una nuova voce deve essere attivata

SE numero totale delle voci attive = MAX

ALLORA SE una voce sta finendo l'involuppo

ALLORA questa voce è interrotta

ALTRIMENTI la nuova voce è posticipata alla liberazione di una voce

ALTRIMENTI la nuova voce è attivata.

L'interattività, le possibilità di ascolto immediato, l'approccio relativamente facile fanno di POD un programma attraente. Inoltre POD è concepito in modo molto didattico, è dotato di un tutorial sulla modulazione di frequenza con lezioni ed esercizi auto-corretti. In definitiva POD (versione 4I) risulta adatto per una buona introduzione alla sintesi sonora e per la composizione.

Project 1.

PROJECT 1 è stato realizzato da G.M. Koeing verso la fine degli anni 60. È dedicato alla composizione della musica strumentale secondo procedimenti di serializzazione musicale provenienti dalla "scuola di Colonia". PROJECT 1 ha segnato la storia della musica contemporanea, perché è uno dei primi programmi che ha formalizzato un processo di composizione - si tratta ovviamente di composizione automatica.

PROJECT 1 compone selezionando alcuni parametri musicali (strumenti, ritmo, altezza, dinamica ecc..) secondo principi pseudo-aleatori. Le regole di selezione dipendono dai processi di serializzazione implementati. Il programma consente di esplorare un gran numero di varianti di uno stesso piano di composizione generale. Vengono generate delle "partiture possibili" della composizione, materiale base da cui possono essere tratti (a mano, secondo il "gusto" dell'autore) dei pezzi che saranno "interpretati", organizzati e trascritti in una vera partitura simbolica da suonare con strumenti tradizionali.

Nel nostro caso la trascrizione è facilitata dalla possibilità di fare eseguire parti del materiale base dal processore 4I con un'orchestra di 16 strumenti FM a timbrica variabile. Quest'adattamento di PROJECT 1 è stato realizzato per la mostra la "Nuova Atlantide", abbiamo quindi cercato di preservare gli aspetti storici del programma e il modo operativo dello studio di Utrecht.

PROJECT 1 tratta i seguenti parametri: strumento, entry delay, altezza, registro di ottava e dinamica. Questi parametri sono fissati per default, ma possono essere definiti dal com-

positore prima dell'esecuzione del programma. Una matrice (Branching Table) consente di selezionare il "grado di regolarità" caratteristico di ciascun parametro. I gradi di regolarità sono numerati da 1 a 7: per il grado 1, il parametro varia in molto irregolare, mentre per il grado 7 varia molto regolarmente.

Gli strumenti sono rappresentati da numeri, nella fase di trascrizione il compositore li associerà a strumenti reali.

Il ritmo è definito in funzione degli entry delay e del fattore metronomico.

L'armonia è basata sulla scelta d'intervalli particolari e dei registri di ottave (da 1 a 9). Il programma elabora dei gruppi di note che formano degli accordi secondo una serie dodecafonica completa.

La dinamica costituisce un altro tipo di materiale da serializzare; i livelli dinamici variano da ppp (1) a fff (8), e sono associati ad ogni accordo.

Il programma memorizza la partitura codificata su file, questo file è ripreso in un secondo tempo, decodificato in tempo reale, e le note vengono eseguite dai 16 strumenti a modulazione di frequenza.

Vale la pena notare il modo particolare di trattare gli involucri di ampiezza e di indice di modulazione. Gli involucri sono fissati in una tabella che verrà letta una volta per ogni nota. PROJECT 1 non tratta le durate in modo esplicito, è quindi stato deciso di associare ad ogni strumento un proprio timbro ed un'evoluzione temporale particolare che lo caratterizzerà per tutta l'esecuzione della partitura. La scelta degli involucri determinerà quindi l'articolazione temporale della partitura eseguita. Questi parametri timbrici possono essere definiti prima dell'esecuzione in modo interattivo.

La gestione dell'esecuzione è quindi relativamente semplice, visto che si tratta di aggiornare i parametri "statici" (portante, modulante, fattore di ampiezza, e durata): la tecnica utilizzata è il polling. L'esecuzione non prevede la possibilità di intervenire gestualmente. La polifonia viene gestita grazie ad una struttura dati (FREE) organizzata ad anello che rappresenta gli strumenti liberi. Ogni volta che un timer indica la fine di una nota, uno strumento è liberato e si inserisce nell'anello FREE puntato dal puntatore FREEFST. Ogni volta che una nota deve essere attivata, il numero dello strumento libero che può suonare questa nota è prelevato dall'anello nella posizione puntata da FREELST. Questo ciclo viene eseguito fino all'esaurimento delle note della partitura.

CONCLUSIONE.

Il sistema 4I è disponibile presso il C.S.C. dal 1982, dall'inizio del suo sviluppo fino ad oggi ha consentito la realizzazione di più di 10 pezzi musicali, ed è inoltre stato uno strumento di grande aiuto per la didattica. Si è sempre cercato di inserirlo in ambienti diversificati, dove il tempo reale copriva solo un'aspetto delle possibilità effettive del sistema. Può quindi essere considerato come strumento da concerto, strumento di prove in vista di elaborazioni più raffinate in tempo differito, come strumento didattico e di ricerca. L'ultimo progetto in atto a suo proposito riguarda la sua simulazione sul VAX 8600 dell'Università nell'ambiente di sviluppo 4X. Lo scopo di questo progetto è quello di sviluppare un micro-programma 4I, integrarlo nelle ROM del processore 4X, per disporre eventualmente di sei schede 4I contemporanee, con il vantaggio dell'esperienza già acquisita e dei programmi disponibili per il controllo di questo processore.

Bibliografia.

- AA. VV. — *Bollettino LIMB 4*. Biennale di Venezia, 1984.
- AZZOLINI F., SAPIR S. — *Score and/or gesture: A real time system control for the digital sound processor 4I*. Proc. of the 1984 International Computer Music Conference, 25-34, 1984.
- CAPPELLO F., DE POLI G. — *Note sull'implementazione del sistema 4IPODX*. Rep. CSC, Univ. di Padova, 1986.
- GRAZIANI M. — *Note sulla realizzazione di "Aquam flare in media labia tua" per trombone e sistema 4I dal vivo*. Atti del 7° Colloquio di Informatica Musicale, Roma, 1988.
- HALLER H.P. — *La tecnica del "live electronic" allo studio sperimentale di Friburgo*. In Verso Prometeo, L. Nono, eds. Biennale di Venezia-Ricordi, Milano, 36-46, 1984.
- KOENIG G.M. — *Project 1. Electronic Music Reports 2*. Istitut voor Sonologie, Utrecht, 1970.
- KOENIG G.M. — *Integrazione estetica di partiture composte mediante elaboratore*. Bollettino LIMB 3, Biennale di Venezia, 27-34, 1983.
- REIBEL G., PACQUETEAU J.M. — *Les étapes de la réalisation en musique électro-acoustique*. Musique en jeu, 8 Editions du seuil, Paris, 29-35, 1972.
- SAPIR S. — *Computer music in tempo reale*. Strumenti musicali, Gruppo editoriale Jackson, Milano, 64, 103-108; 65, 75-81; 1985.
- SAPIR S. — *Processeurs audio-numériques temps réel et informatique musicale: mise en oeuvre d'un système d'exploitation et utilisations pour la recherche, la production et la pédagogie musicales*. Thèse de Doctorat d'Etat. Université d'Aix-Marseille II.
- SAPIR S., DE POLI G. — *Verso MUSIC V in tempo reale: Un software per il processore numerico di suoni 4I*. Atti del 5° Colloquio di Informatica Musicale, Univ. di Ancona, 24-35, 1983.
- SAPIR S., VIDOLIN A. — *Interazioni fra tempo e gesto*. Bollettino LIMB 5, Biennale di Venezia, 25-33, 1985.
- TRUAX B. — *The computer composition-sound synthesis programs POD4, POD5 and POD6*. Sonological reports 2, Istitut voor Sonologie, Utrecht, 1973.
- TRUAX B. — *The POD System of interactive composition programs*. Comput. Music J, 1, 3, 30-39; 1977.
- TRUAX B. — *Computer music composition: The polyphonic POD System*. IEEE Computer 11, 8, 40-50; 1978.
- TRUAX B. — *The PODX system: Interactive compositional software for the DMX-1000*. Comput. Music J. 9, 1, 29-38; 1985.
- WALRAFF D. — *The DMX-1000 signal processing computer*. Comput. Music J. 3, 4, 44-69; 1979.

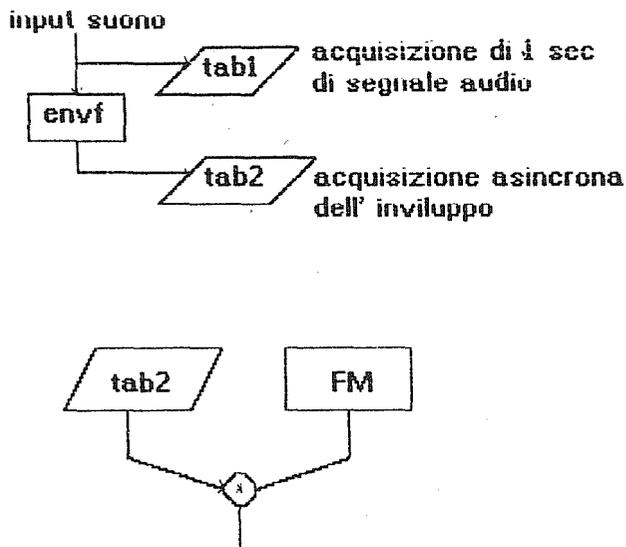


Fig. 1 - Acquisizione asincrona di 2 segnali e uso del 2° segnale come controllo di ampiezza di un blocco di sintesi in modulazione di frequenza.

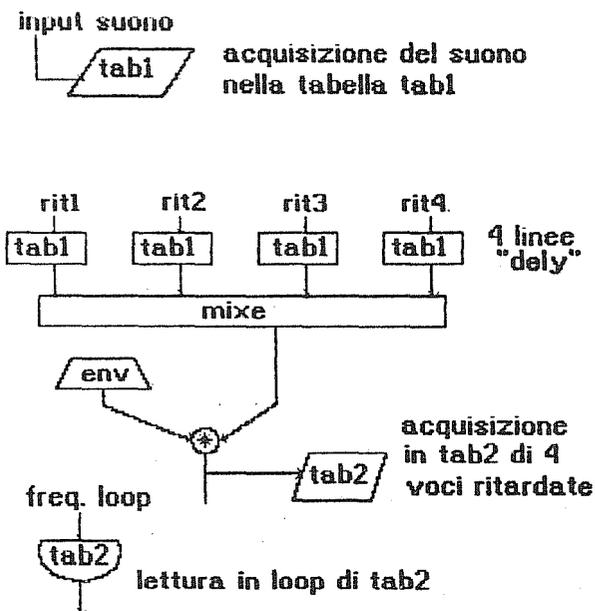


Fig. 2 - Stratificazione di 4 voci ritardate inserite in "loop".

CONSIDERAZIONI SUI PRIMI QUATTRO ANNI DI ATTIVITA' DELLA SIM NELL'INFORMATICA MUSICALE

N. Sani
SIM Srl - Roma

La presenza di specifici interventi sulle attività della SIM mi spinge a realizzare un discorso di sintesi sulle attività della struttura in questi primi quattro anni della sua costituzione e sull'attività attuale in campo musicale.

SIM (sigla di Società Informatica Musicale) nasce all'inizio dell'84 come struttura privata (una Srl) da un gruppo di persone già attive nell'informatica musicale (Del Duca, Galante, Lupone, Nottoli, Sani). La struttura associativa all'epoca era puramente nominale ed aveva la stessa sigla (SIM - Studio per l'Informatica Musicale). Con quella sigla erano già state effettuate pubblicazioni sul progetto di un primo sistema per la sintesi del suono (a cui aveva partecipato anche Petrarca).

Come società la SIM è stata la prima struttura privata a carattere imprenditoriale nel campo dell'informatica musicale. La prima a considerare questo settore non solo un terreno di incontro e confronto culturale e scientifico, ma anche un ambito di mercato. Nei primi due anni sono stati progettati e realizzati tre sistemi per la computer music basati su DSP TMS 320: Soft Machine, System Fly e SPU 03. Sono i sistemi che vengono oggi utilizzati nei diversi campi di applicazione: Soft Machine per il DSP e la computer music, Fly per la computer music in tempo reale, SPU 03 per la didattica.

Dal 1985 la SIM è know-how center ufficiale della Texas Instruments Italia. Nel 1987 è stato aperto il design center di Roma e la divisione industriale di Monza (MI).

L'attività nel campo dell'informatica musicale si affianca oggi ad una più vasta attività nei campi di applicazione dell'elaborazione digitale dei segnali audio e video e della progettazione di circuiti integrati (ASIC).

Per la parte grafica è stato scelto il dispositivo TMS 34010 (GSP) della Texas Instruments, sul quale sono in fase di studio alcune applicazioni al campo musicale.

Il piano di attività per la computer music è il seguente. Lo Studio per la ricerca audio della SIM è attrezzato per realizzare produzioni musicali in proprio o in coproduzione con terzi. Attualmente è in fase di realizzazione un lavoro di Fausto Razzi in collaborazione con il CSC di Padova.

SIM realizza ricerche al proprio interno o in collaborazione con altri centri di ricerca. Attualmente è parte di un progetto finalizzato presentato al CNR con altri Centri di informatica musicale italiani (CSC, LIM, DIST, CNUCE), per la realizzazione di una "Stazione di lavoro musicale intelligente".

Fornisce attrezzature e supporto per tesi di laurea sull'informatica musicale e per applicazioni del DSP nell'audiovisuale.

Recentemente ha preso il via una ricerca con il Teatro Nuovo di Milano, che intende affrontare, insieme con autori e operatori teatrali, il problema delle nuove possibilità espressive introdotte dall'utilizzazione di tecnologie di elaborazione in tempo reale della voce.

Per quanto riguarda la composizione musicale è stato di recente realizzato un nuovo sistema in ambiente MIDI, che si affianca al precedente MSYS 7 di Giorgio Nottoli. Questo sistema, realizzato da Antonio Pellecchia, è frutto di una collaborazione con il Dipartimento di Informatica e Sistemistica della Facoltà di Ingegneria dell'Università di Roma. Il nome

è TESIMUS e si tratta di un linguaggio di programmazione ad alto livello e procedurale per la composizione di partiture. Permette di costruire procedure per la definizione di eventi musicali complessi o di funzioni di manipolazione della partitura. L'interazione con il musicista avviene mediante l'inserimento di istruzioni del linguaggio (che dispone di un proprio set completo), che vengono immediatamente eseguite. In alternativa si può usare il linguaggio per programmare procedure complesse, che possono essere richiamate interattivamente da disco. Per fare un esempio di un'istruzione del set del linguaggio si consideri l'istruzione ADD. Tale istruzione permette di generare nuovi eventi musicali in funzione degli eventi preesistenti. Questo è ottenuto specificando nell'istruzione le caratteristiche delle nuove note rispetto alle precedenti. Con il costrutto PERFORM è possibile definire una qualsiasi relazione funzionale tra la partitura esistente e quella che ne deriva. TESIMUS consente di gestire fino a 16 canali MIDI diversi, con possibilità di acquisizione dei dati da strumento MIDI.

Per la diffusione della computer music la SIM, collabora attualmente con numerose istituzioni culturali, tra cui Musica Verticale, l'Istituto Tempo Reale di Firenze, il Festival dell'Arte Elettronica di Camerino, Ars Electronica di Linz.

Quest'anno verrà affrontato anche il tema delle installazioni sonore con due progetti di Giorgio Nottoli per Camerino e Napoli (Futuro Remoto).

Per concludere, il panorama descritto mostra una realtà in evoluzione non solo della SIM, ma di tutto il mondo dell'informatica musicale in Italia. In particolare il recente progetto finalizzato, che unisce i vari centri che operano nel settore è il primo passo verso una forma di collaborazione nuova orientata alla compatibilità delle metodologie operative e alla specificità dei diversi organismi. È senz'altro opportuno procedere in questa direzione, anche allargando il numero dei partecipanti e dei progetti, per rafforzare un intervento culturale sull'utilizzazione delle tecnologie avanzate, che sia anche luogo di incontro tra ricerca e mercato.

SESSIONE DIMOSTRAZIONI

COMPOSER 1 : UN PROGETTO DI AMBIENTE COMPOSITIVO PER UNA MINI "STAZIONE DI LAVORO" MUSICALE

F. Degrassi
Via Toti 78 - Bari

Il Composer 1, software realizzato per la scheda di sintesi SPU/03 (prodotta dalla S.I.M. di Roma), si inserisce fra i progetti orientati a configurare piccole "stazioni di lavoro" musicali dedicate alla didattica e alla composizione di computer music.

A monte di questo lavoro ho scelto come quadro di riferimento teorico la constatazione sviluppata in maniera sistematica da S. Mc Adams¹, della "debolezza" che spesso si riscontra in strutture musicali che pure usano singoli suoni sintetici di grande complessità. La mancanza di "coerenza di comportamento" che si ricava dall'ascolto di tali strutture sonore è dovuta - forse - all'ancora insufficiente attenzione dedicata dai compositori (o dalla ricerca in generale) al problema della costruzione di un linguaggio musicale timbrico che consenta di utilizzare pienamente le straordinarie potenzialità sonore del computer senza volere sovrapporgli "griglie" compositive proprie di altri strumenti o complessi strumentali².

Sulla possibilità di costruire una sintassi che concateni gli oggetti sonori in modo rilevabile all'ascolto si gioca il futuro della musica informatica, un futuro-secondo alcuni (e tra questi J.J. Nattiez) - già irrimediabilmente compromesso dalla presunta incapacità del "timbro" di... creare da solo un divenire... "poiché"... fisso nel tempo... "al contrario di ritmo e altezza "... parametri lineari e orizzontali che pretendono un seguito ed esigono imperiosamente dei prolungamenti"³.

Questa ipotesi - che considero errata - deriva forse dalla sovrapposizione alla musica elettronica di una teoria musicale che fa uso di parametri propri della musica "acustica" tradizionale come nota, armonia, melodia... Tale sovrapposizione non è una scelta "neutra" ma - usando parole dello stesso Nattiez - un "... principio trascendente..." che il teorico "... pone alla base del suo edificio e che spesso traduce l'"essenza" del fenomeno considerato, come egli la vede..."³.

Ma se si sostituisce - come paradossalmente sembra suggerire lo stesso Nattiez - alla nozione di nota il concetto di "oggetto sonoro", bisognerà anche - sulla scia di questo vero e proprio "mutamento di paradigma" estetico generato dall'irruzione del rumore nella musica - organizzare "... una selezione data di interpretanti..." del fatto sonoro; e "... poiché il significato di una parola dipende in larga misura dalla sua situazione... un termine che apparentemente ha significato univoco, cambia senso a seconda del contesto di utilizzazione..."³. Una teoria che si basi su una nozione di "timbro" mutuata dalla musica strumentale tradizionale non è in grado - evidentemente - di configurare dimensioni lungo le quali il "timbro" possa essere organizzato, "graduato", in modo da poter generare "orizzonti di attese".

C'è bisogno, invece, di un nuovo sistema di parametri (che riformuli lo stesso concetto di "timbro"), di un nuovo quadro di riferimento teorico⁴.

In questo senso McAdams - partendo dal rilevare come parte del nostro processo di apprendimento della musica sia modellato nel dominio del parlato - delinea dei tratti distintivi del suono che fanno - appunto - riferimento alle strutture psicologiche che sottostanno alla percezione del parlato: portanti (suoni statici come le vocali), transizioni (attacco, decadimento, suoni consonantici), contenuto frequenziale, forma spettrale.

Su queste basi, nei limiti consentiti ad un piccolo sistema che - per la sintesi additiva -

genera 16 armoniche sinusoidali e che per ora opera solo su suoni statici, è fondato il progetto di Composer 1 (distanziandosi quindi parecchio dai composer dedicati, in genere, agli strumenti commerciali). Mediante un editing molto semplice è possibile costruire suoni con 4 tipi di contenuto frequenziale: serie armonica, serie armonica traslata, serie armonica compressa, serie armonica espansa⁵. Una volta generati i suoni è possibile, in tempo reale e in forma estremamente semplice, modificare frequenza e ampiezza dei singoli armonici avendone riscontro da istogrammi, ed eventualmente conservare i valori delle frequenze per la fase compositiva vera e propria. Il Composer richiede - a questo punto - la definizione di frequenze "formanti" dall'ampiezza fissa per ogni suono della composizione mentre gli altri armonici variano continuamente in modo random (limitato da un minimo e un massimo di ampiezza raggiungibile). Così si assicurano delle invarianti compositive (i "formanti") e delle variabili affidate al caso. L'oggetto sonoro varia continuamente di "colore" pur mantenendo un "corpo" ben definito ed è possibile collegare i suoni lungo le dimensioni percettive suggerite dai parametri posti a fondamento (armonicità, inarmonicità, distribuzione dell'energia spettrale). In tal modo la fusione o separazione spettrale, il grado di "asprezza" o "luminosità" del suono, la sensazione di movimento degli oggetti in uno spazio timbrico⁶, sono legate sia ad una scelta soggettiva compositiva che ad una componente casuale (sufficientemente circoscritta): queste caratteristiche-unite alla potenziale economicità - rendono il sistema idoneo sia ad applicazioni didattiche che a momenti compositivi (affiancandogli, magari, un piccolo banco di registrazione multipista).

Note.

1) MC. ADAMS S. — *Qualities and functions of musical timbre*. In Proceeding of the I.C.M.C., 1985.

2) Si giunge per questa strada ad indicare in *Répons* di P. Boulez un modello per la nuova musica che faccia uso di strumenti elettronici.

A proposito di *Répons*: Risset J.C. *Arte e scienza: musica elettroacustica numerica*. 1986.

3) Nattiez J.J. — *Il discorso musicale*. 1982.

4) Su questa strada mi sembra orientata la proposta di "Chant" dell'Ircam che, tuttavia, ha ben poche possibilità di essere diffusa tra i compositori al di fuori di un certo giro. In proposito N. Bernardini *Tecniche di sintesi: Chant*, in *Audio Review* n. 58, 1987.

5) MC ADAMS S. — *Fusione spettrale e creazione di immagini auditive*. In Bollettino L.I.M.B. n. 2, 1982.

6) Mi riferisco al concetto di "spazio del colore sonoro" definito da Slawson W. in *Sound color*, 1985.

ELABORAZIONE INTERATTIVA DEL SEGNALE DIGITALE

Paolo Foletto

Divisione Julia, 10 - Ponte di Barbarano Vicentino

Si tratta di un programma per l'elaborazione in tempo differito di file di segnali campionati, che possono essere trasferiti via MIDI.

Il software è stato sviluppato a partire da alcune esigenze, da alcune possibilità e da qualche prospettiva.

Per quanto riguarda le esigenze, si vuole offrire all'utente (MIDI e non) la possibilità di intervenire sulla composizione a livello del singolo campione e di eseguire questa elaborazione in maniera semplice.

Per quanto riguarda le possibilità, ci si riferisce ai "piccoli sistemi" apparsi recentemente sul mercato ed in particolare all'ATARI 1040, che è la macchina su cui è stato sviluppato il software.

Questo computer è dotato di un Megabyte di RAM, porte MIDI, mouse ed un sistema operativo con finestre menù ed icone.

Le prospettive si riferiscono ai prodotti annunciati sempre dall'ATARI. Si tratta di modelli con memoria fino a 4 Megabyte, che fanno intravedere la possibilità di una workstation musicale a basso prezzo.

La struttura del programma è composta da un menù principale con 5 scelte ciascuna delle quali, quando viene prescelta, srotola un ulteriore menù.

Il primo menù riguarda l'I/O dei file e quindi prevede opzioni per caricare e scaricare file dal disco o via MIDI dal campionatore.

Il secondo menù prevede delle opzioni per modificare il suono mediante filtri digitali con possibilità di ottenere delay, riverberi, echi ed equalizzazioni.

Il terzo presenta delle opzioni per modificare l'ampiezza e quindi la possibilità di riscalare e di modificare l'involuppo.

Il quarto menù offre la possibilità di lavorare sui campioni sommando (anche più file), sottraendo, miscelando, invertendo nel tempo e nella fase, e un'opzione per la scelta dei punti di loop.

Il quinto menù, infine, serve per vedere la FFT del segnale che può essere fatta o in un singolo istante oppure nel tempo.

CRONOS: NUOVO SISTEMA A MICROPROCESSORE
PER LA CONDUZIONE GESTUALE IN TEMPO REALE
DI ELABORATORI MUSICALI, SEQUENCER E DRUM-MACHINE

Dott. Ing. M. Brescia
Studio: Via Baltimora 22 - Torino

Sommario.

CRONOS è un nuovo sistema di interfaccia uomo-macchina che affronta il problema di sincronizzare dispositivi musicali elettronici (drum-machine, sequencer o anche sofisticati elaboratori per composizione-esecuzione real-time) con i gesti del direttore d'orchestra. Si tratta di un microelaboratore dedicato il cui input è dato da dispositivi di rilevazione gestuale a contatto o a prossimità, mentre l'output consiste nella generazione di segnali di sincronismo per sistemi elettronici pilotati.

CRONOS automaticamente si mette al passo con il direttore, il quale può avviarlo e lasciarlo andare per conto proprio oppure può condurlo gestualmente in maniera continua. È inoltre possibile intervenire sporadicamente per ottenere variazioni di ritmo lievi o anche drastiche. Le risposte del sistema sono comunque esenti da salti bruschi grazie ad un sofisticato sistema di filtraggio digitale.

1. Il problema della conduzione gestuale nella performance musicale elettronica.

Chi assiste a concerti per nastro magnetico e strumenti tradizionali può rilevare una caratteristica generalmente ritenuta negativa: i solisti, come pure l'eventuale direttore, sono costretti a dipendere dal registratore magnetico per ciò che riguarda l'andamento temporale degli eventi. Non sono consentiti quel respiro e quella elasticità di ritmo che caratterizzano profondamente la musica quando è gestita da "umani". Anche lo spettatore avverte disagio per il fatto che il nastro scorra senza un minimo di influenzabilità. Questo è uno dei principali addebiti che vengono rivolti alla tape-music fin dalle sue origini.

Da alcuni anni la ricerca musicale elettronica, anche nell'intento di rimediare a questa situazione, ha in corso di sviluppo sistemi in tempo reale, sui quali cioè l'uomo può intervenire nel corso della performance per modificare gli eventi in atto in alcuni dei loro parametri, uno dei quali è l'andamento temporale.

A differenza da quanto avviene in un'opera d'arte figurativa, un brano musicale presenta necessariamente un ordinamento temporale; in altri termini, è possibile pensare un'asse dei tempi lungo il quale sono assegnati gli eventi musicali. Variare la velocità di esecuzione in performance è cosa praticamente impossibile sui comuni registratori, ma è molto semplice se si ha a che fare con musica preregistrata in codici MIDI sugli odierni sequencer digitali: è infatti sufficiente agire sulla regolazione del clock di sincronismo di base. Anche le drum-machine sono fornite di questa possibilità.

In questo modo non si risolve però un'importante problema che nasce quando si vuole affiancare un solista al sequencer musicale. In tal caso infatti occorre non solo controllare una velocità, ma anche METTERE AL PASSO gli eventi prodotti dalla macchina con i riferimenti temporali rappresentati dai gesti di un direttore. È evidente che il solo ottenere la giusta

velocità non significa affatto avere anche la messa al passo; è sufficiente qualche prova sperimentale per rendersene conto e la cosa è particolarmente evidente quando i gesti richiedono deroghe da un ritmo costante e meccanico.

Questo problema riguarda chi opera nel campo della ricerca musicale d'avanguardia, ma è sentito anche nella live-performance e nella produzione a tutti i livelli.

2. Finalità del sistema cronos.

Una soluzione al problema su esposto potrebbe essere rappresentata dal sistema CRONOS: infatti esso svolge la funzione di INTERFACCIA INTELLIGENTE FRA I GESTI DIRETTORIALI dai quali si fa condurre al passo, E L'ELABORATORE MUSICALE, IL SEQUENCER o LA BATTERIA ELETTRONICA che vengono pilotati con segnali elettronici di sincronismo prodotti da CRONOS in modo flessibile e concettualmente rigoroso. Mediante CRONOS riacquista significato il poter dirigere gestualmente un ensemble di esecutori su strumenti tradizionali che suonano affiancati da apparati musicali elettronici. Si può dunque rimettere nel giusto verso il rapporto uomo-macchina che la tape-music per la sua natura aveva radicalmente distorto. Non più l'uomo, direttore o esecutore, al seguito della macchina ma la macchina al seguito del gesto umano.

La cosa riguarda a pieno titolo anche il pubblico che, mentre con la tape-music ascolta uno svolgersi di accadimenti predisposti che a nessuno è dato influenzare e vede solo due bobine ruotare, ora, con CRONOS, vede i gesti del direttore e ascolta in stretta correlazione le risposte degli apparati elettronici divenuti sorprendentemente reattivi. Si assisterà quindi all'esecuzione di accelerandi o rallentandi, puntualmente ma elasticamente dipendenti dai movimenti del direttore. CRONOS è anche capace di procedere a ritmo costante se i gesti cessano ed è sempre pronto, come un diligente orchestrale, ad accogliere correzioni di tempo anche sporadiche. In concerti per elaboratore solo, senza esecutori umani, il sistema CRONOS permette alla gestualità del direttore di restituire spettacolarità alla situazione; questo fatto potrebbe venire accolto con gradimento dal pubblico che avrà qualcosa da vedere e dal direttore stesso che rinasce a nuova vita ridivenendo interprete dello svolgersi temporale degli eventi.

Ne trarrebbe vantaggio anche il feed-back globale che lega con varie linee di influenza il direttore al pubblico tramite la performance, e viceversa il pubblico al direttore tramite il proprio "feeling".

3. Prestazioni di CRONOS.

Il sistema CRONOS consta di un microelaboratore dedicato, di un dispositivo di input gestuale e di opportune uscite per i segnali di sincronismo, destinati agli apparati da pilotare (sequencer, batteria elettronica, elaboratore). Ecco un elenco delle principali prestazioni di CRONOS, che costituisce anche una sintetica ed essenziale istruzione d'uso.

a) Avviamento gestuale di un ritmo regolare.

Il direttore (in generale la persona che può agire sull'input di CRONOS) batte un tempo con opportuni gesti: CRONOS si avvia già dal secondo gesto, con il ritmo rilevato, e procede in regime di continuo adattamento alle azioni del direttore.

b) Prosecuzione automatica in assenza di gesti.

Se il direttore smette di fornire segnali gestuali, CRONOS continua a comandare i dispositivi pilotati con il preciso ritmo acquisito dagli ultimi segnali gestuali captati.

c) Ripresa del comando gestuale.

Mentre CRONOS è in prosecuzione automatica, se il direttore decide di reintervenire per imporre un nuovo tempo, è sufficiente un minimo di due gesti per ridefinire una nuova velocità metronomica, quella a cui CRONOS si accosterà con grande prontezza ma senza salti bruschi. Si deve però tener presente che un allungamento del tempo superiore al 75% può richiedere azioni più lunghe.

d) Conduzione ininterrotta con accelerandi e rallentandi.

Il direttore può proseguire ininterrottamente la conduzione gestuale con la possibilità di indurre in CRONOS piccole o grosse variazioni di ritmo semplicemente modificando la frequenza dei gesti. Ciò può avvenire dolcemente o bruscamente: CRONOS è in grado di far fronte con elasticità e piccolissima inerzia a richieste di variazioni anche gravose.

e) Messa al passo.

Quando il direttore esegue gesti ritmici e regolari, CRONOS si mette al passo, ossia si sincronizza non solo in velocità ma anche esattamente in fase; ciò significa che i suoi istanti di battito sono esattamente sincroni con i gesti. È questa una caratteristica di evidente importanza per la performance musicale, specie quando sia necessario affiancare gli apparati elettronici ad un ritmo rigidamente imposto dall'esterno.

f) Corona.

Una fermata per una durata ad libitum può essere ottenuta con un comando particolare impartito a CRONOS. Il ritmo riparte su ulteriore comando del direttore dal punto preciso in cui si era fermato.

4. Descrizione funzionale di CRONOS.

All'accensione il sistema entra in stato di attesa di un primo e di un secondo gesto del direttore: si tratta del minimo indispensabile affinché CRONOS possa partire. Infatti nel momento in cui anche il secondo gesto ha luogo, viene iniziato il pilotaggio degli apparati collegati con ritmo identico a quello appena rilevato dai due gesti suddetti (Particolare tecnico: in realtà la frequenza del segnale di sincronismo è multipla di quella espressa gestualmente. Per esempio le batterie a standard MIDI richiedono una frequenza di sincronismo multipla secondo 24 rispetto al quarto di battuta).

Ciò potrebbe bastare a far procedere da sé le cose a tempo indefinito. Ma in genere si vorrà continuare a "gestire" (in doppio senso) il ritmo. Perciò in generale possiamo figurarci un direttore che continua ad eseguire gesti tendenzialmente regolari o, volendo, con ritmo discretamente variabile, per dare respiro alla musica.

Di fronte a tali possibilità di conduzione CRONOS segue due diversi "modi" di funzionamento, appositamente attuati nel complesso software di gestione:

a) l'inseguimento di frequenza (FF, Frequency Follow) e, subordinatamente;

b) la ricerca dell'aggancio in fase (PLS, Phase Lock Search).

Nel modo FF la frequenza del gesto (o più precisamente il periodo di ripetizione) viene correntemente misurata allo scopo di calcolare le correzioni da attuare ininterrottamente sulla frequenza dei segnali in uscita per inseguire quella di ingresso, supposta senz'altro incoostante. Quando l'accostamento delle due frequenze è soddisfacente, il sistema si commuta automaticamente al modo PLS, cioè di ricerca dell'aggancio in fase.

L'obbiettivo è ora far sì che i segnali in uscita raggiungano la coincidenza con i gesti in ingresso: si tratta quindi di misurare lo sfasamento e di generare opportune correzioni sulla frequenza in uscita.

Se il gesto è regolare, CRONOS raggiunge la perfetta sincronizzazione con un breve transitorio di assestamento. Se il gesto è volutamente animato da sequenze di accelerando o rallentando, CRONOS assorbe con grande prontezza gli scarti di fase, trovandosi in definitiva sempre elasticamente legato al direttore umano.

Se poi i gesti vengono sospesi, CRONOS prosegue da solo; ciò è molto comodo perché libera il direttore dall'obbligo di badare continuamente alla conduzione: che bellezza se anche il guidatore d'automobile potesse permetterselo!

5. Possibili dispositivi di input.

Il più semplice dispositivo di input è un microinterruttore (pulsante, pedale) da azionare ovviamente per contatto della mano o del piede.

In alternativa è possibile usare un sensore di prossimità, dunque azionabile senza contatto, basato su un qualsiasi principio di funzionamento (modificazioni di un campo luminoso, infrarosso, ultrasonoro od a microonde, modulazione di parametri distribuiti di tipo L, C od M di circuiti risonanti).

Il microinterruttore, semplice ed economico, dà luogo ad un evento nettamente definito, mentre sistemi "non contact" permettono maggior libertà per l'utente e maggiore spettacolarità del gesto; per contro, sono più complessi e più soggetti ad interferenze esterne.

I dispositivi sono comunque intercambiabili e usabili in combinazione; di volta in volta si può scegliere il più adatto. È allo studio un piccolo radar ad effetto Doppler in grado di rilevare, come evento significativo l'istante in cui la mano del direttore supera una certa soglia di velocità.

Potrebbe rivelarsi appetibile un sensore tale da presentare un campo d'azione allargato anche... al pubblico, che permetta così a tutti di interagire con il ritmo degli eventi musicali in atto.

6. Destinatari di CRONOS.

CRONOS si rivolge a tutti coloro che desiderano rendere più vivo il controllo delle belle "macchine musicali" che in questi anni stanno nascendo, in particolare a:

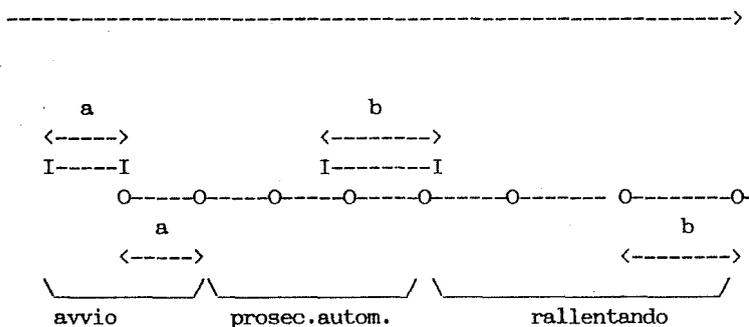
- 1) Computer-Musicisti, ricercatori e compositori in area tecnologicamente innovativa;
- 2) Musicisti di qualsiasi tendenza, che facciano uso di sequencer e batterie elettroniche;
- 3) Amatori delle novità tecnologiche applicate all'interazione uomo-macchina.

7 - ESEMPI DI FUNZIONAMENTO

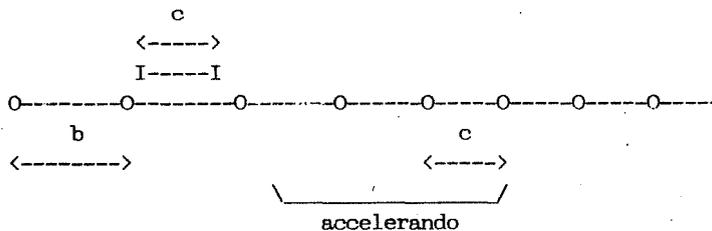
7.1 - SEQUENZA DI AVVIO, PROSECUZIONE AUTOMATICA E RALLENTANDO -

GESTO = I (qui e nel seguito)

RISPOSTA = 0 asse dei tempi



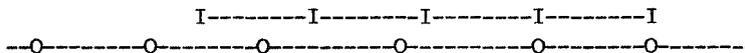
7.2 - SEQUENZA DI ACCELERANDO



7.3 - SEQUENZE DI MESSA AL PASSO

Qui il gesto
richiede ritardo...

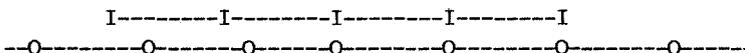
<-->



...e CRONOS rallenta e riaccelera,
ottenendo il perfetto sincronismo

Qui il gesto
richiede anticipo...

<-->



...e CRONOS accelera e rallenta,
sempre allo scopo di sincronizzarsi

PRO.FUMO

D. Tanzi

L.I.M. - Laboratorio di Informatica Musicale - Dipartimento di Scienze dell'Informazione
Università degli Studi di Milano

PRO.FUMO è una composizione basata sulla manipolazione di testi musicali effettuata in tempo reale tramite due elaboratori MSX1 e MSX2.

Per la realizzazione e l'esecuzione del brano sono stati utilizzati programmi sviluppati da Claudio Pagni ed Alberto Stiglitz presso il L.I.M. dell'Università di Milano in un'ambiente hardware basato su computer MSX1 e MSX2, connessi al modulo Yamaha per la sintesi FM SFG-05 e collegati via MIDI a varie periferiche digitali (campionatori e sintetizzatori).

I programmi sono stati scritti utilizzando il linguaggio BASIC (esteso mediante il MUSIC MACRO per il controllo del modulo SFG-05) ed il linguaggio ASSEMBLY per le parti condizionate da problemi di velocità di esecuzione o che interessano lo sfruttamento di caratteristiche hardware non utilizzabili con il BASIC.

Il Sistema per l'Elaborazione Musicale (S.E.M.) risultante consente di trasformare strutture musicali (frammenti melodici) predefinite in opportuni archivi, e di generare in tempo reale dati musicali per l'esecuzione diretta mediante il modulo SFG-05 o periferiche MIDI ad esso collegate. Le trasformazioni avvengono a livello strutturale ed utilizzano operatori che seguono i principi delle trasformazioni omologiche (trasformazioni geometriche che descrivono, mediante un'opportuna rappresentazione, procedimenti tipici utilizzati in vari ambiti della composizione moderna e contemporanea).

Inoltre è possibile configurare il S.E.M. con una certa flessibilità per adattarlo alle esigenze del singolo compositore e nella versione per computer MSX2 comprende anche un modulo per l'improvvisazione in tempo reale utilizzando il mouse come strumento di controllo per la performance.

La definizione dei frammenti melodici in PRO.FUMO è avvenuta sulla base del riconoscimento di alcuni elementi significativi presentati dallo sviluppo di una struttura base o modulo, definita tramite notazione su una polifonia a 4 parti.

L'organizzazione del materiale tematico e ritmico è libera.

L'orchestrazione timbrica si è avvalsa sia di timbri sintetici che di timbri campionati, in parte presenti nella libreria del L.I.M., in parte originali.

L'orchestrazione timbrica può essere riconfigurata in tempo reale in base alle esigenze poste dalla partitura.

SESSIONE POSTERS

LA TARTAMUSA OVVERO L'OGGETTIVAZIONE DELL'ASTRATTO MUSICALE

L. Tarabella, A. Venturi, G. Bertini
CNUCE ed IEI - Istituti del CNR

Il nome Tartamusa è la parodia del nome Tartaruga del linguaggio LOGO: se la Tartaruga si muove in uno *spazio bi-dimensionale statico* attraverso un insieme di primitive come AVANTI, INDIETRO, DESTRA, SINISTRA, PENNA-SU e PENNA-GIU' concretizzando nel movimento *figure geometriche*, la Tartamusa si muove in uno *spazio mono-dimensionale dinamico* con l'uso di un insieme di primitive del tipo SALI, SCENDI, SUONA e TACI, concretizzando nel movimento *melodie*; l'effettivo movimento della Tartamusa avviene perciò sul continuum frequenziale (dal grave all'acuto) che a sua volta si muove, in maniera globale ed uniforme ma indipendentemente dal movimento della Tartamusa stessa, nel tempo. Il tempo è infatti l'elemento vitale per l'esistenza della Tartamusa.

Le primitive Sali e Scendi, in realtà poi, si riducono alla sola *MUOVI n* che ammette come valore parametrico un numero intero il cui segno stabilisce la direzione del movimento; e le Suona e Taci si riducono alla sola *EMETTI*. Si può immaginare una situazione reale come quella in cui un dito della mano sinistra scorre sulla tastiera della chitarra nelle due direzioni per stabilire l'altezza di un suono, è l'azione della mano destra che emette effettivamente quel suono. E se lo spostamento della mano sinistra avviene su distanze intervallari predeterminate nella tastatura, per la Tartamusa è necessario (e possibile) stabilire il salto quantico attraverso la primitiva *DIVIDI - OTTAVA n*: con $n = 12$ si ha per esempio la divisione temperata equabile. E poiché la Tartamusa si muove alla successiva posizione in maniera relativa all'attuale, è necessario poter definire il punto di partenza, e cioè qualcosa di simile all'*HOMME* dei sistemi grafici; per far ciò esiste la primitiva *DIAPASON r*: con $r = 440.0$ viene stabilito il diapason classico. La Tartamusa è quindi quell'oggetto che concretizzando l'astratto musicale, nei suoi spostamenti sul dominio frequenziale, lascia dietro di sé *scie melodiche*.

Ciò detto, è possibile sostituire alla lapidaria asserzione: *la musica è l'arte dei suoni*, un'asserzione apparentemente riduttiva, ma alla fin fine più aperta e promettente quale: *la musica è un processo*. Anzi: la musica è un insieme di sottoprocessi paralleli, dove per processo si intende l'evolversi nel tempo di un insieme di enti in relazione tra loro. In prima approssimazione si possono individuare le seguenti classi di sottoprocessi: *processi frequenziali*, *processi timbrici*, *processi volumetrici*, *processi ritmici*; ed inoltre *processi logici*, dove per processi logici si intende l'evolversi ed il combinarsi di quelle unità che prendono il nome di motivi, incisi, melodie, ritornelli etc.. E se per le prime tre classi di processi non occorre dare specifici ragguagli dato che esse si riferiscono univocamente all'essenza stessa della musica in quanto modulazione frequenziale, di colore timbrico e di intensità sonora, qualche riga deve invece essere spesa a favore della quarta, relativa ai processi ritmici.

Di solito si pensa alla *nota musicale* come ad un'unità atomica contenente due informazioni: la prima relativa alla frequenza e la seconda alla durata; e quando si pensa ad una frase musicale la si concepisce come un tutt'uno di modulazione frequenziale e ritmica. È tuttavia facile ed intuitivo rilevare in una frase musicale quella che è la melodia pura intesa come successione di relazioni intervallari tra un suono e il successivo, e quello che è il puro andamento ritmico visto come successione di durate relative in relazione tra loro. Una volta che si sia eseguita tale operazione, i due fenomeni appaiono essere il risultato di due processi del tutto

distinti. Del resto nello studio della musica esistono due momenti disgiunti nei quali si studiano, nel primo, le interrelazioni tra i suoni nelle loro identità intervallari e completamente atemporali (armonia) e nel secondo, le figure ritmiche del tutto avulse da un contesto armonico (solfeggio).

Se quindi la MUOVI è preposta al movimento della Tartamusa sullo spazio frequenziale, è la EMETTI che ha il compito di generare effettivamente l'evento, mettendo l'opportuno valore in coda alla lista delle altezze (processo frequenziale); alla EMETTI fa poi riscontro la primitiva DURATA-FREQ r atta alla generazione della lista dei valori temporali (processo ritmico); le due liste infine vengono associate in modo ordinale al momento dell'effettiva esecuzione musicale. E poiché la DURATA-FREQ vuole in r un valore che esprime una quantità vicina al concetto di figura musicale (e cioè un valore relativo ad una quantità prestabilita) esiste la primitiva METRONOMO m che stabilisce l'unità di misura cui fare riferimento.

In un primo momento, l'aver posto il suffisso -FREQ per la primitiva che genera i valori temporali da associare alle frequenze, può sembrare una ridondanza; ma se si pensa che poco sopra abbiamo definito una classe di sottoprocessi dove sia volumi che timbri hanno vita propria, si vede la necessità di avere primitive che generino durate riferibili a questi altri parametri. La Tartamusa per variare il suo volume sonoro utilizza la primitiva VOLUME m che imposta il nuovo valore in relazione al precedente in misura del valore algebrico di m ; esiste così anche la primitiva ZERO-VOLUME e la primitiva DURATA-VOL r che genera le durate associate ai volumi.

Un discorso più delicato è quello invece relativo ai processi timbrici: il timbro, come infatti sappiamo, a differenza degli altri parametri fisici come la frequenza e l'intensità che sono esprimibili in quantità mono-dimensionali, ha bisogno di essere definito in termini di algoritmo di sintesi e di un insieme di parametri associati. In prima istanza diciamo perciò che esiste la primitiva TIMBRO *nome*, che vuole in *nome* un record contenente tutte le informazioni del caso, e la primitiva DURATA-TIM r che mantiene attiva una figurazione per una durata di tempo r .

Una primitiva di estrema importanza è infine la VOCE n che indirizza tutte le altre primitive a generare eventi per la voce indicata in n .

Ma è dunque il LOGO l'ambiente dentro cui la Tartamusa si muove? Sì, ma non necessariamente! Cioè, dato che in un qualunque linguaggio è possibile scrivere routines o primitive che assolvano i dovuti compiti e richiamabili simbolicamente con un nome che ne esprima l'azione, l'ambiente può essere un qualunque linguaggio di programmazione: e se l'aver associato la grafia musicale ad un linguaggio di programmazione fu in passato motivazione e considerazione iniziale per il disegno e la realizzazione di PRIMULA visto appunto come Linguaggio di PRogrammazione MUSicale, il trasporre gli stessi concetti sui linguaggi di programmazione noti, fa ora di PRIMULA una metodologia descrittiva ed implementativa della musica; e della TARTAMUSA la sua oggettivazione.

Controllare la Tartamusa con tutta la ricchezza di strutture di controllo di cui è dotato un linguaggio di programmazione ad alto livello, significa avere in modo naturale ed a costo zero un sequencer di per sé molto sofisticato e perciò in grado di soddisfare le più diverse esigenze architettonico-musicali.

I vantaggi che emergono da questa impostazione sono numerosi e veramente appetibili: intorno a ogni linguaggio esiste cultura, documentazione e perché no, simpatia; ogni calcolatore ha i propri ambienti di sviluppo programmi, i propri file-system e le proprie modalità di lavoro: il lavoro di implementazione della metodologia PRIMULA è così ridotto all'essen-

ziale, e cioè alla scrittura del nucleo di primitive, alla definizione delle strutture dati, ed al generatore di codice musicale tipo-MIDI o semplicemente MIDI. Ma tutto questo, fatto per un particolare linguaggio e per un particolare calcolatore, è presto fatto anche per altri linguaggi e per altri calcolatori: cambia solo la sintassi, le modalità operative al contorno, ma la metodologia implementativa è esattamente la stessa. Certo, cambiando linguaggio si possono avere possibilità inesistenti in altri; cambiando calcolatore si possono avere più ampi spazi di memoria, velocità di calcolo più elevate e ambienti di lavoro più confortevoli. Ma ciò che rimane costante è l'approccio che vede un linguaggio, anzi, la *programmazione* in sé, come strumento descrittivo ed implementativo della musica.

Impostata la metodologia, l'unico problema è dunque *da dove iniziare*, dato che lo spazio d'azione si presenta essere tri-dimensionale: Computer x Linguaggio x Apparecchiatura-di-sintesi. La scelta della prima realizzazione della metodologia è andata verso tre standards: Macintosh, Pascal, TX81Z. Giustificazioni? Disponibilità, prestazioni, costi, simpatia. Ma proprio perché questo è solo un inizio, tutto quanto detto e tutto quello che si dirà è valido per qualunque altra configurazione.

È bene comunque chiarire subito un eventuale dubbio: il Pascal (come la maggior parte dei linguaggi di programmazione noti e largamente utilizzati) non consente la *concorrenza*, non dà cioè la possibilità di descrivere fenomeni che abbiano ad accadere contemporaneamente ad altri; e questo invece sembrerebbe essere necessario nel nostro caso, visto che abbiamo preferito la definizione che vede la musica come un fenomeno ad alto parallelismo. Fortunatamente il parallelismo che ci interessa non implica, in generale, interrelazioni tra processi, ma solo sincronismo: perciò uno stesso programma può essere pensato sezionato in più parti (come del resto accade per una composizione musicale) ciascuna delle quali a run-time genera una struttura dati contenente la successione degli eventi relativi ad ogni particolare parametro di una particolare voce. In questa visione la composizione può essere vista in maniera *incrementale*, e cioè: in un primo tempo ci si può occupare dell'impianto globale della composizione, e dell'aspetto melodico e ritmico; in un secondo momento si può aggiungere un complesso di routines che si occupi dell'aspetto timbrico, e in un terzo momento (se l'apparecchiatura di sintesi lo permette) di routines che gestiscano il movimento dei suoni in uno spazio simulato.

È importante perciò capire che esistono sostanzialmente tre momenti che si iterano tra loro:

- 1) scrittura del brano in termini di routines che facciano uso delle primitive del kernel;
- 2) attivazione del programma e conseguente generazione delle tabelle degli eventi (processo trasparente all'utente);
- 3) generazione del codice MIDI (o tipo-MIDI) a partire dalle tabelle generate. Si noti che, essendo l'oggetto del lavoro un unico programma, la terza fase è divisa dalla seconda solo logicamente, e non fisicamente: si vuole con ciò far trasparire che delle routines del kernel fa parte anche la routine *Execute* che, una volta attivata, genera il codice seriale di eventi sincronizzati, a partire dalle tabelle generate. La struttura tipica di un programma che definisca un brano musicale polifonico in metodologia PRIMULA, realizzata in PASCAL è dunque:

```
Program BranoMusicale;
```

```
  include Kernel;
```

```
  ..... routines che definiscono il brano dell'utente .....
```

Procedure Start;

Parte 1; (* viene invocata la routine che definisce i parametri della voce 1 *)

Parte 2; (* 2 *)

Parte n; (* n *)

end;

begin

Start; (* attiva il processo di generazione delle tabelle Eventi/Durate come da programma *)

Execute (* attiva la generazione del codice MIDI a partire dalle tabelle Eventi/Durate *)

end.

La *Start* è una routine definita dall'utente e serve ad attivare il programma/brano. È bene puntualizzare che la forma programmatica esposta è solo una delle tante possibili; ed è altrettanto importante ribadire che per la generazione degli eventi, il programma deve fare uso delle routines del kernel: routines, che possono quindi essere considerate come l'interfaccia tra l'utente e la *metodologia*.

Le primitive del Kernel sopra esposte, costituiscono solo il minimo indispensabile per controllare il movimento relativo e multi-dimensionale della Tartamusa. Ora, pur insistendo sull'aspetto relativo del movimento della Tartamusa, il kernel di primitive può essere ampliato e/o variato a piacimento dall'utente. Per esempio, potrebbero far parte del kernel alcune primitive che gestiscano il movimento sui gradi di una scala predefinita: SCALA VET, dove vet è un vettore contenente valori indicanti distanze intervallari; MUOVI-SCALA n, che sposta la frequenza sui gradi della scala in quel momento attiva, a seconda del valore (con segno) di n.

Con la metodologia PRIMULA ed il movimento della Tartamusa, si vuole dunque proporre un approccio implementativo che, date le caratteristiche di versatilità e di generalità, potrebbe costituire un primo momento di discussione per la definizione di uno standard di lavoro da utilizzare nel settore dell'Informatica Musicale; si tratterebbe in ultima analisi di definire un insieme di primitive abbastanza generalizzato al quale attenersi per parlare una lingua comune in vista di dare soluzione a quell'annoso problema conosciuto come *standardizzazione del software*. Nulla vieterebbe poi di creare dialetti privilegiati o addirittura minoranze linguistiche. Quello che si vuole mettere in evidenza è che questa metodologia possiede la caratteristica fondamentale di essere sempre uguale a sé stessa a qualunque livello di sofisticazione, inserita in qualunque linguaggio e/o ambiente, su qualunque calcolatore ospite e con qualunque apparecchiatura di sintesi sonora la si usi.

Durante il Colloquio verranno presentate due implementazioni campione:

Macintosh + LightSpeedPascal + TX81Z Yamaha

Commodore64 + EnhancedBasic + SID (l'EnhBasic consente la def. e la chiamata di subs simboliche).

Bibliografia.

PORENA B. — *N. 2 per la Composizione*. Ed. Ricordi, 1983.

- PAPERT S. — *Mindstorms*. Basic Book Inc., New York, 1980 (trd. Italiana - Emme Edizioni, Milano 1984).
- SCHOENBERG A. — *Fundamentals of Musical Composition*. Faber & Faber Ltd, London, 1967, (trd. italiana, Suvini/Zerboni, Milano, 1969).
- TARABELLA L., VENTURI A. — *La Tartamusa*. Nota CNUCE/CNR C-87 n. 5.
- BERTINI G., TARABELLA L. — *Il modulo i sintesi sonora con TMS32010*. Nota CNUCE/CNR C-87 n. 16.
- TARABELLA L. — *The Primula Machine*. CMJ Vol. 11 n. 2.
- INSIDE MACINTOSCH — *Addison Wesley Publ. Company, Inc., 1987*.
- LIGHTSPEEDPASCAL — *Think Technology Inc., 1986*.
- TX81Z USER'S MANUAL — *Yamaha*.

WAVETABLE COME AUTOMA CELLULARE: UNA NUOVA TECNICA DI SINTESI

J. Chareyron
Via Moisé Loria, 50 - Milano

Il presente intervento descrive una nuova tecnica di sintesi, nella quale un automa cellulare viene usato come base dell'algoritmo di generazione del suono.

L'algoritmo presentato offre due grossi vantaggi: è semplice da mettere in atto senza la necessità di hardware specializzato, ed i suoni generati presentano una grande varietà timbristica.

Il lato negativo è legato alla vastità dell'insieme degli automi utilizzabili. È difficile identificarne gli elementi più interessanti e inoltre gli strumenti classici del DSP sono in gran parte inutilizzabili per analizzare l'effetto sonoro prodotto dall'automa in esame. Sarà necessario creare nuovi strumenti per agevolare l'uso pratico dell'algoritmo presentato.

Un precedente nell'uso di automi cellulari in un processo musicale è identificabile nel contributo di Joel Ryan e Ray Edgar alla mostra allestita nel quadro dell'ICMC 1986. Le configurazioni sequenziali di un automa lineare a due stati erano tradotte numericamente dal calcolatore in codici Midi di altezza di note e trasmessi a un sintetizzatore esterno.

Nella tecnica di sintesi presentata in questo intervento l'automa cellulare viene invece usato direttamente a livello di segnale. Le celle dell'automa sono i punti successivi di una wavetable, il nuovo stato di una cella diventa il valore del campione da mandare a un DAC.

Un automa cellulare è un'insieme di caselle dotate ognuna di uno stato (simbolizzato da un numero intero) che si modifica nel tempo. Ad ogni battito di un immaginario orologio, lo stato di ogni cella viene ricalcolato a partire dagli stati dell' "intorno" della cella (composto dalla stessa cella e dalle celle che gli stanno vicine) secondo un algoritmo di calcolo chiamato regola di transizione. Per conoscere l'evoluzione di un automa cellulare sono perciò necessari due elementi: la situazione iniziale e la regola di transizione (una funzione a valori interi definita sull'insieme degli "intorni" possibili di una cella). Un noto esempio di automa cellulare bidimensionale è il famoso gioco della vita ("Life") di John Conway in cui ogni cella può avere due soli stati, vita o morte.

Un automa lineare è un automa unidimensionale. Le celle sono disposte lungo una riga (ogni cella dispone dunque di due vicini) e gli n stati possibili vengono rappresentati con interi compresi tra 0 e $n-1$. L'algoritmo più comune usato come regola di transizione consiste nel sommare lo stato della cella con quello delle due celle vicine e attribuire il nuovo stato secondo una tabella. Una regola di transizione si trova così determinata con la conoscenza di una funzione a valori interi (da 0 a $n-1$) definita sull'insieme dei numeri interi da 0 a $3(n-1)$, o ancora di un vettore a $3n-2$ componenti intere comprese tra 0 e $n-1$.

Nell'algoritmo di sintesi presentato, i p elementi di una wavetable sono le p caselle di un automa lineare. Il numero intero che caratterizza lo stato di ogni cella è il valore dell'elemento corrispondente della wavetable. La wavetable viene letta ciclicamente e i dati vengono poi trasmessi a un DAC. Dopo ogni lettura i valori della wavetable vengono aggiornati a seconda della regola di transizione vigente. Il calcolo di ogni nuovo campione esige tre operazioni (due addizioni e la lettura di una tabella di riferimento) che possono essere eseguite in tempo reale da un comune calcolatore senza necessità di hardware specifico.

Il meccanismo appena esposto (ricalcolo in tempo reale degli elementi di una wavetable) ricorda quello usato dall' algoritmo di Karplus-Strong. Vedremo infatti più avanti che è possibile simulare quest'ultimo con un automa particolare.

Ogni automa (e di conseguenza ogni suono generato) dipende da due tabelle: la wavetable di partenza (p elementi) e la tabella che rappresenta la regola di transizione ($3 \cdot 2^{ab} - 2$ elementi, dove b è la risoluzione del DAC, espressa in bits). La dimensione della seconda tabella è piuttosto importante, circa 192K words per un campionamento a 16 bits (o la metà se si usa un automa simmetrico). Non è un vero problema data la disponibilità di RAM e di memorie di masse sempre più confortevoli. Se la memoria a disposizione è limitata, è possibile usare una tabella molto più ridotta al prezzo di un'attività computazionale appena più intensiva.

La dimensione delle tabelle usate lascia prevedere l'enormità del numero di automi (dunque di suoni) disponibili. Per un campionamento a 16 bit il numero delle sole regole di transizione supera il valore 10 alla potenza 250.000! Anche se una grande maggioranza di questi automi risultano inutilizzabili musicalmente (il suono evolve rapidamente in rumore), rimane un ingente numero di automi interessanti e la varietà degli effetti musicali è considerevole.

Questa ricchezza è anche un punto di debolezza dell'algoritmo presentato: la ricerca degli automi utilizzabili e la loro classificazione diventa un'impresa piuttosto ardua, che necessita della creazione di strumenti adatti.

Per uno studio teorico dell'algoritmo di sintesi presentato si può considerare il processo di elaborazione come l'azione di un filtro digitale su un segnale limitato alla wavetable originale.

L'equazione differenza si può allora scrivere

$$y(n) = x(n) + f(y(n-p-x) + y(n-p) + y(n-p+1)) \quad (1)$$

dove * p è la lunghezza della wavetable

* $x(n) = 0$ per $n < 0$ e $n > p-1$. Per $0 \leq n < p$, i valori di $x(n)$ sono quelli presenti nella wavetable di partenza.

* f è la funzione che definisce la regola di transizione.

Per poter proseguire nell'analisi con gli strumenti tradizionali del DSP, è necessario che la funzione f sia lineare. Purtroppo non è il caso della stragrande maggioranza delle funzioni che definiscono le regole di transizione.

È comunque utile soffermarci lo stesso su questo caso particolare (f lineare), perché l'analisi conduce a conclusioni semplici e i risultati musicali sono particolarmente interessanti.

Se f è lineare, l'equazione (1) diventa

$$y(n) = x(n) + a[y(n-p-1) + y(n-p) + y(n-p+1)]$$

dove a è un numero di modulo inferiore a 1/3 (condizione necessaria per impedire che il risultato della funzione esca dal limite di una parola da b bit).

Il termine $x(n)$ diventa nullo dopo la prima lettura della wavetable e possiamo considerare l'intero processo come la combinazione di un delay di lunghezza p-1 e di un calcolo di media effettuato sui tre ultimi campioni (con un guadagno a).

La somiglianza con la versione di base dell'algoritmo di Karplus-Strong diventa evidente. Quest'ultimo combina anche lui un delay e un calcolo di media, con due differenze: il delay ha per lunghezza p, e la media viene calcolata sui due ultimi campioni. Il comportamento dei due filtri digitali associati ai due algoritmi è molto simile, mostrando un guadagno massimo per la frequenza della fondamentale e un guadagno minore (in linea di massima decrescendo con la frequenza) per le altre componenti. Un automa più vicino ancora all'algo-

ritmo di Karplus-Strong può essere creato usando un altro tipo di regola di transizione (la somma delle tre celle viene ponderata con un coefficiente 2 per la cella centrale).

L'ascolto conferma questi risultati: con una wavetable iniziale casuale, si ottengono timbri gradevoli, buone simulazioni di suoni di corde pizzicate.

Per tutti gli altri automi, i metodi classici del DSP non sono di grande utilità e bisogna creare nuovi strumenti per agevolare la "caccia" agli automi.

Fissiamo qualche punto di partenza. Innanzitutto è comodo isolare una classe particolare di regole di transizione, che possiamo chiamare simmetriche, per le quali la funzione f dell'equazione (1) è dispari. In altre parole due forme d'onde simmetriche rispetto all'asse delle ascisse daranno luce a due nuove forme d'onde anch'esse simmetriche. Questo è un comportamento molto frequente nei suoni naturali ed elettronici ed è logico privilegiare questo tipo di regole. Si ottengono così due vantaggi: l'evoluzione nel tempo è più facile da prevedere, e la tabella della regola di transizione può essere ridotta di metà. Non bisogna comunque escludere gli altri automi tra i quali troviamo anche elementi di grande interesse musicale.

Altre osservazioni aiutano a sfoltire il campo della ricerca. In primo luogo la maggior parte delle applicazioni musicali richiede che l'evoluzione della wavetable sia progressiva e senza sbalzi brutali. Un modo semplice per ottenere questo risultato consiste nello scegliere la funzione f tale che il valore di $f(x)$ sia vicino alla terza parte del valore della variabile (ricordiamo che la variabile rappresenta la somma di tre punti successivi della wavetable).

A questo punto possiamo distinguere i casi nei quali i valori di f sono inferiori alla terza parte della variabile, e quelli nei quali sono superiori. La prima soluzione produce un suono che si estinguerà da solo dopo un numero abbastanza ridotto di generazioni. La seconda condurrà a una situazione analoga nel caso di regole di transizione nonsimmetriche (con stabilizzazione al livello massimo della tensione invece che al livello zero). Con regole simmetriche e wavetable di partenza sufficientemente semplici (una serie di valori positivi seguita da una serie di valori negativi) si ottengono suoni particolarmente interessanti: l'automa si stabilizzerà su una forma d'onda rettangolare. Scegliendo una wavetable di partenza con uno spettro semplice otterremo così un suono di composizione spettrale crescente. Già questi primi esempi permettono di immaginare un ipotetico strumento basato sull'algoritmo di sintesi descritto: al segnale note-on viene fatto scattare un automa corrispondente a un suono di crescente ampiezza/ricchezza armonica; al successivo segnale di note-off si cambia per una regola di transizione che presenta il comportamento opposto. È interessante notare che l'algoritmo provvede non solo al controllo del timbro del suono, ma anche a quello della sua ampiezza.

Naturalmente la funzione f può anche prendere valori alternativamente più grandi e più piccoli della terza parte della variabile. Diversi schemi di evoluzione sono allora possibili: stabilizzazione dei valori della wavetable su una o più costanti, vibrazioni intorno a queste costanti, estinzione progressiva, ecc..

Per lo stadio successivo della ricerca è necessario stabilire un "piano di battaglia" articolato in più tappe: selezione di diverse famiglie di automi con caratteristiche interessanti; per ogni famiglia costruzione dell'insieme delle regole di transizione secondo un unico schema di calcolo parametrato da un numero ridotto di variabili; analisi spettrale dinamica del suono generato per diversi valori dei parametri e della situazione iniziale; elaborazione delle regole che legano le variazioni di un parametro all'evoluzione della composizione spettrale del suono.

Il processo precedente dovrà naturalmente essere ripetuto diverse volte e affinato ad ogni

iterazione. Lo scopo di questo studio è la costruzione di un catalogo di automi parametrabili, disponibili per generare gli effetti desiderati.

Naturalmente, lo studio dell'algoritmo presentato dovrà prendere in considerazione i risultati ottenuti nella scienza degli automi lineari. La ricerca è molto attiva, ma gli studi riguardano prevalentemente automi con un limitato numero di stati e sono di rado applicabili al nostro caso.

Un'importante risultato generale è la classificazione degli automi lineari in quattro grandi famiglie. Due di queste classi corrispondono ad un comportamento semplice: in tempi abbastanza ridotti la configurazione diventa stabile o periodica. Questi automi sono i più adatti alla generazione di suoni musicali e si prestano bene al tipo di analisi descritta nei paragrafi precedenti. Le cose si complicano con la terza famiglia di automi: l'evoluzione è più caotica, i suoni presentano ricche strutture e inattesi cambiamenti di andamento, ma tendono spesso al rumore. Diventa molto difficile prevederne l'evoluzione.

Gli automi che compongono l'ultima famiglia sono detti computazionalmente irriducibili, ossia non esiste un algoritmo (che non sia l'applicazione della regola di transizione) con il quale calcolare lo stato dell'automa per le future generazioni. In parole povere l'evoluzione dell'automa è del tutto imprevedibile, il solo modo per conoscere il suo stato futuro è di lasciarlo "vivere" e aspettare con pazienza. Questo carattere è certamente un handicap nelle applicazioni musicali. Chi lavora nel campo dell'informatica musicale desidera (giustamente) avere pieno controllo sui mezzi che adopera. In compenso, l'imprevedibilità del suono generato da questi automi è proprio quello che dà loro un carattere eccezionale. Il suono non obbedisce ad una legge matematica rigida, ma sembra godere di una propria autonomia. Le attuali tecniche di sintesi sono tutte computazionalmente riducibili, il suono generato è facilmente controllabile ma spesso troppo prevedibile. Un processo computazionalmente irriducibile esce da ogni tipo di standardizzazione e porta un pizzico di sorpresa che può giovare alle opere di computer music. I suoni elaborati in questo modo allargano il campo delle scelte a disposizione di musicisti e ricercatori.

Un grosso vantaggio della tecnica di sintesi presentata è la possibilità di creare con lo stesso algoritmo suoni di natura del tutto diversa, dalle semplici simulazioni dell'algoritmo di Karplus-Strong alle complesse strutture dei suoni prodotti da automi computazionalmente irriducibili.

Questi sono solo i primi elementi nello studio dell'algoritmo presentato. È già stato indicato come si intende proseguire in questo lavoro. La ricerca può essere allargata anche in altre direzioni:

- Uso di automi che obbediscono ad un altro tipo di regola di transizione (il nuovo stato di una cella non sarebbe più determinato dalla somma degli stati delle tre celle vicine). Una modifica consiste nel ponderare diversamente le tre celle (si è visto l'interesse di questa soluzione per la simulazione dell'algoritmo di Karplus-Strong) ma esistono altri algoritmi di calcolo del tutto diversi.
- Utilizzazione dell'algoritmo presentato anche come tecnica di elaborazione del suono. La wavetable viene rifornita periodicamente dei campioni in ingresso e il dispositivo restituisce in uscita lo stato dell'automa dopo una o più generazioni.

Il campo di ricerca, si vede, è molto vasto. I primi risultati ottenuti sono incoraggiati e spingono ad andare avanti in questo studio.

COLLEGAMENTO MIDI PER COMPUTERS DEC PDP11 E VAX

P.I. F. Sani
Via Trieste 8 - Carmignano di Br. - PD

L'affermarsi del protocollo MIDI per l'informatica musicale è stato recepito solo dai costruttori di piccoli computers mentre in quelli di dimensioni "industriali" l'esiguità del numero di applicazioni musicali ha impedito ai produttori di realizzare interfacce adeguate.

Per quanto concerne i sistemi DEC PDP11 e VAX il problema si può risolvere facilmente utilizzando una interfaccia del tipo DL11 opportunamente modificata (sono consigliate le schede vecchie ad una sola linea).

Le modifiche da apportare alle schede sono di due tipi:

- 1) Sostituzione del quarzo oscillatore per portare il BAUD RATE a 31250;
- 2) Installazione dei connettori MIDI IN e MIDI OUT.

Per calcolare la frequenza del nuovo quarzo bisogna usare la seguente

formula: $QN = 31250 * QV / MBR$
dove: $QN =$ quarzo nuovo
 $QV =$ quarzo installato
 $MBR =$ massimo baud-rate della scheda

(in ogni caso deve risultare un multiplo di 500khz).

È necessario verificare che gli integrati divisori di frequenza abbiano una banda passante sufficiente e, nel caso devono essere sostituiti con altri della serie 74H - ad alta velocità.

Infine bisogna sostituire i condensatori del circuito risonante con altri di valore inferiore in proporzione alla variazione di frequenza.

(N.B. Per semplificare si può sostituire il circuito oscillante con un oscillatore al quarzo del tipo integrato).

Per quanto riguarda i connettori MIDI, dovendo questi lavorare con segnali di tipo TTL andranno collegati direttamente sulla logica tramite dei cavi schermati bypassando i driver ed i receiver dei segnali RS232.

I segnali da utilizzare si chiamano TTL SERIAL IN e TTL SERIAL OUT e vanno isolati dai segnali RS232 tagliando le relative piste.

Questo tipo di modifica è stato realizzato in collaborazione con il Centro di Sonologia Computazionale dell'Università di PD che lo utilizza nel sistema 4X-PDP11/34.

SISTEMI A BASSO COSTO PER L'ELABORAZIONE DEL SUONO

Stefano Petrarca - STREAM Roma

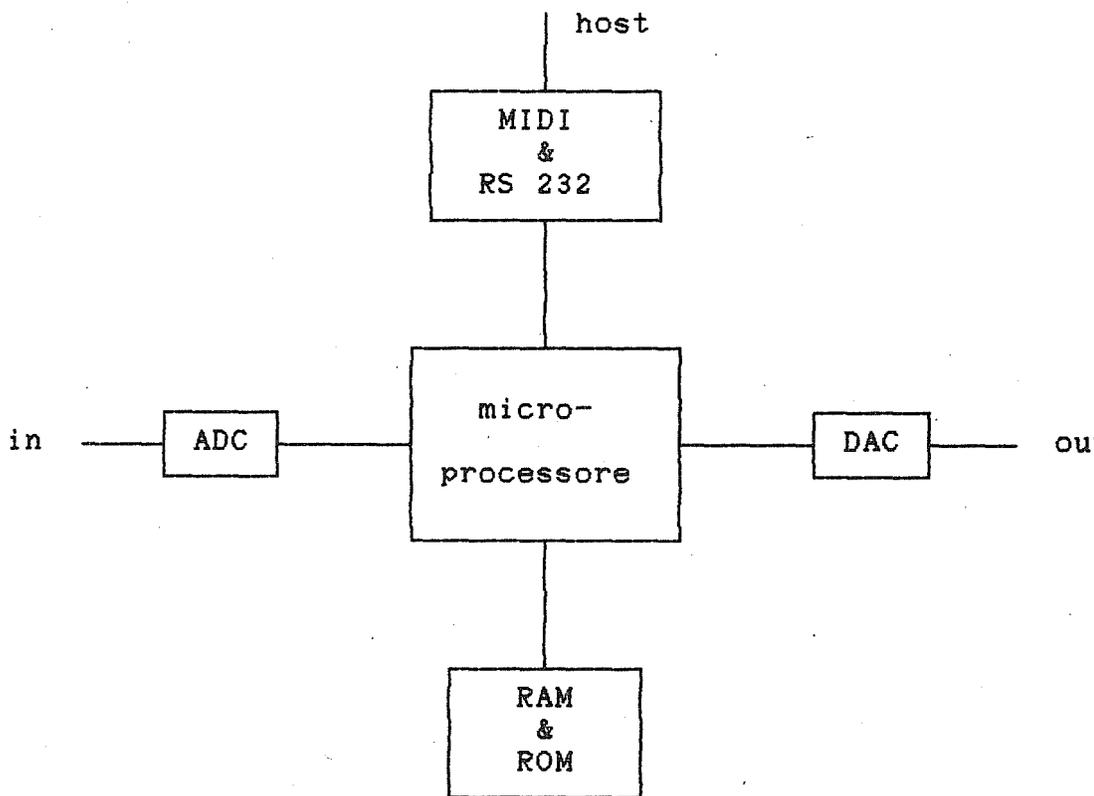
Cercando di fare un bilancio della frenetica proliferazione di aggeggi elettronici che possono essere usati (o specificamente progettati) per la sintesi e l'elaborazione del suono talvolta si sente il bisogno di un buon taglio stile Occam. Ci si potrebbe domandare se c'è sempre bisogno di tutta la potenza messa a disposizione dalle più moderne macchine elettroniche per fare della buona musica; la risposta è incerta e la tecnologia elettronica è affascinante e trascinate e, senza adeguato controllo, può diventare, per il musicista di buona volontà, una trappola che inevitabilmente conduce verso quella confusione tra mezzi e fini così caratteristica nella nostra epoca fatta di consumi forsennati, di cura per le cose inessenziali e di perdita graduale di una dimensione progettuale nell'ansia di vivere alla giornata senza fermarsi mai a riflettere sulle sciocchezze — o le prodezze — commesse e su quelle che si programmano per il futuro. Dopo l'inevitabile tirata moralistica, anch'essa qualche volta ancora di moda, penso possa essere interessante parlare della filosofia che sta alla base dei sistemi SAU per l'elaborazione del segnale acustico. Prima di tutto va detto che i sistemi di cui sto parlando sono dedicati esclusivamente alla manipolazione di suoni provenienti da sorgenti concrete e, dunque, all'uso in concerti dal vivo. Lo STREAM da sempre è particolarmente attivo in questo settore ritenuto particolarmente fecondo per le interessanti prospettive che offre da un punto di vista strettamente musicale e mai i componenti dello STREAM hanno messo in secondo piano la ricerca compositiva per lanciarsi in effimere avventure tecnologiche. Un punto di forza dei SAU è il loro basso costo; le loro prestazioni, è vero, non sono elevatissime dato che le tecnologie usate non sono le più recenti possibile, ma, nella maggior parte dei casi, sono più che sufficienti per ottenere i risultati desiderati. Poiché sono state usate tecnologie mature risulta evidente che un altro vantaggio di questi sistemi è la loro semplice programmabilità che permette di ottenere elaborazioni, anche complesse, senza doversi rompere la testa con algoritmi pipeline o con processi concorrenti o, peggio, con le complicazioni dell'elaborazione parallela che, forse non serve ricordarlo, richiedono, tra le altre cose, l'uso di costosi sistemi di sviluppo. I due punti di forza testé elencati permettono una elaborazione parallela e non concorrente dato che, per ogni tipo di manipolazione del suono che si desideri ottenere durante l'esecuzione in concerto, basta avere il relativo SAU opportunamente programmato per fornire quel dato effetto.

Da un punto di vista hardware il sistema SAU è costituito da un convertitore A/D a 10 bit, che può campionare segnali di frequenza fino a 20kHz, opportunamente dotato di sample & hold e filtro antialiasing con pendenza superiore a 48dB/oct, una sezione di elaborazione formata da un micro-processore 65C02 con clock a 4MHz, 8kbytes di RAM e 8kbytes di EPROM, un convertitore D/A a 12 bit con relativo filtro di ricostruzione con pendenza maggiore di 48 dB/oct, una sezione periferica che comprende un'interfaccia MIDI e una porta seriale con protocollo RS232; il tutto alloggia su una scheda formato Eurocard (160×100) e, tramite la sezione periferica, può essere facilmente connesso, nel caso l'elaborazione desiderata preveda un controllo in tempo reale, con un host.

Nella figura sotto è mostrato uno schema a blocchi del sistema:

Naturalmente, vista la genericità dell'architettura, ciò che interessa maggiormente saranno gli algoritmi per le varie elaborazioni sonore del caso. I relativi programmi trovano posto in EPROM e, realizzata un'applicazione per un concerto, per la successiva esecuzione sarà

sistema:



possibile cambiare completamente funzione semplicemente sostituendo la EPROM con quella contenente il software relativo alla nuova situazione. Le funzioni realizzate o in via di realizzazione sono:

- un PDC (Pitch to Digital Converter) che acquisisce un segnale acustico e ne misura la frequenza usando diversi tipi di algoritmi (conteggio tra gli zero-crossing, distanza tra i picchi positivi e negativi, etc.) sugli stessi campioni; la misura definitiva sarà prodotta da una decisione a maggioranza e sarà convertita nel relativo codice MIDI (con il key velocity pari all'ampiezza di picco della finestra di segnale analizzato).
- un sistema di modificazione non stazionario del suono tramite l'uso di tabelle di distorsione la cui forma varia in funzione del tempo che permette di ottenere effetti molto interessanti con strumenti musicali a suono fisso. La variazione delle tabelle viene ottenuta tramite un particolare algoritmo di interpolazione. Una variante di questo effetto permette di distorcere una sorgente concreta usandone un'altra come tabella di waveshaping.
- una "chorus machine" che genera un effetto coro sul segnale di ingresso sommando 8 fasi diverse (random o selezionabili) di quest'ultimo. Tale algoritmo, opportunamente variato, consente di ottenere un effetto simile a quello prodotto da un harmonizer o da un reverbero.

COMPOSIZIONE TRAMITE AUTOMI CELLULARI

Una tecnica per ottenere risultati complessi partendo da elementi semplici

T. Battista, M. Giri

Laboratorio per la musica Elettronica del Conservatorio "F. Morlacchi di Perugia"

Gran parte dei fenomeni complessi esistenti in natura sono il risultato dell'interazione di elementi estremamente semplici: il moto turbolento dei fluidi o la formazione di cristalli di neve sono processi le cui componenti obbediscono a leggi relativamente semplici. Scienze come la fisica e la biologia hanno da lungo tempo definito dei modelli per lo studio di queste componenti.

Da qualche anno si stanno sviluppando ricerche sulla complessità dei fenomeni tramite la simulazione al calcolatore di sistemi dinamici noti come automi cellulari. Questi sistemi sono caratterizzati da comportamenti locali molto semplici che producono risultati così articolati da sfuggire ai tradizionali strumenti di analisi.

Anche nel sistema musicale colto occidentale la complessità degli eventi sonori è generalmente ottenuta con la messa in gioco di elementi semplici: ci è sembrato quindi interessante sfruttare i modelli ad automi cellulari utilizzandoli non per simulare ma per generare fenomeni (musicali) complessi.

L'applicazione di questi modelli alla composizione musicale si è rivelata estremamente efficiente: intanto la semplicità del calcolo delle microcomponenti permette la traduzione delle strutture in eventi musicali e la loro esecuzione in tempo reale, senza bisogno di complicazioni intermedie; inoltre, tramite questo sistema, è possibile produrre una enorme varietà di strutture musicali. L'applicazione degli automi cellulari inaugura una serie di possibilità inedite per la composizione al calcolatore.

Definizione degli automi cellulari.

Possiamo immaginare l'automa cellulare come un "calcolatore" estremamente semplice che può assumere un piccolo numero di stati (generalmente due: 0,1). Ad ogni unità di tempo (generazione) lo stato dell'automa può cambiare seguendo delle regole date dal programmatore. Disponendo un certo numero di automi cellulari lungo una linea, questi si possono far interagire facendo dipendere lo stato di un automa dallo stato dei due automi vicini. Partendo da una certa configurazione di automi (in cui alcuni sono nello stato 0 ed altri nello stato 1), si può, per esempio, stabilire che lo stato di un automa alla generazione successiva sarà 1 se la somma degli stati degli automi vicini è 1 e sarà zero negli altri casi. È evidente che l'automa in questione influenzerà a sua volta gli automi vicini, e questi altri automi e così via.

Il calcolo dello stato successivo di un automa è semplice, ma la regola a cui obbedisce l'intera configurazione di automi è estremamente complessa e lo sviluppo delle generazioni nel tempo non è prevedibile se non statisticamente.

Una delle caratteristiche più interessanti di questo modello è la sua indeterminabilità globale, ottenuta a partire da semplici processi deterministici.

Un possibile impiego musicale degli automi cellulari.

Una fila di automi cellulari può essere letta come un numero binario; questo numero può essere interpretato come l'indirizzo di una tabella in cui sono codificati degli eventi musicali semplici (sotto forma, ad es., di informazioni MIDI).

Dividendo una linea di automi in gruppi di 8 bits otteniamo una serie di bytes che fungono da indirizzi per la tabella di eventi musicali. Questi eventi risulteranno correlati tra loro mediante le leggi che governano gli automi.

È possibile controllare la densità degli eventi regolando la percentuale delle informazioni contenute nella tabella; è anche possibile creare una tabella di cicli di ritardo che stabilisce, volta per volta, il numero delle generazioni di attesa tra la lettura di un evento e il successivo.

Variando opportunamente le leggi che governano gli automi, le configurazioni di partenza e le tabelle, è possibile gestire tutti i parametri di strutture musicali estremamente complesse.

Con questi sistema sono stati realizzati "Frammenti per Moebius" per nastro magnetico e "Amoebius" per clarinetto, contrabbasso e nastro magnetico. Per la realizzazione dei rispettivi nastri è stato utilizzato un sistema Yamaha TX816 controllato da elaboratore.

SESSIONI DI ASCOLTO

JUEGO DE VELOCIDADES

L. Todoroff

Questo lavoro costituisce la messa a punto di un tipo di composizione basato sull'uso sistematico di accelerandi e ritardandi.

Le variazioni di dinamica, l'uso del colore timbrico e la distribuzione in altezza del suono, determinano le espansioni, le compressioni e il moto direzionale generale del sistema.

Il brano è stato realizzato con il Computer YAMAHA CX5M e i suoi due software in dotazione.

FM Voicing Program - FM Music Composer.

Ladislao Todoroff.

Nato a Parana il 6 febbraio 1942, Argentina.

Studi Musicali di Violino e Composizione all'Università Nazionale di Musica della Città La Plata.

Integrante del Centro de Altos Estudios Musicales Latinoamericanos, diretto dal Compositore Alberto Ginastera; esegue nell'anno 1966 il primo brano di Musica Elettronica fatto in Argentina "Estudio 0".

Trasferito al Venezuela nell'anno 1976 svolge contemporaneamente le attività di Violinista dell'Orchestra Sinfonica di Maracaibo e di Compositore per i programmi educativi della Televisione Educativa dell'Università del Zulia.

Nel Concorso di Composizione del Festival di Musica Latinoamericana, Venezuela 1978; ottiene una Menzione Onorifica per la sua composizione per Orchestra Sinfonica "Hablemos del Comienzo".

Residente in Italia dall'anno 1982, continua a svolgere la sua attività di ricerca musicale nel suo studio e di violinista presso l'Orchestra del Teatro Regio di Torino.

TENDENZE

N. Sani

Per nastro magnetico

Tendenze è stato realizzato nel febbraio 1987 presso gli studi RAI-Radio Tre, nell'ambito del progetto di produzione di dodici composizioni elettroacustiche di musica radiofonica, ideato da Pasquale Santoli. L'organico utilizzato è stato il quartetto d'archi e sistemi per la sintesi e l'elaborazione del suono in tempo reale. Il materiale elettronico è costruito in base alla struttura del timbro degli archi ed è elaborato in modo da formare una cassa di risonanza e una struttura che amplia il materiale strumentale. Tutto il lavoro è caratterizzato dalle microvariazioni timbriche degli strumenti e dal loro "tendere" verso zone critiche causate dall'incontro di più elementi e dalla loro interazione.

Hanno partecipato alla realizzazione di *Tendenze*:

SPAGNOLETTI L. — sintesi ed elaborazione digitale del suono;

RIVELLI V. — tecnico del suono;

SCHIAFFINI G. — produttore esecutivo;

BUDEER S. — violino;

SUTTON P. — violino;

VARTOLOMEI D. — viola;

CAVALLO V. — violoncello.

Nicola Sani.

Nato a Ferrara, nel 1961, vive a Roma. Ha studiato con Domenico Guaccero, Antonio Scarlato, Guido Baggiani, Giorgio Nottoli. Sue composizioni sono state eseguite in Italia e all'estero. Ha avuto commissioni dalla WDR di Colonia e dalla RAI. Dal 1984 è amministratore delegato della SIM, uno dei principali centri europei nell'informatica musicale. Collabora con giornali e riviste tra cui l'Unità, Paese Sera, Rockstar. È redattore di Musica/Realtà. Come operatore musicale ha organizzato numerose manifestazioni in Italia e all'estero. Collabora con il Festival Ars Electronica di Linz ed è responsabile del settore musica del Festival dell'Arte Elettronica di Camerino.

ARCHI ELETTRICI

G. Tedde

Questa composizione è nata come esecuzione/realizzazione col sintetizzatore di una musica concepita originariamente per un organico strumentale acustico.

La musica di partenza è stata scritta da me nel 1986 per quattro flauti con il titolo "ARCHI" ed ha vinto il premio "EVANGELISTI" dello stesso anno a Roma.

Essa si presta facilmente a questa trasposizione perché è basata sullo studio degli armonici dei suoni del flauto, che sono stati realizzati sul sintetizzatore con un attento dosaggio dell'apertura e della chiusura dei filtri.

Il lavoro è stato prodotto nell'aula di musica elettronica del Conservatorio di Cagliari, e sono stati impiegati due registratori ed un sintetizzatore.

Voglio qui ringraziare l'insegnante del corso di musica elettronica di Cagliari, Nicola Bernardini, per i preziosi consigli che mi hanno permesso di portare a termine un buon lavoro.

Giorgio Tedde.

Nato a Cagliari nel 1958 ha compiuto nella stessa città i suoi studi al Conservatorio e, all'Università ottenendo il massimo dei voti e la lode nel Diploma di composizione e nella Laurea in Fisica.

Ha partecipato ai Colloqui di Informatica Musicale con contributi tecnici (Ancona, 1983) e teorici (Napoli, 1985), e nel 1986 ha preso parte all'International Computer Music Conference (L'Aja, 1986) con l'articolo "Phenomenology of Musical Communication".

Ha partecipato con le sue opere ad importanti festival e rassegne di musica e arte contemporanea sia in Italia (Roma, Torino, Siena, Cremona, Cagliari, etc.) che all'estero (Darmstadt, Parigi, Buenos Aires, Rio De Janeiro, etc.), e i suoi lavori sono stati registrati e diffusi dalla radio nazionale italiana, tedesca, belga, argentina, brasiliana, etc..

È stato edito in Italia da EDIPAN e all'estero da Edition Transatlantiques, ed ha inciso su disco due lavori.

Le sue composizioni hanno avuto riconoscimenti e premi in Italia (Roma, 1986, Messina, 1987) ed all'estero (Spittal and der Drau, 1986/87).

Attualmente insegna Cultura Musicale Generale presso il Conservatorio di Sassari.

TRA I SUONI ESTREMI

F. Galante

Via Gussone 38 - Roma

Questa opera prosegue lungo l'itinerario tracciato dai miei precedenti lavori elettronici, in cui mi sono preoccupato di studiare e organizzare musicalmente fenomeni percettivi che si realizzano lavorando all'interno delle bande critiche.

Il risultato cercato è quello di costruire un processo sonoro derivato dalle opportune aggregazioni di eventi semplici (magari esclusivamente sinusoidali) sintetizzati a distanze frequenziali che cadono in dette zone critiche oppure proiettati lungo l'arco di uno spazio acustico dilatato.

"Tra i suoni estremi" esce da un contesto tradizionale di dialettica delle parti, ma è un processo continuo, microvariato, in cui l'informazione sonora è il risultato contemporaneo delle forme acustiche e psicologiche derivate dal movimento nel suono, dallo spazio percettivo che esso crea in relazione alla memoria individuale dell'ascolto.

"Tra i suoni estremi" è stato realizzato nel 1987 presso la S.I.M. di Roma tramite un sistema originale digitale di sintesi del suono realizzato presso i laboratori della società e controllati da personal computer.

La sintesi del suono è del tipo additivo di tabelle.

Francesco Galante.

Nato a Roma nel 1956, ha compiuto studi scientifici presso l'Università di Roma e di composizione musicale elettronica presso il Conservatorio di Frosinone con Giorgio Nottoli.

Nel 1979 ha frequentato i corsi internazionali di musica elettroacustica presso il G.M.E.B. di Bourges, studiando con P. Boeswillwald, G. Baggiani e D. Keane.

Nel biennio 1980-81 è stato responsabile della Associazione Musica Verticale.

È tra i fondatori della Società di Informatica Musicale di Roma.

I suoi lavori sono stati eseguiti in importanti manifestazioni in Italia e all'estero.

DEVE ESSERE TENUTO LONTANO DA FONTI DI LUCE

R. Doati

Centro di Sonologia Computazionale dell'Università di Padova - (1985/86)

La composizione fa uso di un lessico costituito da pochi elementi semplici subordinati a leggi compositive che non si riducono ad associazioni cumulative, ma conferiscono proprietà d'insieme distinte da quelle degli elementi.

Un primo gruppo di regole è dato dall'ordinamento in scala dei parametri a basso livello che determinano la costruzione dei singoli suoni: numero di parziali, ritardi di entrata, durate fisiche relative, involuppi d'ampiezza, involuppi spettrali, tempi d'attacco, deviazioni di frequenza e di ampiezza, ecc..

A livello superiore è collocato uno spazio timbrico a due dimensioni che controlla l'armonicità degli spettri: espansione-compressione, traslazione. Esse determinano uno dei fattori indagati da Steve McAdams nel suo studio sulla formazione di immagini uditive.

Le regole formali, infine, sono dapprima utilizzate per l'organizzazione dei singoli elementi in strutture ritmiche, e in seguito per l'organizzazione di tali strutture nello spazio compositivo.

Carattere fondamentale di una struttura in quanto organizzazione di elementi significanti è l'autoregolazione (la struttura controlla e regola sé stessa, diviene fenomeno naturale). Ora il ritmo assicura la propria autoregolazione con i mezzi più semplici quali simmetrie e ripetizioni. Da ciò deriva l'uso di strutture ritmiche semplici e regolari con un profondo carattere di ripetizione. Il reticolo timbrico sopra descritto sottolinea, e più spesso altera questa autoregolazione ponendo in conflitto organizzazione ritmica e organizzazione timbrica.

Suoni concreti isolati rappresentano la relazione fra il mondo sintetico e quello reale.

Per un'analisi più approfondita si rimanda all'intervento "Processi cognitivi come parametri compositivi" pubblicato in altro luogo in questi atti.

Roberto Doati.

Ha studiato musica elettronica con Albert Mayr presso il Conservatorio "L. Cherubini" di Firenze, informatica musicale con Pietro Grossi presso l'Istituto di Elaborazione dell'Informazione del CNUCE-CNR di Pisa e con Alvisè Vidolin presso il Conservatorio "B. Marcello" di Venezia dove si è diplomato con la tesi "Un'applicazione musicale della teoria gestaltica sulla percezione di strutture temporali". Nel 1978 ha preso parte al "Summer Workshop in Computer Music" organizzato dall'UNESCO presso l'Università di Padova e l'Istituto di Musicologia dell'Università di Aarhus (Danimarca).

Dal 1979 lavora come compositore e ricercatore nel campo della percezione musicale presso il Centro di Sonologia Computazionale dell'Università di Padova e i risultati delle sue indagini sono presentati nel corso dei Colloqui di Informatica Musicale che si svolgono periodicamente presso le più importanti Università italiane. Le sue opere sono state eseguite in numerosi festival di musica contemporanea internazionali e diffuse radiofonicamente. Per la composizione "Gioco di velocità" (incisa su disco EDIPAN), è stato invitato nel 1981 a partecipare alla manifestazione "Venezia Opera Prima" organizzata dal Teatro "La Fenice" in Venezia e dedicata ai giovani compositori. Nel 1984 ha realizzato, insieme ai compositori Patella e

Torresan, un'opera elettronica commissionata dalla RAI per la radiodiffusione.

Svolge attività didattica nell'ambito dei Corsi di Informatica Musicale istituiti dal CSC di Padova e attraverso corsi e conferenze itineranti.

Dal 1983 al 1987 è stato Segretario dell'Associazione di Informatica Musicale Italiana (AIMI). Nel 1983 inizia a collaborare con La Biennale di Venezia in veste di conduttore del Centro Documentazione Musica Contemporanea (responsabile Mario Messinis), e del Laboratorio di Informatica Musicale (LIMB) (responsabile Alvisè Vidolin). Per la Biennale ha inoltre curato, con Mario Messinis, la sezione "Europa 50/80: generazioni a confronto" del XLII Festival di Musica Contemporanea e, nel quadro delle attività del LIMB, i seminari "Il Sistema 4I e il tempo reale", "Interpolazioni" e, recentemente, l'esposizione "Nuova Atlantide - Il continente della musica elettronica 1900-1986".

TING

K.F. Gerber
Member of CMA - Gema

The basic sequence of "Ting" consists of eight successive "Soundwedges". Upon these other units of variable texture are added. The idea for the shaping of sound was to superpose fast percussive notes of a 16-voice FM-synthesizer module. This was controlled dynamically through the MIDI-bus, just as the digital effect-processor applied.

Using the 68000-processor-based ATARI ST computer I developed the routines for composition and control for the movement of the masses of sound.

The transitions composed from staccato to textures of clusters and noise bands finally have been achieved after a fine-adjustment of the FM-synthesizer's envelopes to the extremely fast running note informations.

The recording was done in analog technique in studio "Lupus Polaris" in Munich, Germany on May, 27th 1987. The mixdown is due to Penko of CHAOS SOUND UNLIMITED.

Production systems: ATARI ST, Tascam 38, Yamaha SPX90, Yamaha TX7. Software by the composer.

Performance information: Tape 1/4 inch 2-track stereo 38 cm/sec. Haeder for OdBmad-justment followed by piece of 8"25.

Amplify loud as acceptable but clearly beyond the level of violating the audience's tympanum (organ of hearing).

Karl F. Gerber.

Born 1954 in Lörrach, West-Germany.

Began to play electric bass at the age of 16 and double-bass at the age of 20.

Studied physics and musicology at the Albert-Ludwigs-Universität Freiburg im Breisgau and received S.Sc. of Physics from the Ludwig-Maximilians-Universität München, Munich, Germany.

He studied double-bass at the Jazz School Munich with Adelhard Roidinger and Joe Haider and worked with modern jazz groups from 1979 till 1984.

1975 first experiments with live-electronics within his jazz group "integrated circuit"; development of sequencer-hardware.

From 1979 on: interested in common symmetries in arts and science.

1984 software for real-time improvisation on home-computer, written in machine-language.

Since 1985 working on the "Gerber MIDI-Improviser"-system, a realtime-program for improvisation. 1987 computer-controlled interactive sound-installation for the art exhibition "Perseus", at GALERIE X, Munich, lasting six weeks!

Is convinced that sound-synthesis, composition, performance and the construction of instruments should not be left to specialists.

RAN

M. Pedrazzi

RAN è stato realizzato tra Aprile e Maggio dell'87, in parte al Dipartimento di Insiemistica Sistemistica e Telematica (DIST) dell'Università di Genova ed in parte nello studio dell'autore.

Per la sintesi degli eventi sonori è stato utilizzato il linguaggio di programmazione CMUSIC implementato su VAX, più Microvax e PDP per la sintesi audio con l'assistenza tecnica di Corrado Canepa, ed altri sistemi digitali (campionatore, sintetizzatore).

Le sintesi utilizzate sono additiva, moltiplicativa e campionamento.

RAN vuole essere un divenire sonoro dove alla creatività umana si sostituisce, in maniera diversa, un controllo pseudocasuale e quindi un alternarsi, a diversi livelli di generazione, del processo creativo-compositivo da parte della macchina e da parte dell'uomo.

La aleatorietà del processo creativo della macchina, in ogni caso delimitato a priori, all'interno del processo creativo umano, ne determina le scelte.

RAN (HERAN), parola tedesca, ha un significato di moto verso sé, di avvicinamento, di venire incontro ed è in questo spirito, in questo vicendevole venirsi incontro tra casualità e determinazione che si realizza l'equilibrio.

Quindi non due opposti incompatibili ma complementari, ognuno stimolo dell'altro.

Marco Pedrazzi.

Nato a Modena nel 1959.

Studi musicali e corsi di composizione al Conservatorio "G. Verdi" di Milano col maestro Giovanni Battista Zotti.

Inizialmente autodidatta segue studi di Musica Elettronica e Computer Music a Milano prima al CEDME, poi all'EMIT col prof. Goffredo Haus ed al Centro di Sonologia Computazionale dell'Università di Padova col compositore James Dashow.

Dal 1984 si occupa di Musica Elettronica ed Informatica Musicale e svolge attività compositiva nel suo studio a Milano.

Ha rapporti di collaborazione col Dipart. di Insiemistica Sistemistica e Telematica dell'Università di Genova.

La sua prima esecuzione pubblica avviene a Milano nel 1982 con un brano di musica da camera per trio.

Ha scritto diversi brani di musica da camera ed ha realizzato musiche su commissione per presentazioni commerciali.

Attualmente collabora in uno studio di registrazione per produzioni musicali.

BIRDWATCHING

B. Fagarazzi

Centro di Sonologia Computazionale dell'Università di Padova

L'idea alla base di questa composizione è nata dalla teoria dei frattali di B.B. Mandelbrot.

Questa teoria, ormai ben nota, postula l'autosimilarità quale principio informatore dei processi naturali, dalla struttura dell'atomo alla distribuzione delle galassie.

Fin dal 1982 lavoro alle applicazioni musicali di questa teoria ed in particolare del "volo di Lévy".

"BIRDWATCHING" è una composizione che corona tutto il lavoro di questi anni cercando di esprimere un concetto di evento musicale il più possibile svincolato da schemi strutturali deterministici e quindi da condizionamenti socio-culturali, cercando inoltre di superare la dicotomia tra armonico e inarmonico lasciando la decisione all'evoluzione naturalmente autosimile dell'evento sonoro.

Il principio dell'autosimilarità viene rispettato anche nella struttura basata su sequenze generate secondo l'algoritmo del "Volo di Lévy" ed organizzata in maniera gerarchica in una struttura dinamica di tipo non deterministico. Dette sequenze si evolvono su spazi timbrici paralleli secondo una partitura finale generata basandosi sempre sul "Volo di Lévy".

L'impronta che si è voluto dare a questa composizione è un continuo alternarsi e sovrapporsi di voli la cui coerenza non è decisa a priori ma è virtù dell'interazione autosimile tra materiale sonoro e struttura.

L'effetto ottenuto ricorda l'alternarsi di attese e disattese di aspettative suscitato dall'osservazione del volo degli uccelli.

Bruno Fagarazzi.

Nato a Venezia nel 1953.

Si è laureato in ingegneria elettronica presso l'Università di Padova nell'a.a. 1977 ed ha studiato Computer Music al Conservatorio "Benedetto Marcello" di Venezia sotto la guida di Alvise Vidolin, concludendo nel 1986.

Collabora al CSC fin dal 1983 dove si occupa di composizione automatica.

Ha partecipato al V e VI C.I.M..

"Birdwatching" è la sua opera prima.

CAMPANA DEL TRAMONTO

S. Petrarca

“Campana del Tramonto” è l’articolazione strutturale di elementi lessicali quali: uno schema di entrate, una modalità di costruzione timbrica e una legge di variazione delle frequenze. Tali elementi di base sono presentati a vari livelli strutturali e assumono, nel corso del tempo, significati differenti in un’articolazione formale di tensioni e rilassamenti - anticipazioni e riferimenti al già avvenuto - dovuta sia alla loro presenza contemporanea e costante, ma continuamente variata e permutata, sia ad una legge globale di variazione non lineare della densità verticale. Quest’ultima è strettamente legata alle leggi di variazione che sono insite nel lessico del brano, garantendo così l’unità linguistica del lavoro.

Che non tragga in inganno il titolo: “Campana del Tramonto” non è musica descrittiva né impressionista né, tantomeno, un poema sinfonico. Il riferimento del titolo nulla ha a che vedere con alcuni suoni, presenti in alcuni punti del lavoro, che possono ricordare, appunto, delle campane; la presenza e la composizione spettrale di tali elementi sonori hanno una motivazione strettamente strutturale (ebbene sì! Essi sono la “degenerazione” di quella fascia acuta che appare all’inizio del brano)

Stefano Petrarca.

Nato a Roma nel 1954, ha compiuto privatamente studi di pianoforte, chitarra e composizione; ha studiato tromba con M. Bartoletti. Nel 1981 ha conseguito la laurea in Matematica presso l’Università di Roma con una tesi sulle tecniche matematiche applicate alla musica. Nel 1983 si è diplomato in musica elettronica con Giorgio Nottoli presso il conservatorio “L. Refice” di Frosinone.

Attivo nel campo della musica elettronica dal 1977, per 3 anni ha lavorato all’Istituto di Acustica del CNR sulla sintesi digitale del suono. Attualmente è responsabile dello Studio per la ricerca elettroacustica musicale (STREAM) di Roma dove svolge attività sia di musicista che di ricercatore.

VILLOTTA

M. Querzola
Soprano E. Negro

Nastro realizzato presso il Centro di Sonologia Computazionale Università di Padova
Realizzazione all'elaboratore di A. Provaglio

Villotta per soprano e nastro magnetico elaborato al computer nasce dalle suggestioni provocate dalla lettura di una poesia di P.P. Pasolini intitolata "Canto delle campane" tratta da Seconda Forma de "La meglio gioventù" Ed. Einaudi che riporta i seguenti versi:

« Non rimpiango una realtà ma il suo valore / Non rimpiango un mondo ma il suo colore /
Tornando senza corpo là dove le campane cantavano parole di dovere sorde come tuoni /
non piango perché quel mondo non torna più, ma piango perché il suo tornare è finito /
Sono restato con tutto: solo senza il più grande inganno, quello che pareva la ragione del
[vivere mio e del mondo:
torno, passando sui ponti crollati, come un australiano ».

La struttura musicale ricalca sotto certi aspetti la musica folclorica Friulana; reitera quattro volte in senso retto e quattro in senso inverso una serie di dodici frequenze. L'elaboratore appoggia la voce con l'alone delle campane prendendone spunto dal suono percussivo delle stesse; opera dei glissandi e dei "cluster" e mantiene la coerenza del discorso musicale sia nella manipolazione degli effetti sopra enunciati che ricalcando le frequenze proposte dalla voce.

Marco Querzola.

Nato a Udine il 6 agosto 1960 e diplomato in pianoforte presso il Conservatorio di Musica "J. Tomadini" sotto la guida della prof.sa M.G. Cabai, si è in seguito perfezionato sotto la scuola del Maestro G. Lovato. Ha frequentato la Scuola di Clavicembalo e Musica Antica tenendo concerti in pubblico e riscuotendo consensi.

Frequenta attualmente la Scuola di Composizione presso il Conservatorio Statale di Musica "J. Tomadini" sotto la guida del Maestro D. Zanettovich e la Scuola di Musica Corale e di Direzione di Coro sotto la guida del Maestro G. Zanetti.

Ha scritto inoltre: QUAESITIO per Soprano Oboe e frequenza su testo di M. Grattoni. MACRO STRUTTURA per Violino Viola Violoncello Contrabbasso Voce recitante e frequenze su testo di F. Kafka TRE balletto per Flauto Arpa Pianoforte.

HY-LUR

A. Di Scipio
per nastro magnetico

La realizzazione di questo brano è avvenuta nell'inverno 1986-87; la prima esecuzione assoluta, nella versione per tromba e nastro, è stata data il 21 maggio 1987 al Teatro S. Agostino in L'Aquila, da A. Tonelli, tromba.

La sintesi numerica del suono è stata curata sul sistema "Fly" nella versione prototipo che M. Lupone mi ha gentilmente messo a disposizione, con il software di controllo su Apple II di L. Bianchini.

Ai suoni digitali sono stati sovrapposti suoni di tromba; tutti i segnali sono stati infine processati attraverso un Lexicon PCM 70, e mixati presso lo Studio di Musica Elettronica del Conservatorio dell'Aquila.

"Hy" è la parola con la quale gli Egizi indicavano "musica" e "gioia", concetti presso di loro inscindibili.

"Lur" è il nome del più antico aerofono bronzeo ritrovato sulle coste baltiche della Scandinavia (primo millennio a.C.).

Agostino Di Scipio.

Nato a Napoli il 1962, studia composizione e musica elettronica presso il Conservatorio dell'Aquila; ha inoltre approfondito gli studi con R. Boesch (Bourges, algoritmi di sintesi sul DSP TMS320), J. Daschow (Padova, "Composizione e sintesi") e R. Doati (Padova, "Psicoacustica per la musica informatica").

Svolge attività divulgativa con concerti e seminari introduttivi.

È autore di brani strumentali ed elettronici, questi ultimi realizzati mediante elaboratore.

PER LO DOLCE SILENTIO DE LA NOTTE

A. Caprioli

Per pianoforte e nastro magnetico elaborato dal processore 4i
strumento e ambiente esecutivo di Sylviane Sapir

Centro di Sonologia Computazionale dell'Università di Padova (processore 4i: progetto
Giuseppe Di Giugno, IRCAM, Parigi; corealizzazione Laboratorio di Informatica
Musicale della Biennale di Venezia)

Breve nota sulla forma della composizione.

La composizione "Per lo dolce silentio de la notte", presentata in prima assoluta a Salisburgo, nell'ambito dei Concerti del "Mozarteum" (Wiener Saal, 26 maggio 1987) dal pianista Marco Raspanti - di cui viene presentata all'ascolto in questo Colloquio la registrazione -, eseguita (dopo la prima esecuzione polacca a Cracovia) per la prima volta in Italia nel dicembre 1987 dal pianista Bernardino Beggio a Padova, nell'ambito del "Computer & Art Festival 1987", comprende nel proprio organico due strumenti:

- il pianoforte, suonato dal vivo da un esecutore, che utilizza, tra gli altri, due serie particolari di armonici dello strumento;
- il processore 4I dal vivo (sostituibile da nastro), con uno strumento e un ambiente esecutivo progettati da Sylviane Sapir, che permettono una sintesi e una spazializzazione del suono attraverso quattro canali. Tale strumento ("SARQUA") è costituito dall'ampliamento (raddoppio delle possibilità polifoniche a 8 voci e aggiunta della funzione di spazializzazione) del precedente strumento "INTER 2", utilizzato per la prima esecuzione dell'opera "Prometeo" di Luigi Nono.

L'algoritmo di sintesi (riportato qui nella descrizione di Sylviane Sapir) è composto da otto voci, tutte con la stessa struttura. Esse sono miscelate in uscita fra loro, e l'uscita del mixer viene spazializzata attraverso quattro canali indipendenti. I parametri di ingresso di questo algoritmo vengono gestiti in tempo reale, oppure secondo i dati indicati nella partitura. Nei dettagli, l'algoritmo di ciascuna voce è ispirato a quello già utilizzato per la realizzazione dell'ambiente esecutivo "INTER 2" dell'opera "Prometeo" di Luigi Nono, ed è stato descritto nel bollettino "LIMB" n. 5, S. Sapir, A. Vidolin, Venezia, 1985. Ricordiamo che esso può essere diviso in due parti: una parte di generazione del segnale, mediante la tecnica di distorsione non lineare (modulo DNLS), e una parte di controllo automatizzato, per le microvariazioni di alcuni parametri (indice di distorsione, frequenza base, frequenza della modulazione di ampiezza). Questa tecnica permette di distorcere una funzione sinusoidale mediante una funzione di distorsione, al fine di produrre un suono complesso, il cui spettro varia dinamicamente in funzione dell'indice IDX. Questo indice dipende a sua volta dall'andamento di un'ulteriore funzione. Il suono finale è involupato grazie all'uso del generatore di segmenti di rete (LUNI). La scelta dei vari parametri e delle funzioni permette di creare dei suoni che ricordano le risonanze di bicchieri o campane di vetro. Le uscite delle otto voci sono miscelate digitalmente; i livelli di ingresso del mixer sono controllabili da partitura, e consentono di bilanciare le varie voci tra di loro. Il segnale in uscita del mixer viene sommato a sé stesso tre volte, per recuperare in dinamica prima della spazializzazione e della conversione

finale. La spazializzazione è effettuata mediante quattro oscillatori, che involuppano il segnale verso le quattro uscite.

Essi dispongono della stessa funzione F16, e sono sfasati fra di loro di un quarto di periodo. In questo modo si ottiene la circolazione del segnale attraverso i quattro canali. La funzione F16 determina il percorso del suono nello spazio.

L'ideazione, ispirazione e definizione della forma del brano sono basate su due testi letterari analizzati strutturalmente, ai quali si è data una serie di corrispondenze musicali, secondo la definizione di un rapporto tra struttura letteraria (lettura "forte" del testo) e struttura musicale. Inoltre, l'interazione di musica analogica e musica digitale è collegata alla "lettura" sovrapposta dei due testi letterari. La struttura intenzionale letteraria (particolare struttura della rima nella canzone petrarchesca) è in rapporto con la struttura intenzionale musicale data dalla spazializzazione. I testi letterari sono:

- 1) la Canzone CCXXXVII dai "Rerum Vulgarium Fragmenta" di Francesco Petrarca;
- 2) due brani dalle Elegie di Tibullo (I,3,89-94; II, 1, 87-90).

Il motivo che unisce idealmente le due riletture è quello dell'interpretazione del "topos" letterario della notte, nel rapporto tra l'alessandrinismo di Tibullo e quello di alcune canzoni (come appunto quella in oggetto) di Petrarca, il cui schema compositivo non pare affatto estraneo a certa "musica reservata" tardo-rinascimentale; questo rapporto di continuità delle epoche alessandrine ha suggerito all'autore di riportare (trepidante, e perciò timoroso e discreto) alla luce alcune voci del mito della notte, così come esso si presenta ai giorni contemporanei: ritrovando gli echi, le evocazioni o nostalgica ironia di talvolta, cui esso volentieri riconduce, per conoscere ciò a cui si vorrebbe sfuggire, e a cui pure si indugia.

E, similmente, l'atmosfera e la scelta degli strumenti (il pianoforte trattato con aura di "Notturmo", e il suono di sintesi del processore interagente con quarti e ottavi di tono con la parte del pianoforte) vogliono realizzare le reali o surreali reincarnazioni del mito della notte e delle sue evoluzioni nella mente e nel "suono interiore" dell'uomo contemporaneo, alla ricerca e nel timore del "proprio" silenzio ("Già mai tranquilla notte", F. Petrarca, R.V.F. CCXXXVII, 13).

Alberto Caprioli.

Nato nel 1956 a Bologna, all'attività musicale di compositore e direttore d'orchestra affianca gli studi letterari presso la Facoltà di Lettere e Filosofia dell'Università di Bologna. Ha compiuto gli studi di composizione con Franco Margola e Camillo Togni, partecipando a Seminari di Karlheinz Stockhausen, Boguslaw Schaeffer (presso cui è stato invitato a frequentare un seminario al "Mozarteum" di Salisburgo), Franco Donatoni, Gilberto Bosco, del C.S.C. a Padova e del L.I.M.B. a Gorizia.

Dal 1979 al 1983, a seguito di una Borsa di Studio del Ministero Austriaco delle Scienze, ha compiuto gli studi di direzione d'orchestra con Otmar Suitner presso l'Accademia di Vienna; contemporaneamente ha partecipato ai Corsi delle Accademie di Weimar e di Nizza, della Radio Olandese con Kirill Kondrashin, dell'Arena di Verona con Franco Ferrara, dell'Accademia Chigiana di Siena con Carlo Maria Giulini. Ha diretto presso la Radio Austriaca e il Musikverein di Vienna, l'Opera di Stato di Berlino, in Romania e in Italia, dove ha debuttato nel 1983, dirigendo a Bologna l'Orchestra Filarmonica di Dresda.

Dal 1983 al 1987 la Radio Austriaca di Salisburgo ha registrato e trasmesso alcune sue

composizioni di musica da camera, una delle quali ha ottenuto una menzione d'onore al Concorso Internazionale di Avignone.

Sue composizioni sono state registrate dalla Radio Tedesca SDR di Stoccarda, dalla RAI 3, ed eseguite nei Festival di Salisburgo, Cracovia, Padova, Udine, Perugia.

In occasione della "Internationale Musikfest" di Stoccarda 1988, la Radio Tedesca SDR gli ha commissionato una composizione cameristica.

È in corso di preparazione, presso la casa discografica Sonoton di Monaco di Baviera (serie "Pro Viva"), un disco monografico comprendente alcune delle sue composizioni.

RITMI BIOLOGICI
per flauto in sol o in do amplificato e nastro magnetico

M. Biasutti

Ritmi biologici è un lavoro che intende proporre un nuovo modo di concepire il tempo, il ritmo musicale in una dimensione più corporea ed a misura d'uomo. Tale dimensione viene ricercata non soltanto lasciando all'esecutore una maggior libertà interpretativa, ma trasponendo alcuni elementi biologici in musica. È stata infatti presa in considerazione la ciclicità del ritmo respiratorio, elemento primordiale della vita, e trasposto nella tecnica flautistica, sfruttando, per la produzione di suoni, sia la fase di espirazione che quella di inspirazione. Nel brano è individuabile questa ciclicità ritmica biologica.

In questo lavoro, concepito a sezioni, vengono anche approfondite le possibilità del flauto di produrre suoni ad altezza indeterminata a seguito di una serie di laboratori condotti su alcuni strumenti a fiato. Sulle quattro sezioni del nastro magnetico, costituito da suoni concreti campionati, viene ad inserirsi il flauto amplificato in un continuo intrecciarsi timbrico e dinamico con il nastro. La parte flautistica è caratterizzata da una continua ricerca sonora verso i timbri eolici del flauto, particolari per il loro carattere misterioso e introspettivo.

Michele Biasutti.

Michele Biasutti, nato a Udine, ha studiato flauto con Mariann Fischer e musica elettronica con Alfonso Belfiore presso il conservatorio Pollini di Padova. Nel corso degli studi ha partecipato ad alcuni corsi di perfezionamento flautistico.

Si è interessato di strumenti antichi collaborando ad una serie di conferenze-concerto sulla storia del flauto.

Nel 1985 ha formato con altri artisti il Contempo Ensemble, gruppo per la ricerca e la diffusione della musica contemporanea. L'attività del gruppo è molteplice: oltre a favorire la collaborazione fra artisti di vari settori partecipa a rassegne e realizza registrazioni radiofoniche.

Attivo anche come compositore ha ottenuto il terzo premio assoluto al IX Concorso Internazionale di Musica elettroacustica "L. Russolo" di Varese.

Parallelamente agli studi musicali ha continuato gli studi universitari laureandosi con il massimo dei voti in Psicologia a Padova.

Attualmente sta pubblicando il lavoro di tesi, uno studio teorico sperimentale sulle attitudini musicali realizzato con la collaborazione bibliografica di esperti stranieri.

Autore di numerose conferenze e lezioni a carattere musicale per conto di diverse associazioni culturali, collabora con la casa di Edizioni Musicali Bancaneve con la quale pubblica i suoi lavori.

Tiene una rubrica di musica sulla rivista di poesia "zeta".

ROTA
for Piano or Harpsichord and Tape

R. Růžicka

Description of the computer program for the composition.

The length of each composition is determined by the number of bars or tones and rests. It is necessary to define the proportion of tones and rests, ornamented and normal notes, the proportion of tremolos (trills, rolling, frullatos etc.) and groups (leaning notes, grouplets, arpeggios etc.). The range of melody is determined by defining the lowest and the highest Pitch, the set of tones that should be used is defined by the type of mode. The selection of rhythmic values, melodic intervals and intervals between tones in chords is based on chosen distribution functions. Tremolo is defined by the interval between the two tones, the number of tones in ornaments by the minimum and maximum number, polyphony by the number of parts. Two types of transposition (modal and instrumental) allow modifications of basic modes and automatic corrections of the parts of transposing instruments. The selection of tones, rhythms and some other elements is done through pseudorandom number generation according to a defined discrete distribution. Separate utility programs extract individual parts from the score using for each instrument a suitable clef and signs and allowing the use of traditional notation.

The Program can also be used for research in the field of artificial intelligence, especially for research on human thinking by means of imitating the activity of the brain through the use of random processes and their transformation by the composer and by the computer into a piece of art.

The composition has been recorded in the Electroacoustic Laboratory of the Czechoslovak Radio in Plzen with the collaboration of Programmer Dr. Josef Gerbrich and sound engineer Ing. Cesmir Kadlec.

Rudolf Růžicka.

R. Růžicka (born in 1941) studied composition at the Janacek Academy of Performing Arts in Brno. Later he went his studies taking a two-ager course of electroacoustic music in Prague. Since 1969 he has been teaching at the Janacek Academy where he lectures on composition of serious contemporary music using computers in the courses for graduate students.

Mr. Růžicka has written more than thirty compositions with the help of computer. His first computer composition is "Electronia B" from 1965. Other important instrumental, vocal and electroacoustic computer composition are: Cosmic Symphony for large orchestra, Cantata ai ai a, Concerto - Symphony for violin and orchestra, Discordia, Malefica, Arcanum.

R. Růžicka won first Prizes at the international competitions of electroacoustic music: "Musica Nova" 1970 (composition "Gurges"), competition Marcel Josse 1984 in Paris ("Tibia I.") and further Prizes, and honourable mentions. He was named a member of jury of the "International Computer Music Competition Newcomp 1987" in Boston.

JARDIN SECRET I

K. Saariaho

Jardin Secret I (Secret Garden 1984-85) is a first etude in a series of works, in which I have concentrated myself in making research with computer on the relations between different musical parameters and processes, and at the same time my own musical thinking.

I shaped primary models for few different, well-defined musical situations. Starting from these models, I intended to control various musical parameters, without forgetting their particular features. I wanted to find out whether the various parameters could be treated starting from the same outset point.

Since I have been interested for a long time in the idea of (musical) interpolation, my starting-point was also partly determined by this interest in relation to various specific musical and physical parameters.

The dominating type of process in this etude is a gradual interpolation between two points, realized on different parameters in different scales. Sometimes this process is combined with sharp transitions between different characteristics in other parameters.

The colour and tone levels of the sounds are intertwined into an indissoluble knot, for as one operates with the different physical parameters of sound independently one exposes its structure, breaks its "uniform surface", and thus a single tone is a many-dimensioned sound process with its colour and tone-level.

The programs have been entirely realized with the FORMES system which runs on IRCAM's VAX 11/780 computer and can be linked with the CHANT sound synthesis and processing program on the array processor FPS-100. All the sound material for the piece have been synthesized with this version of CHANT.

The mixing of the sound material was made with Sony 24-track digital tape recorder.

Kaija Saariaho.

Kaija Saariaho was born in Helsinki 1952. After study of several instruments and the visual arts, she began her composition studies under Paavo Heininen at the Sibelius Academy in Helsinki, and continued them in Freiburg, West Germany under Brian Ferneyhough and Klaus Huber. Since 1980 Saariaho has worked in several electroacoustic studios: in the Finnish Radio's experimental studio, the Sudwestfunk's studio in Freiburg, IRCAM's and GRM's digital studios in Paris, and at EMS in Stockholm. She has received the Finnish Government's artist salary for the years 1983-86 and 88-92 for her compositional work and the Fulbright Grant, and was the recipient of the Kranichsteiner Preis at Darmstadt 1986. She also received a mention of honour in the Koussevitzky Prize 1987 for the recording of *Verblendungen* (BIS-LP, The Finnish Radio Symphony Orchestra, dir. Esa-Pekka Salonen).

Saariaho is working at IRCAM (Paris), using the computer to research her compositional ideas.

Among her recent works are *Verblendungen* (orchestra and tape, 1982-84), *Lichtbogen* for chamber ensemble and electronics (1985-86), *Jardin Secret I* (tape, 1984) and *Jardin Secret II* (harpsichord and tape, 1984-86), and *Io* (1986-87) for ensemble, tape and electronics,

a commission for the Ensemble Inter Contemporain to celebrate the 10th anniversary festival of the Pompidou Center. Currently Saariaho is working on a string quartet with electronics, a commission from the Lincoln Center for Kronos Quartet.

Saariaho's music is frequently played throughout Europe, as well as in the USA, Canada and Japan, and is available on the BIS, Jaselp and Finlandia record labels.

