



Università degli Studi di Udine

Dottorato di ricerca in Studi storico artistici e audiovisivi
Curriculum audiovisivi

Tesi di dottorato di ricerca

ENTRANCE

SPETTRALISMO E COMPOSIZIONE ASSISTITA ALL'ELABORATORE IN FAUSTO ROMITELLI

Dottorando
Alessandro Olto

Relatore
Chiar.mo Prof. Angelo Orcalli

Anno Accademico 2015–2016

INDICE

INTRODUZIONE.....	5
CAPITOLO 1: LA RIVOLUZIONE DEI SUONI COMPLESSI	9
1.1 Il suono. La complessità svelata.....	9
1.1.1 <i>Dai risonatori meccanici al sonografo.....</i>	9
1.1.2 <i>La rivoluzione digitale.....</i>	12
1.1.3 <i>L'analisi per sintesi.....</i>	13
1.1.4 <i>Lo spazio timbrico.....</i>	18
1.1.5 <i>Mutamenti nella scrittura musicale.....</i>	20
1.2 La scrittura strumentale delle 'tecnoforme'	23
1.2.1 <i>La simulazione della reiniettività dell'eco a nastro.....</i>	23
1.2.2 <i>La modulazione ad anello.....</i>	24
1.2.3 <i>La modulazione di frequenza.....</i>	27
1.2.4 <i>Trasformazioni dello spettro.....</i>	30
CAPITOLO 2: LA COMPOSIZIONE ASSISTITA ALL'ELABORATORE.....	35
2.1 Le origini	35
2.1.1 <i>La fase pionieristica e la crisi degli anni Settanta.....</i>	35
2.1.2 <i>Dalla composizione algoritmica alla moderna CAO.....</i>	38
2.2 La CAO all'IRCAM	42
2.3 Il linguaggio LISP.....	43
2.3.1 <i>Il paradigma funzionale.....</i>	43
2.3.2 <i>La nascita di Common Lisp.....</i>	44
2.3.3 <i>Common Music.....</i>	46
2.3.4 <i>Da Formes a Patchwork.....</i>	47
2.4 Patchwork.....	48
2.4.1 <i>La patch: una nuova veste grafica per LISP.....</i>	48
2.4.2 <i>Il controllo di Csound in Patchwork.....</i>	54
2.4.3 <i>Il controllo di Chant in Patchwork.....</i>	58
2.4.4 <i>SP-Data.....</i>	61
CAPITOLO 3: FAUSTO ROMITELLI ALL'IRCAM.....	65
3.1 La partecipazione al <i>Cursus international de composition et d'informatique musicale</i> 65	65
3.1.1 <i>Verso la Francia.....</i>	65
3.1.2 <i>L'istituzione del Cursus.....</i>	66

3.2	<i>Natura morta con fiamme</i>	67
3.2.1	<i>Spazializzazione</i>	67
3.2.2	<i>Elettronica</i>	70
3.2.3	<i>Il programma WORKLISP</i>	70
3.2.4	<i>Patchwork?</i>	79
3.3	Nuove prospettive compositive	79
3.3.1	<i>Verso un'organizzazione del timbro</i>	79
3.3.2	<i>Una personale riflessione sul timbro. L'esperienza come compositeur en recherche</i> ..	81
3.4	La collaborazione con Laurent Pottier	87
CAPITOLO 4: <i>ENTRANCE</i>		89
4.1	Lo studio delle fonti	90
4.1.1	<i>Le fonti Ricordi</i>	91
4.1.2	<i>Le fonti IRCAM</i>	91
4.1.3	<i>Le fonti dell'archivio privato di Laurent Pottier</i>	94
4.2	Analisi dell'opera	94
4.2.1	<i>Il motivo ispiratore</i>	94
4.2.2	<i>Articolazione formale</i>	95
4.2.3	<i>Analisi della sezione 1B</i>	104
4.2.4	<i>La parte elettronica della sezione 1B</i>	112
4.2.5	<i>Il montaggio in Pro Tools</i>	125
4.2.6	<i>Simulazione del processo di montaggio</i>	126
4.3	L'esperimento di ricostruzione della parte 1B	130
CONCLUSIONI.....		139
APPENDICE A: NUOVE FONTI MUSICALI.....		143
A.1	Stato dell'arte	144
A.2	Le fonti di <i>EnTrance</i>	150
A.2.1	<i>Recensione</i>	150
A.2.2	<i>Accesso alle fonti originali</i>	152
A.2.3	<i>Interpretazione dei dati: ambienti software</i>	153
A.3	Nuova organologia	155
A.3.1	<i>Il sintetizzatore Yamaha SY99</i>	155
A.3.2	<i>Il campionatore SampleCell</i>	161
A.4	Le performance	164
APPENDICE B: FAUSTO ROMITELLI. <i>PERTINENCE DU TIMBRE</i>		169
APPENDICE C: CODICE LISP.....		189
C.1	Definizioni delle funzioni LISP in <i>lispdef</i>	189
C.2	<i>WORKLISP</i>	226
BIBLIOGRAFIA.....		235

INTRODUZIONE

Negli anni Ottanta e Novanta, la ricerca musicale all'IRCAM ha riservato un posto particolare alla composizione assistita all'elaboratore, in particolare allo sviluppo di sistemi informatici di rappresentazione e di elaborazione di dati in grado di assorbire, rileggere e potenziare l'esperienza dello spettralismo.

Una moltitudine di linguaggi di programmazione, sistemi di controllo e di sintesi del suono vennero sviluppati dal centro di ricerca parigino, per offrire ai compositori mezzi e ambienti software in grado di favorire la sperimentazione con la nuova organologia e ampliare i confini della scrittura musicale. Si apriva così l'importante stagione della musica composta con l'ausilio del personal computer, sostenuta dalla convinzione generale che gli sviluppi in ambito ingegneristico e la miniaturizzazione dei processori, con il passaggio dai grandi sistemi di elaborazione (basti pensare al computer *PDP-11*) a software di programmazione visuale sviluppati per piattaforme *Macintosh*, avrebbero prodotto un cambiamento di paradigma nel pensiero compositivo.

In quegli anni Fausto Romitelli, giovane compositore goriziano già allievo di Franco Donatoni, approda all'IRCAM, prima come studente ai *Cursus international de composition et d'informatique musicale* (1990-1991) e poi con una borsa di studio in qualità di *compositeur en recherche* (1993-1995). Nel centro parigino, Romitelli si confronta direttamente con il mondo della sintesi digitale del suono e del suo controllo informatico. *Natura morta con fiamme* (1991) per quartetto d'archi ed elettronica e *EnTrance* (1995) per soprano, ensemble ed elettronica sono gli esiti più significativi di questo periodo. Si tratta di opere dove è innegabile l'influenza del pensiero spettrale, in particolare dello spettralismo di Tristan Murail, che si configurava come possibile ponte tra lo spettralismo processuale di Gérard Grisey e lo spettralismo funzionale.¹

Nelle nuove forme di scrittura e di rappresentazione del suono, così come nella possibilità offerta dal dominio numerico di estrarre parametri dall'analisi del fenomeno

¹ Si veda ANGELO ORCALLI, "La pensée spectrale", in Nicolas Donin, Laurent Feneyrou (a cura di), *Théories de la composition musicale au XXe siècle. Volume 2*, Lione, Symetrie, 2013, pp. 1511-1573.

acustico, il compositore trova la possibilità di trattare gli spettri non solo nella loro dimensione qualitativa ma nel dato discreto, parametrico. Romitelli coniuga così regole combinatorie, in un chiaro debito verso l'insegnamento di Donatoni, con la ricerca sulle tecniche di distorsione dello spettro e, più in generale, con le tecniche spettrali sdoganate dai compositori del gruppo dell'Itineraire.

Il mio lavoro prende in esame questa fase della produzione romitelliana e in particolare *EnTrance*, opera della prima maturità, in cui Romitelli raggiunge una sintesi tra i modelli compositivi a bagaglio della sua formazione e un'approfondita ricerca sul timbro condotta nell'alveo teorico-ideologico che in quegli anni caratterizzava l'IRCAM. La tesi mostra come in *EnTrance* chiaro sia l'orientamento compositivo che Romitelli seguirà negli anni a venire: il compositore trova nell'idea della trance, nella ricerca su una sorta di regressione dello stato di coscienza, una sintesi del campo di tensione tra la processualità del suono inteso in senso griseyano e l'utensile informatico: trattare parametri discreti e categorizzare così il timbro in seno al discorso musicale.

La tesi si articola in quattro capitoli e tre appendici. Nel primo capitolo si individuano i fattori che hanno posto le basi della 'rivoluzione dei suoni complessi' di cui parla Tristan Murail e le loro ripercussioni sulla scrittura musicale degli anni Settanta e Ottanta, in particolare nella corrente della musica spettrale. Particolare risalto viene dato al trasferimento nella scrittura strumentale delle 'tecnoforme' derivate dalle prassi del laboratorio elettroacustico, con esempi tratti da composizioni di Murail (simulazione della reiniettività dell'eco a nastro in *Memoire/Erosion*, modulazione di frequenza in *Gondwana*) e di Gérard Grisey (modulazione ad anello e distorsione spettrale in *Jour, Contre-jour*).

Il secondo capitolo prende in esame l'evoluzione storica dei sistemi per la composizione musicale assistita all'elaboratore (CAO, *composition assistée par ordinateur*). Si traccia un percorso che dagli albori della *computer music* porta, all'inizio degli anni Novanta, allo sviluppo presso l'IRCAM di Parigi di *Patchwork*, un software per la CAO che, integrando i paradigmi di programmazione funzionale e visuale, permette la manipolazione grafica di oggetti. Saranno riportati inoltre elementi del linguaggio di programmazione LISP. Il percorso delineato si rivelerà funzionale ai casi di studio esposti nei capitoli successivi.

Il terzo capitolo presenta l'esperienza parigina di Fausto Romitelli nel lustro 1990-1995. Si prenderà in considerazione l'opera *Natura morta con fiamme* (1991), per quartetto d'archi ed elettronica, composta durante la frequentazione del *Cursus international de composition et d'informatique musicale* all'IRCAM. Saranno valutati aspetti inerenti la spazializzazione e la generazione dei suoni che vanno a costituire la parte elettronica. Si tratterà inoltre l'evoluzione del linguaggio compositivo dell'autore attraverso alcuni esempi tratti da

Mediterraneo I. Les idoles du soleil (1992), che permettono di far chiarezza sull'affinamento delle tecniche compositive che verranno poi impiegate anche nella scrittura di *EnTrance* (1995).

Nel quarto capitolo la ricerca si focalizza sull'opera *EnTrance* di Fausto Romitelli, per soprano, ensemble ed elettronica, composta all'IRCAM con l'assistenza musicale di Laurent Pottier. Nata nel solco dell'estetica spettrale, l'opera si configura come un rituale finalizzato ad uno stato di trance, in cui la strutturazione del timbro passa attraverso procedimenti al tempo stesso spettrali e combinatori. Sulla base dello studio delle fonti cartacee e informatiche, sarà presentata un'analisi formale dell'opera e, più nel dettaglio, verrà proposta la ricostruzione del processo creativo impiegato nella composizione di una sezione del brano. L'analisi permette inoltre di individuare probabili errori commessi in fase di montaggio nella parte elettronica; l'ipotesi, convalidata dal confronto diretto tra le due dimensioni dell'opera, dalle annotazioni presenti in partitura, così come dal testo redatto dal compositore per presentare il brano, suggerisce la proposta di un progetto editoriale volto, attraverso un processo di rigenerazione della parte elettronica, a risolvere i problemi di coerenza con la dimensione strumentale dell'opera.

In appendice, si indagheranno i problemi di preservazione e restituzione alla vita concertistica del caso di studio principale. *EnTrance* prevede infatti un'organologia 'allargata', in quanto oltre al soprano e agli strumenti acustici tradizionali, si compone di una parte elettronica creata in studio, un sintetizzatore e un campionatore. Attraverso lo studio delle fonti e una storia delle esecuzioni dell'opera, si suggeriranno strategie editoriali in grado di rispondere ai problemi relativi all'obsolescenza tecnologica dei diversi dispositivi elettronici.

Verrà inoltre presentato uno scritto inedito di Romitelli, redatto durante il suo soggiorno parigino, che fa luce sulla maturazione del suo linguaggio compositivo: l'ancoraggio dichiarato ai fondamenti della linguistica e della fonologia gli consente di definire un sistema di opposizioni in cui strutturare e categorizzare il timbro, argomento centrale della sua ricerca. Saranno infine riportati programmi in codice LISP sviluppati dal compositore durante la sua permanenza all'IRCAM.

CAPITOLO 1: LA RIVOLUZIONE DEI SUONI COMPLESSI

1.1 Il suono. La complessità svelata

1.1.1 *Dai risonatori meccanici al sonografo*

La principale rivoluzione che ha investito la musica del secondo novecento – affermava Tristan Murail nel 1980 – è la ‘rivoluzione dei suoni complessi’. Le sue cause non sono da ricercarsi in una ridiscussione della scrittura musicale ma, più profondamente, nella maggiore comprensione della natura del suono in sé, favorita dallo sviluppo di nuove tecniche di analisi.²

La ricerca sulle ‘qualità’ del suono, sul ‘timbro’, ha interessato gran parte del secolo scorso. Già nei primi decenni del Novecento numerosi studi hanno cercato di validare sperimentalmente l’opinione maggiormente diffusa secondo cui la qualità sonora degli strumenti era legata essenzialmente ad una forma d’onda periodica, e quindi rappresentabile in serie di Fourier.³

² «La révolution la plus brutale et la plus marquante qui ait affecté le monde musical dans les années récentes n’a pas pris sa source dans une quelconque remise en cause de l’écriture musicale (sérielle ou autre), mais bien plus profondément dans le monde des sons eux-mêmes, autrement dit dans l’univers sonore que le compositeur est invité à gérer. L’ouverture sans précédent du monde des sons que nous connaissons actuellement ne peut néanmoins manquer d’avoir des conséquences sur les techniques d’écriture elles-mêmes, pour tout compositeur soucieux d’assumer la place qui lui revient dans l’évolution de la musique». TRISTAN MURAIL, “La révolution des sons complexes”, in *Darmstädter Beiträge zur neuen Musik*, XVIII, Mainz, Schott, 1980, poi raccolto in TRISTAN MURAIL, *Modèles & Artifices*, textes réunis par Pierre Michel, Strasbourg, Presses universitaires de Strasbourg, 2004, p. II.

³ Principio già espresso nel 1877 da Helmholtz: «On inquiring to what external physical difference in the waves of sound the different qualities of tone correspond, we must remember that the amplitude of the vibration determines the force or loudness, and the period of vibration the pitch. Quality of tone can therefore depend upon neither of these. The only possible hypothesis, therefore, is that the quality of tone should depend upon the manner in which the motion is performed within the period of each single vibration. For the generation of a musical tone we have only required that the motion should be periodic, that is, that in any one single period of vibration exactly the same state should occur, in the same order of occurrence as it presents itself in any other single period». HERMANN L.F. HELMHOLTZ, *On the Sensations of Tone as a Physiological Basis for the Theory of Music*, traduzione in inglese di Alexander J. Ellis, London / New York, Longmans, Green, and Co., 1895, p. 19.

Tuttavia la teoria helmholtziana non era in grado di giustificare la capacità umana di riconoscere uno strumento acustico anche se registrato con distorsioni dello spettro che ne alterano la forma d'onda.⁴

L'avvento della tecnologia elettroacustica e magnetica consentirà di fare un passo ulteriore. Come è noto, avvalendosi delle tecniche di registrazione, su disco e poi su nastro, Pierre Schaeffer potrà indagare l'evoluzione temporale del suono e mettere in luce il contributo dell'inviluppo dinamico sulle forme e sulla percezione degli oggetti sonori. Alterando il normale utilizzo dei sistemi di riproduzione con le tecniche di rallentamento e riproduzione al contrario del suono, nel corso degli anni Cinquanta Schaeffer dimostrerà su basi sperimentali che:

- Il suono degli strumenti acustici possiede generalmente tre fasi distinte: un transitorio d'attacco, una fase di tenuta (assente nei suoni percussivi), e una fase di rilascio;
- esiste una diretta correlazione tra il transitorio d'attacco, l'evoluzione delle parziali e il timbro percepito: «Le timbre perçu est une synthèse des variations de contenu harmonique et de l'évolution dynamique; en particulier, il est donné dès l'attaque lorsque le reste du son découle directement de cette attaque».⁵ Questo aspetto è comprovato da una serie di esperimenti: a) la riproduzione al contrario di un suono di pianoforte ne altera definitivamente il timbro; b) il taglio su nastro del transitorio di attacco di un suono di campana, isolando quindi la fase di risonanza, presentava affinità con un suono di flauto.⁶

⁴ Si veda ad esempio l'esperimento percettivo condotto da Halson V. Eagleson e Oran W. Eagleson in "Identification of Musical Instruments When Heard Directly and over a Public-Address System", in «The Journal of the Acoustical Society of America», v. 19 n. 2, 1947, pp. 338-342. L'esperimento evidenzia che il tasso di riconoscimento di strumenti acustici tradizionali non varia in modo sostanziale tra l'ascolto diretto del suono acustico e l'ascolto attraverso un sistema di diffusione a bassa qualità. Naturalmente va considerato che i sistemi di diffusione in voga negli anni Quaranta erano soggetti a notevoli distorsioni del segnale e ad una risposta in frequenza estremamente limitata.

⁵ PIERRE SCHAEFFER, *Traité des objets musicaux*, Parigi, Editions du seuil, 1966, p. 231.

⁶ Risset e Wessel ricordano come già Stumpf, nel 1910, fece notare che l'eliminazione del transitorio d'attacco comportava delle difficoltà nel riconoscimento degli strumenti. JEAN-CLAUDE RISSET, DAVID L. WESSEL, "Exploration of Timbre by Analysis and Synthesis", in Diana Deutsch (a cura di), *Psychology of Music*, New York-Londra, Academic Press, 1982, p. 29. Inoltre va sottolineato che nel 1926 Stumpf, parlando del *Klangfarbe*, distingue caratteristiche 'interne' e caratteristiche 'esterne' che concorrono alla sua definizione: quelle interne si riferiscono alla 'vera' struttura spettrale del suono, dal numero di parziali e dal loro bilanciamento; quelle esterne, invece, coinvolgono aspetti temporali e dinamici (transitori di attacco e di rilascio), così come la presenza di componenti rumorose nell'attacco, modulazioni ed altre fluttuazioni. Si veda ALBRECHT SCHNEIDER, "Change and Continuity in Sound Analysis: A Review of Concepts in Regard to Musical Acoustics, Music Perception, and Transcription", in Rolf Bader (a cura di), *Sound – Perception – Performance*, Springer, 2013, p. 83.

La possibilità di intervenire direttamente sul materiale sonoro, attraverso operazioni di taglio, dilatazione temporale e inversione, «rendono palesi [...] le proprietà di dissimmetria e transitorietà insite nella maggior parte degli eventi sonori, proprietà considerate, fino ad allora, prerogativa esclusiva dei suoni percussivi». In altri termini, per Schaeffer «il suono musicale, per quanto raffinato, è fisicamente un rumore».⁷

Ancora nel 1954, altri studi dimostrano la consapevolezza dell'importanza dei transienti d'attacco. Richardson, analizzando la risposta di alcuni strumenti a fiato – nello specifico, le canne d'organo –, cerca di tracciare il comportamento delle parziali nel transitorio iniziale del suono:

Few wind instruments have transients which are exact copies of their steady states; either 'underblown tones' or 'reed partials' occurs in the first tenth of a second until overlaid by the principal tones of the column in its steady state. In spite of their evanescent nature, the view is now held that it is these transients which enable the listener to distinguish the sounds of different musical instruments or between two of the same class. [...] The transient is indeed part of the "formant" of an instrument, and ought to be exhibited as a characteristic alongside the steady-state spectrum.⁸

Un passaggio decisivo nell'analisi tempo variante dei segnali sonori è avvenuta con la realizzazione del sonografo che implementa un banco di filtri analogici e traccia graficamente la morfologia spettrale del suono. Questo strumento ha permesso di mettere in luce un aspetto fondamentale delle limitazioni di principio dell'osservazione dei fenomeni ondulatori e dell'analisi spettrale: il sistema è soggetto al ben noto principio di indeterminazione tempo/frequenza: la scelta di filtri con larghezza di banda stretta che permettono di estrarre informazioni precise sulle frequenze va a scapito del dettaglio della rappresentazione nel tempo. Viceversa, filtri a larghezza di banda maggiore restituiscono nel dettaglio le trasformazioni temporali ma sono imprecisi nel dominio delle frequenze.⁹

La tecnica sonografica divenne popolare negli anni Sessanta e Settanta: con particolare enfasi ne parla Emile Leipp per il quale il sonogramma:

⁷ ANGELO ORCALLI, *Fenomenologia della musica sperimentale*, Potenza, Sonus, 1993, p. 177.

⁸ E.G. RICHARDSON, "The Transient Tones of Wind Instruments", in «The Journal of the Acoustical Society of America», v. 26 n. 6, 1954, p. 962.

⁹ Questa declinazione del principio di indeterminazione tempo/frequenza è tuttora presente nella tecnica di analisi spettrale più in voga: la DFT (Discrete Fourier Transform).

Est une véritable partition musicale, comportant exactement les renseignements que l'on trouve sur la partition classique, avec cette différence toutefois qu'il est possible de mesurer avec précision la fréquence et la durée de chaque note, ainsi que son évolution dynamique.¹⁰

Le analisi condotte da Leipp confermarono la natura mutevole, 'organica' del suono: esso sembra possedere una nascita, un'evoluzione ed una fine. Tale osservazione comporta l'abbandono dell'idea di oggetto sonoro in quanto tale, a favore di una visione quasi biologica: il suono come 'essere vivente'.¹¹

La ricerca di Leipp costituisce un passo importante nello svelamento della complessità intrinseca al suono acustico. Tuttavia l'informazione sonografica non forniva elementi sufficienti per la ri-sintesi elettronica dei suoni analizzati. Per Jean-Claude Risset e David Wessel, questa difficoltà «points out the need to extract significant features from a complex physical structure. Also, one must be able to control through synthesis the aural relevance of the features extracted in the analysis. Only recently has this been possible».¹² Questa strada condusse allo sviluppo del metodo di analisi per sintesi.

1.1.2 La rivoluzione digitale

I primi tentativi di utilizzare il computer come strumento per la sintesi del suono risalgono alla metà degli anni '50, in seno alle ricerche condotte dai Laboratori Bell, New Jersey, sulla possibilità di trasmettere un segnale audio vocale in digitale. La banda frequenziale limitata sufficiente a codificare in modo percettivamente funzionale il segnale vocale e la spinta industriale che puntava agli enormi vantaggi commerciali che ne sarebbero derivati furono fattori determinanti per il successo del progetto, tali da permettere ai ricercatori Bell di intravedere ulteriori sviluppi: con l'incremento, già all'epoca esponenziale, della capacità di calcolo dei processori, da lì a poco sarebbe stato possibile sviluppare sistemi di conversione A/D in grado di codificare un segnale audio con una larghezza di banda sufficiente a trasmettere

¹⁰ ÉMILE LEIPP, *Acoustique et Musique, données physiques et technologiques, problèmes de l'audition des sons musicaux, principes de fonctionnement et signification acoustique des principaux archétypes d'instruments de musique, les musiques expérimentales, l'acoustique des salles*, Parigi, Masson, 1984, p. 87, citato in ANGELO ORCALLI, "La pensée spectrale", p. 1545.

¹¹ Si veda ANGELO ORCALLI, *Fenomenologia della musica sperimentale*, p. 180. La visione organica che contraddistingue la ricerca sul suono condotta da Leipp avrà importanti ricadute in ambito musicale, in particolare nell'estetica di Gérard Grisey. La sua composizione *Partiels* (1975), ad esempio, è fondata sull'analisi sonografica di un suono di trombone.

¹² JEAN-CLAUDE RISSET, DAVID L. WESSEL, "Exploration of Timbre by Analysis and Synthesis", p. 30.

anche un segnale musicale. È in questo clima entusiasmante e pionieristico che si inquadra il lavoro di Max Mathews, che fin dai primi esperimenti vide chiaramente anche le possibilità compositive offerte dai sistemi digitali: non solo la codifica del segnale musicale, ma anche e soprattutto lo sviluppo di sistemi in grado di creare ex novo il suono nel dominio numerico. Nel 1957 Mathews sviluppa il software *MUSIC I*, seguito subito nel 1958 da *MUSIC II*. Si trattava di sistemi molto semplici, limitati alla generazione di poche forme d'onda, difficilmente manipolabili (*MUSIC I* permetteva la sintesi di una singola forma d'onda triangolare, mentre *MUSIC II* consentiva la sovrapposizione di solo quattro segnali presi da un repertorio di 16 forme d'onda). Il passo però era compiuto: nasceva la sintesi digitale del suono. Gli sviluppi del software *MUSIC X* sono noti: da lì a poco sarebbe seguito *MUSIC III* (1960) che ampliava le potenzialità di controllo dei parametri di generazione, e nel 1962, grazie anche all'interesse del direttore del gruppo di ricerca dei Laboratori Bell, John Pierce, e alla collaborazione con la Princeton University garantita da Jim Randall, nacque *MUSIC IV*. Lo sviluppo del software poteva ritenersi completo: *MUSIC IV* fornirà il modello di base per tutte le successive versioni e per gli altri software/linguaggi di programmazione dedicati alla sintesi del suono, tra cui il più famoso resta ancora oggi *Csound*.

1.1.3 L'analisi per sintesi

Milton Babbitt, in un suo saggio del 1965, intravide le possibilità analitiche offerte dal computer in relazione allo studio dei fenomeni acustici. Fino a quel momento, la computer music si era occupata della sintesi di suoni 'nuovi', finalizzati sia alla composizione musicale sia agli studi sulla psicologia della percezione. La sensazione collettiva che grazie al computer si potesse virtualmente generare qualsiasi tipo di suono aveva infatti monopolizzato la ricerca del settore. Babbitt auspicava invece l'avvio di una ricerca sul suono e sulle sue qualità timbriche, in un approccio 'non-compositivo':

The direct computer production of sound [...] already has its own valuable literature.¹³
The noncompositional uses of this method, however, are not alluded to in this literature and bear listing. [...]

¹³ Babbitt si riferisce ai lavori condotti da Max Mathews e Joan E. Miller presso i Laboratori Bell, con lo sviluppo del software *Music IV*, e della sua applicazione nel settore della percezione musicale presso il Princeton University Computer Center. Si vedano: MAX MATHEWS, "An Acoustic Compiler for Music and Psychological Stimuli", in «The Bell System Technical Journal», v. 40, n. 3, 1961, pp. 667-694; JAMES TENNEY, "Sound-Generation by means of a Digital Computer", in «Journal of Music Theory», v. 7 n. 1, 1963, pp. 24-70; J. K. RANDALL, "A Report from Princeton", in «Perspectives of New Music», v. 3 n. 2, 1965, pp. 84-92.

The investigation of the dimension of so-called ‘timbre’, not in order to ‘duplicate’ timbres, but to test the results of duplicating their dimensionality, this involving analysis by synthesis, is most easily undertaken by the interrelation of the computer as instrument of sound analysis and production. [...]

The first decade of general computer use has but just ended; little of that decade of computer time has been applied to musical research or production, so there is as yet relatively little to show and only slightly more to tell, but the second decade can be only radically different; indeed, it already is.¹⁴

A metà degli anni '60, Mathews e Risset iniziavano lo studio delle tecniche di sintesi nel dominio digitale. Le prove condotte mediante la tecnica della sintesi per forma d'onda fissa non avevano portato a risultati convincenti. Fu necessario allora individuare nuovi sistemi di analisi che potessero fornire ulteriori dati utili per la sintesi. Il principio dell'analisi per sintesi è così riassumibile:

- La registrazione di un suono acustico viene analizzata al fine di fornire una descrizione fisica del comportamento del suono;
- i parametri così ottenuti sono impiegati per generare un suono tramite sintesi additiva¹⁵ a computer;
- il suono di sintesi viene confrontato percettivamente con il suono originale per valutarne la somiglianza. Nel caso in cui il modello risulti insufficiente, vengono svolte ulteriori analisi – oppure si interpretano diversamente i dati forniti dalle precedenti analisi – che conducono alla definizione di nuovi parametri.
- la modifica dei parametri è accompagnata ad ogni passo da una valutazione percettiva: si delinea in questo modo la rilevanza di un particolare parametro nella percezione del timbro;
- la ricerca si conclude quando il suono di sintesi non è più distinguibile dal suono originale.

¹⁴ MILTON BABBITT, “The Use of Computers in Musicological Research”, in «Perspectives of New Music», v. 3, n. 2, 1965, p. 82.

¹⁵ La sintesi additiva è una tecnica che affonda le sue radici in un'estensione in ambito acustico del teorema di Fourier secondo il quale ogni forma d'onda periodica può essere modellata come somma di segnali sinusoidali. La sintesi prevede l'impiego di una serie di oscillatori controllati in ampiezza e frequenza; i segnali generati dagli oscillatori sono quindi sommati tra loro per generare spettri complessi. Pur con un elevato potenziale, questa tecnica è però estremamente dispendiosa in termini computazionali in quanto necessita di un numero elevato di oscillatori (la sintesi di suoni complessi può richiedere diverse centinaia di parziali). Per un approfondimento si rimanda a CURTIS ROADS, *The Computer Music Tutorial*, Cambridge-Londra, The MIT Press, 1996, p. 134 e segg.

È possibile quindi descrivere il principio dell'analisi per sintesi con il seguente schema (figura 1):

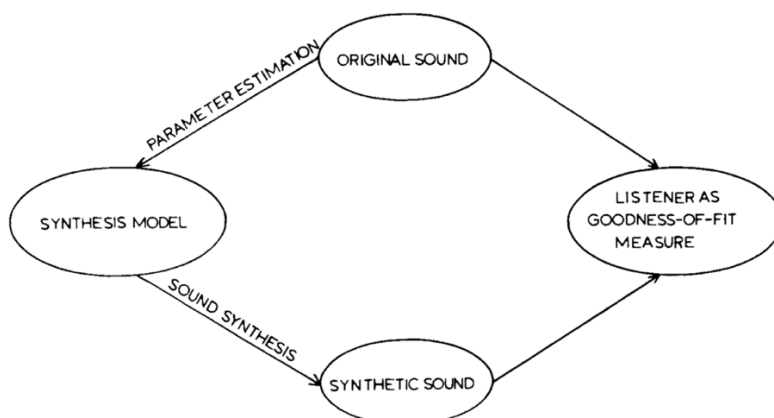


Figura 1 – Modello concettuale dell'analisi per sintesi.¹⁶

1.1.3.1 Analisi

L'analisi spettrografica del suono di tromba, registrato in una camera anecoica, suggeriva come, ad una data intensità, ogni strumento abbia una propria struttura formantica, che rimane invariata al mutare della frequenza fondamentale della nota suonata (figura 2).

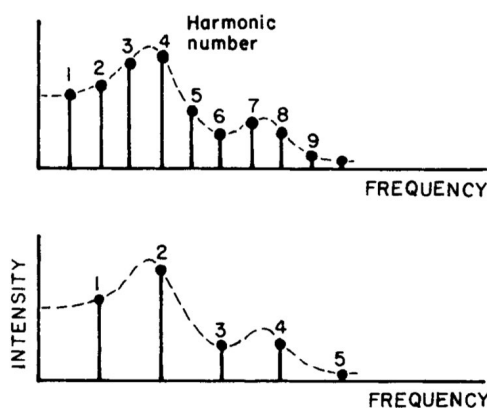


Figura 2 – Diagramma comparativo di due suoni suonati a distanza di un'ottava. Si noti l'invariabilità delle formanti spettrali.¹⁷

¹⁶ JEAN-CLAUDE RISSET, DAVID L. WESSEL, "Exploration of Timbre by Analysis and Synthesis", p. 44.

¹⁷ *Ivi*, p. 28.

Tuttavia lo spettrogramma, come descritto in precedenza, non era in grado di fornire dati sufficientemente precisi per comprendere l'evoluzione temporale delle singole componenti, soprattutto all'interno del transitorio d'attacco.

Grazie alla tecnologia digitale, i suoni vennero digitalizzati e quindi analizzati attraverso un programma informatico sviluppato nel 1961 da Mathews, Miller e David secondo *Pitch Synchronous Analysis*:¹⁸ questo tipo di analisi assume che il suono sia quasi periodico e, definendo un punto-ampiezza ad ogni periodo, fornisce una visualizzazione delle traiettorie in ampiezza di ciascuna parziale nel tempo (figura 3):

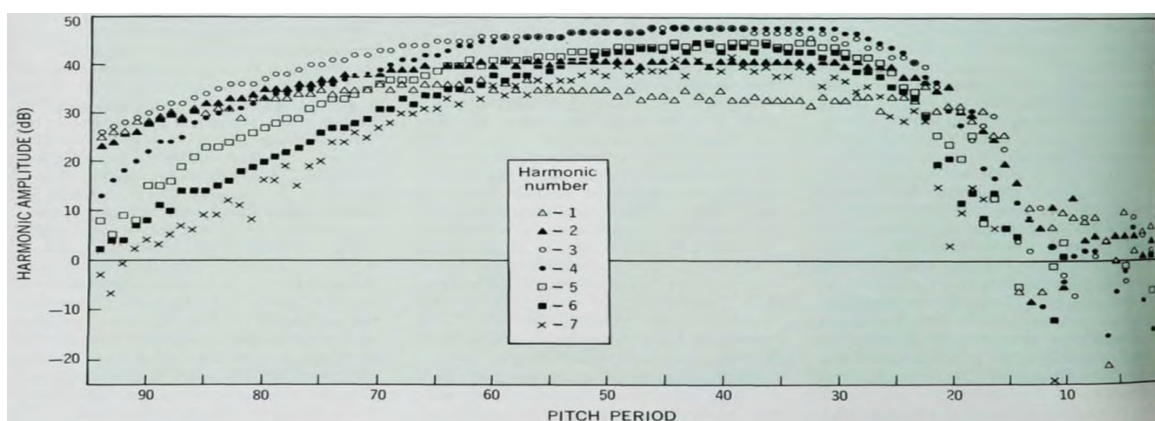


Figura 3 – Pitch Synchronous Analysis di un suono di tromba (Do#₅) della durata di 0,16 secondi (in ascisse, il tempo espresso in numero di periodi). Dettaglio delle prime 7 parziali. L'asse temporale è rovesciato, in quanto l'analisi forniva risultati più precisi sull'attacco del suono se riprodotto al contrario.¹⁹

Questo tipo di analisi permette inoltre di evidenziare alcune caratteristiche del suono ancora insondate:

The pitch-frequency display shows frequency glides at the beginning of the tone, periodic modulation at a rate of about 7 Hz (corresponding to vibrato) and quasi-random fluctuation at a much higher rate; the low-order harmonics build up faster than the high-order ones during the attack, [...] and similarly the low-order harmonics have a longer decay.²⁰

¹⁸ MAX MATHEWS, JOAN E. MILLER, E.E. DAVID JR, "Pitch Synchronous Analysis of Voiced Sounds", in «The Journal of the Acoustical Society of America», v. 33 n. 2, 1961, pp. 179-186.

¹⁹ JEAN-CLAUDE RISSET, MAX MATHEWS, "Analysis of Musical-Instruments Tones", in «Physics Today», v. 22 n. 2, 1969, p. 26.

²⁰ *Ivi*, p. 27.

1.1.3.2 Sintesi

Le funzioni che descrivono l'evoluzione delle parziali vengono quindi approssimate attraverso segmenti lineari (figura 4) e sono infine affidate al programma MUSIC IV, sviluppato da Max Mathews nei primi anni Sessanta.²¹

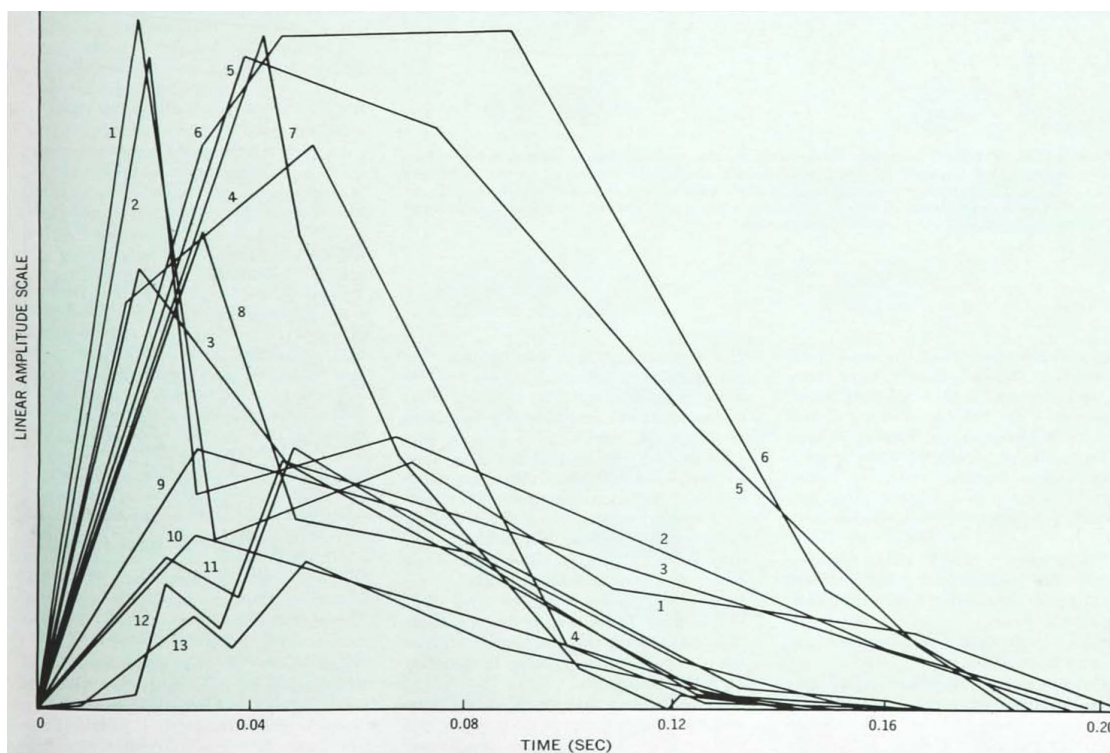


Figura 4 – Suono di tromba. Evoluzione nel tempo delle prime 13 parziali armoniche di un Re_4 della durata di 0,2 secondi.²²

La sintesi ha dimostrato la validità della *Pitch Synchronous Analysis*. Il suono generato con il modello raffigurato in figura 4 è risultato infatti molto simile al suono reale di tromba, tant'è che un test percettivo: «listeners were unable to distinguish which were the synthesized tones at better than the chance level of 50%. Highly trained musicians could do only slightly better (60%)».²³

Il progressivo affinamento della sintesi tramite variazione dei parametri ha permesso a Risset e Mathews di valutare la loro incidenza in rapporto alla percezione. È stato notato come ad esempio rapide fluttuazioni in ampiezza delle parziali fossero perlopiù irrilevanti.

²¹ Sul funzionamento del programma di sintesi sviluppato da Mathews, si rimanda a MAX MATHEWS, "The Digital Computer as a Musical Instrument", in «Science», v. 142 n. 3592, 1963, pp. 553-557.

²² JEAN-CLAUDE RISSET, MAX MATHEWS, "Analysis of Musical-Instruments Tones", p. 27.

²³ *Ibidem*

È stato notato altresì come altri parametri, già evidenziati nell'analisi, contribuiscano in modo sostanziale alla percezione del timbro. In particolare:

- La forma (formanti) dello spettro, che nella tromba manifesta un picco intorno ai 1500 Hz: «As the fundamental frequency is changed, the quality is kept more nearly constant if we use a spectrum with a formant structure rather than by simply keeping the same amplitude relations between the harmonics»;²⁴
- la struttura del transitorio d'attacco, della durata di circa 20 ms, che presenta un innesco delle prime parziali ben più rapido rispetto a quelle di ordine superiore;
- le fluttuazioni pseudo-aleatorie in frequenza (vibrato).

Questa scomposizione del suono, benché meno visivamente immediata rispetto a quanto può rappresentare uno spettrogramma, fu di fondamentale importanza: l'utilizzo del paradigma percettivo, se confrontato con il modello acustico della sorgente, garantiva da un lato la possibilità di ridurre la complessità di calcolo, dall'altro permetteva di focalizzare l'attenzione sui parametri più pregnanti che contribuiscono alla definizione del timbro. Grazie al metodo sperimentale dell'analisi per sintesi, infatti, la nozione di timbro mutava definitivamente. Ora descritto come evoluzione dinamica di relazioni interparametriche, il timbro può essere astratto dagli strumenti acustici tradizionali per essere studiato nelle sue possibili trasformazioni.

1.1.4 Lo spazio timbrico

È in questo contesto che prende forma il lavoro di John Grey al CCRMA di Stanford, in cui si propose di indagare le relazioni percettive tra i timbri di sedici diversi strumenti sulla stessa nota, $Mi\flat_3$. Analogamente a quanto fatto da Risset e Mathews, con l'aiuto di James Moorer gli stimoli furono generati a computer per sintesi additiva a partire dall'analisi del suono acustico. Grey non utilizzò la *Pitch Synchronous Analysis* bensì un sistema di filtri heterodina applicati nel dominio digitale (e quindi più precisi di quanto ottenibile analogicamente). Basandosi sulle osservazioni di Risset, anche in questo caso le funzioni che descrivono l'involuppo delle parziali sono state semplificate empiricamente – Grey parla

²⁴ *Ivi*, p. 28. L'analisi per sintesi rispondeva così definitivamente al dibattito che sul finire del XIX secolo aveva coinvolto Helmholtz e Hermann, ossia: per mantenere pressoché invariato il timbro di uno strumento nel momento in cui muta la frequenza fondamentale, lo spettro deve mantenere i rapporti di ampiezza tra le diverse parziali oppure è la posizione assoluta dei formanti a dover rimanere invariata? Si veda: FRITZ WINCKEL, *Music, Sound and Sensation: A Modern Exposition*, New York, Dover, p. 12 e segg.

di una riduzione del numero di segmenti da 300-500 a 4-8 – senza comportare sensibili differenze percettive (figura 5).

A partire da questi 16 suoni, un test condotto per via percettiva ha consentito di elaborare i dati ottenuti tramite matrici di similarità e quindi di disporre gli stimoli in uno spazio tridimensionale, rappresentato in figura 6. Grey suggerisce una possibile interpretazione delle tre dimensioni che compongono il grafico:

- Dimensione I (verticale): attiene la distribuzione dell'energia spettrale. Ad esempio, il corno (FH) presenta una banda spettrale ristretta con energia concentrata sulle basse frequenze; viceversa, lo spettro del trombone con sordina investe una larga banda di frequenze, e l'energia nelle prime parziali è ridotta;
- dimensione II (orizzontale): rappresenta la sincronicità delle parziali nei transitori di attacco e di rilascio. In questo caso, i legni ad ancia semplice si oppongono agli archi;
- dimensione III (profondità): considera l'energia presente nel transitorio di attacco e, più in generale, ne indica il grado di inarmonicità. Qui archi, clarinetti e sassofoni si oppongono agli ottoni e al fagotto.

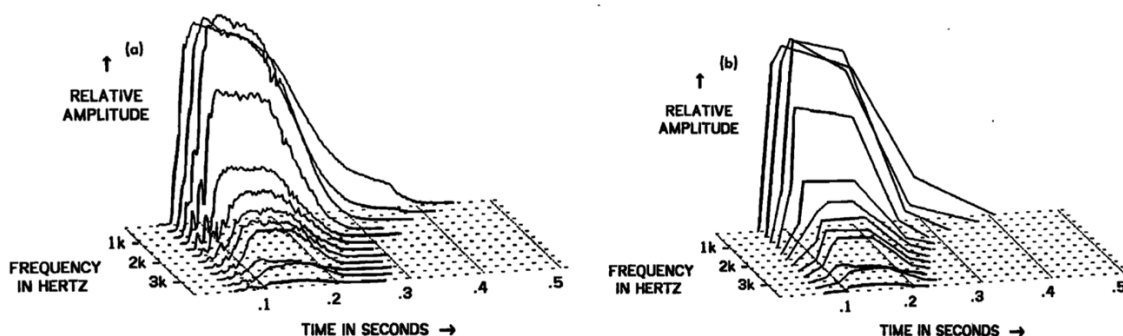


Figura 5 – Rappresentazione dell'involuppo delle parziali di un suono di tromba: a) come dato dall'analisi via eterodina; b) semplificato.²⁵

²⁵ JOHN GREY, "Timbre discrimination in musical patterns", in «Journal of the Acoustical Society of America», v. 62 n. 2, 1978, p. 468.

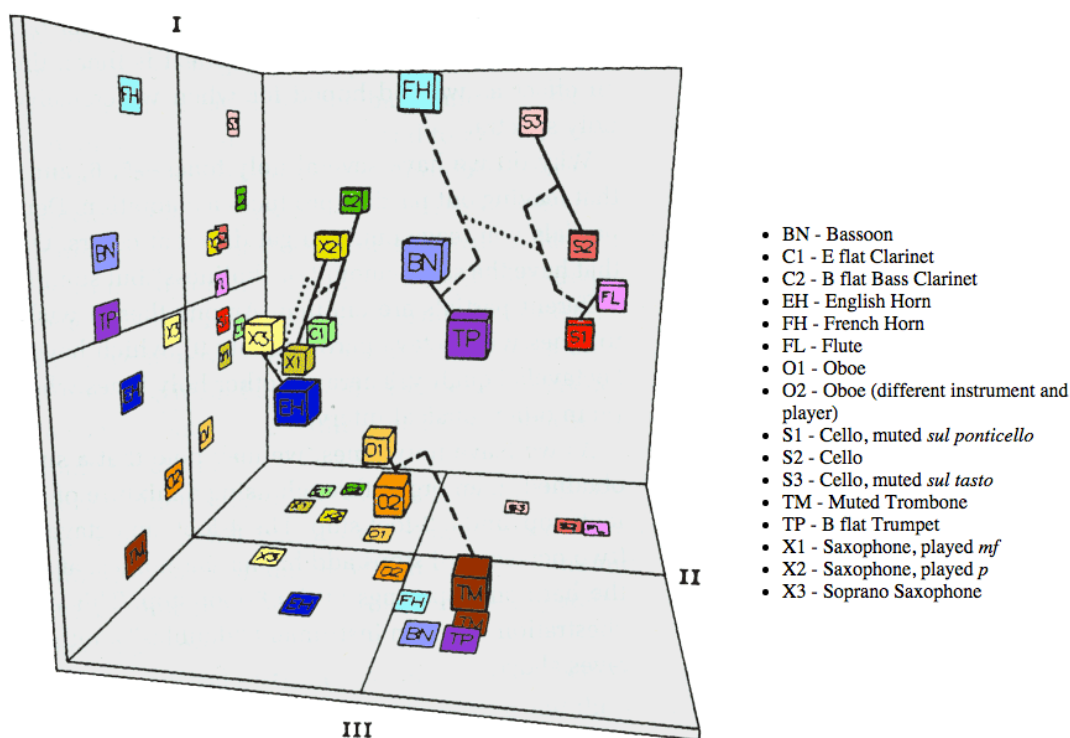


Figura 6 – Spazio timbrico di Grey.²⁶

Le linee in figura 6 che collegano i diversi strumenti sono ottenute dall'analisi con un algoritmo di *clustering* gerarchico.²⁷ Esse suggeriscono possibili percorsi timbrici:

Grey ha dimostrato, attraverso la sintesi computerizzata, che si possono tracciare per interpolazione delle traiettorie tra un timbro e l'altro, creando nuove immagini acustiche per transizione continua; dal suono del violoncello si giunge, ad esempio, a quello del corno attraverso una serie di espressive mutazioni timbriche.²⁸

1.1.5 Mutamenti nella scrittura musicale

Le ricadute sulla composizione musicale delle scoperte in campo acustico finora presentate sono rintracciabili in particolar modo nella musica del gruppo dell'*Itinéraire*, che a partire dagli anni Settanta maturò l'estetica della musica spettrale. Il gruppo, formato da Gérard Grisey, Tristan Murail, Hugues Dufourt e Michaël Levinas, pose l'accento sulla necessità

²⁶ JOHN GREY, "Scaling of musical timbre", in «Journal of the Acoustical Society of America», v. 61 n. 5, 1977, p. 1272.

²⁷ Si tratta di un algoritmo sviluppato nel 1967 da Stephen Johnson presso i Laboratori Bell. Si veda JOHN GREY, "Scaling of musical timbre", p. 1272.

²⁸ ANGELO ORCALLI, *Fenomenologia della musica sperimentale*, p. 188.

di individuare procedimenti di orchestrazione in grado di superare le specifiche qualità dei singoli strumenti a favore di nuove strutture timbriche. Nel 1991, Grisey dimostra come la comprensione del campo acustico svelato dalla ‘rivoluzione dei suoni complessi’, che l’autore declina nel concetto di indagine ‘microfonica’, abbia ad esempio trovato le sue manifestazioni nella tecnica della sintesi strumentale:

Depuis quelques années, l’électronique nous permet une écoute microphonique du son. L’intérieur même du son, ce qui était caché et occulté par plusieurs siècles de pratiques musicales essentiellement macrophoniques, est enfin livré à notre émerveillement. D’autre part, l’ordinateur nous permet d’aborder des champs de timbres inouïs jusqu’à ce jour et d’en analyser très finement la composition. L’appréhension de ce nouveau champ acoustique encore vierge a rafraîchi notre écoute et déterminé de nouvelles formes ; il est enfin devenu possible d’explorer l’intérieur d’un son en étirant sa durée et de voyager du macrophonique au microphonique à des vitesses variables.

Seules, la synthèse électronique et la synthèse instrumentale nous permettent d’aborder cette dimension nouvelle. Dans la synthèse instrumentale [...] c’est l’instrument qui exprime chaque composante du son et, à la différence de la synthèse électronique, ces composantes sont si complexes qu’elles constituent déjà une micro-synthèse.²⁹

Attraverso l’impiego di diverse tecniche quali ad esempio il controllo sull’inviluppo dei singoli suoni³⁰ o la transizione tra diverse ‘forme’ spettrali, Grisey trasferisce quindi le scoperte in campo acustico e psicoacustico nella scrittura orchestrale: «Retenons de ces multiples traitements que la source instrumentale disparaît au profit d’un timbre synthétique totalement inventé et non donné a priori par les instruments. Le timbre et la hauteur sont donc composés simultanément et l’instrumentation, au sens traditionnel, est lettre morte».³¹

²⁹ GERARD GRISEY, “Structuration des timbres dans la musique instrumentale”, in Jean-Baptiste Barrière (a cura di), *Le timbre: métaphore pour la composition*, Bourgois / IRCAM, 1991, pp. 352-385, poi ripreso in GERARD GRISEY, *Écrits, ou l’invention de la musique spectrale*, a cura di Guy Lelong, Parigi, Éditions MF, 2008, p. 89.

³⁰ Emblematico è l’utilizzo dell’inviluppo che Grisey fa ad esempio in *Jour, Contre-jour* (1979), per 13 musicisti, organo elettronico ed elettronica: analogamente a quanto già sperimentato da Schönberg in *Farben* (1909), Grisey è consapevole dell’importanza del transitorio d’attacco nel riconoscimento degli strumenti acustici; la quasi totalità dei suoni indicati in partitura è caratterizzata da un inviluppo a doppia forcilla che di fatto ne neutralizza il timbro. Questo principio microfonico è inoltre esteso alla macroforma, rendendo l’intera opera «sans début ni fin véritable» (GÉRARD GRISEY, *Écrits, ou l’invention de la musique spectrale*, p. 142).

³¹ GERARD GRISEY, “Structuration des timbres dans la musique instrumentale”, p. 90.

1.2 La scrittura strumentale delle 'tecnoforme'

Come suggerisce Murail, con lo sviluppo delle tecniche elettroacustiche si è verificata una fecondazione reciproca tra musica strumentale ed elettronica, in grado di fornire nuovi schemi, nuove forme, nuove idee per la composizione. Tra i presupposti su cui si reggeva la scrittura del gruppo dell'Itinéraire, un elemento fondamentale è costituito dal trasferimento delle pratiche sviluppatasi nel laboratorio elettroacustico alla scrittura orchestrale.

Processi derivati dalle operazioni condotte su nastro magnetico, simulazione di sistemi di sintesi non lineare, trasformazioni spettrali, costituiscono infatti una costante nella musica spettrale degli anni Settanta e Ottanta.

1.2.1 La simulazione della reiniettività dell'eco a nastro

Per la composizione di *Mémoire/Erosion* (1976), per corno e nove strumenti, Murail si ispira ad una nota tecnica derivata una prassi laboratoriale dell'audio analogico: la reiniettività dell'eco a nastro. Il nastro magnetico scorre tra due magnetofoni, separati da una precisa distanza. Il segnale, registrato dal primo magnetofono, viene letto dal secondo. L'output viene quindi inviato nuovamente al primo magnetofono, che lo fissa nuovamente nel nastro assieme ad eventuali altri suoni. Si tratta di un processo reiniettivo nel quale si può osservare una graduale e costante erosione del suono:

Le son, à force d'être recopié et surtout d'être sans cesse remixé avec les nouveaux sons qui arrivent, s'use, se dégrade, se transforme, se détruit. Le son se mélange de bruit blanc, des fréquences finissent par émerger, des rythmes par se créer, des parasites apparaissent.³⁴

La procedura può esser letta in termini di entropia, che Murail considera positiva nel senso di un passaggio progressivo che porta dall'ordine al disordine: «L'univers entier est soumis à sa loi ; l'érosion naturelle est une de ses manifestations, qui détruit les structure géologiques, jusqu'à créer le désordre, le désordre maximum étant l'indifférenciation».³⁵

Questo procedimento consente a Murail di integrare in modo 'naturale' il rumore nella scrittura strumentale, in un processo continuo di trasformazione a partire dal suono 'puro' del corno. Esemplicando, la prima sezione di *Mémoire/Érosion* è generata a partire da un suono di corno intonato sul Do₄ (figura 8).

³⁴ TRISTAN MURAIL, "La révolution des sons complexes", in TRISTAN MURAIL, *Modèles & Artifices*, p. 16.

³⁵ Ivi, p. 17.



Figura 8 – Struttura armonica della prima parte di Mémoire/Érosion.³⁶

Gli archi passano gradualmente da una posizione ordinaria fino a una sollecitazione delle corde prossima al ponticello, liberando così le parziali superiori (passo *a*); in particolare, viene ‘reinnestata’ nel sistema la terza parziale (Sol₃). Analogamente al fenomeno del *wow and flutter*³⁷ nella tecnologia magnetica, il Do subisce quindi una flessione (Do – Do_↓ – Do, passi *b* e *c*). I nuovi suoni, reinnestati, producono altri armonici che vanno a destabilizzare definitivamente l’input iniziale (passi *d*, *e*, *f*). Quest’ultimo infine scompare a favore del Sib₄ (*g*). Da qui in avanti, le componenti si auto-alimentano fino al punto di saturazione (*i*); il Sib prende via via sempre più forza (un fenomeno che Murail ha sovente osservato nell’eco a nastro analogico – passi *g*, *b*, *i*, *j*); infine, le componenti si riducono fino ad evidenziare la settima e la nona parziale del Sib (*k*).

1.2.2 La modulazione ad anello

Sviluppato originariamente per l’industria delle telecomunicazioni, a partire dai primi anni Cinquanta il modulatore ad anello (RM) fu introdotto come sistema di elaborazione del segnale audio nei principali studi di musica elettronica (NWDR di Colonia, 1954; NHK di Tokio, 1955; Studio di fonologia della RAI di Milano, 1956). La sua semplicità ne favorì

³⁶ TRISTAN MURAIL, “La révolution des sons complexes”, p. 21.

³⁷ Indica il grado di imprecisione dei dispositivi meccanici preposti al trascinamento del supporto magnetico, che si traduce in una variazione della frequenza del segnale registrato.

l'immediata diffusione, tanto che già nel 1961 circa la metà dei maggiori studi se ne era dotata.³⁸

Il funzionamento del modulatore ad anello è semplice. Nel caso più elementare, quello di un oscillatore sinusoidale, il RM va a modulare l'ampiezza della frequenza portante f_p operando una moltiplicazione tra i segnali. La modulazione, affidata ad un secondo oscillatore, segue anch'essa un andamento sinusoidale. Quando la frequenza della modulante f_m è inferiore ai 20 Hz, si parla di *low frequency oscillator* (LFO), e il risultato della modulazione è percepito come un effetto di tremolo. Se f_m è invece superiore a 20 Hz, la variazione è sufficientemente rapida per generare due bande laterali e, allo stesso tempo, sopprimere la frequenza portante. Il suono risultante sarà quindi costituito dalla somma e dalla differenza delle frequenze f_p e f_m . Ad esempio, se la portante è impostata a 900 Hz e la modulante a 500 Hz, otterremo due frequenze laterali: una a 400 Hz (900 – 500) e una a 1400 Hz (900 + 500). Se i segnali in ingresso non sono sinusoidali ma complessi, la moltiplicazione tra i segnali produrrà un risultato estremamente ricco, costituito da somme e differenze di ciascuna parziale portante con ciascuna parziale modulante.

Applicazioni in musica del modulatore ad anello sono innumerevoli: basti citare ad esempio il caso di *Mixtur* di Karlheinz Stockhausen in cui l'orchestra, amplificata dal vivo, viene modulata in tempo reale tramite RM;³⁹ oppure, *Mutations* di Jean-Claude Risset in cui la modulazione è applicata nel dominio digitale;⁴⁰ o ancora, *Jour, Contre-jour* di Gérard Grisey, in cui l'organo elettrico è trattato dal vivo con RM.

Il modulatore ad anello trova la sua controparte acustica nel fenomeno dei suoni di combinazione, scoperti da Tartini e poi sistematizzati da Helmholtz a metà del XIX secolo. Questo principio ha ispirato la scrittura di Gérard Grisey, che ne farà uso diffusamente nella sua produzione della seconda metà degli anni Settanta. Grisey afferma che attorno ad ogni intervallo, o ad ogni complesso di suoni, esiste un'aura di suoni differenziali e addizionali la cui complessità dipende dalla ricchezza spettrale di quelli che definisce suoni *generatori*. Usando una metafora luminosa, dichiara che «les sons ont une ombre»⁴¹ costituita dall'insieme dei suoni di combinazione:

³⁸ Si veda HUGH DAVIES, "A Simple Ring-modulator", in «Musics», n. 6, 1976, pp. 3-5.

³⁹ Si veda KARLHEINZ STOCKHAUSEN, "Notes on Mixtur (1964)", in «Electronic Music Review», n. 1, 1967, pp. 18-21.

⁴⁰ Si veda CHARLES DODGE, THOMAS JERSE, *Computer Music. Synthesis, Composition, and Performance*, Schirmer, 1997, pp. 108-114.

⁴¹ GERARD GRISEY, "Structuration des timbres dans la musique instrumentale", p. 103.

Le spectre d'un son étant couramment nommé timbre (en allemand, *Klangfarbe*, couleur du son), je propose de nommer l'ensemble des sons différentiels et additionnels produit tant à l'intérieur du spectre d'un son isolé qu'entre deux sons entendus simultanément : l'ombre du son (en allemand, *Klangschatten*).⁴²

In *Jour, Contre-jour* (1979), la metafora luminosa viene declinata nel rapporto luce / ombra = suoni generatori / suoni di combinazione.⁴³ A partire da uno spettro, Grisey articola la microforma selezionando di volta in volta due frequenze base che diverranno suoni generatori. Una volta presentati, essi generano 'l'ombra del suono' composta da suoni differenziali e addizionali. Associando i due suoni generatori alle frequenze portante e modulante del RM, possiamo descrivere il sistema impiegato per calcolare le frequenze dei suoni di combinazione in questi termini:

- Suoni differenziali: $(f_p - f_m)$, $(2f_m - f_p)$, $(3f_m - 2f_p)$, $(2f_p - 2f_m)$, $(3f_p - 3f_m)$;
- suoni addizionali: $(f_p + f_m)$, $(2f_p + f_m)$, $(f_p + 2f_m)$, $(3f_p + 2f_m)$.

Le frequenze così ottenute sono quindi convertite in notazione tradizionale e affidate in ultima battuta all'ensemble strumentale.

Un ultimo aspetto da sottolineare è come il rapporto tra frequenza portante e modulante, 'suoni generatori' per Grisey, possa generare spettri armonici, 'sans ombre' (se il rapporto è intero) o, nel caso contrario, distorti:

Certains intervalles sont sans ombres parce que les sons résultants ne font que renforcer la lumière de leurs harmoniques. D'autres au contraire produisent un réseau infiniment complexe de sons résultants donc les fréquences sont fort éloignées de celles contenues dans les sons générateurs et leurs harmoniques. Cette analogie avec l'ombre portée est d'ailleurs à l'origine de *Jour, Contre-jour*.⁴⁴

⁴² *Ibidem*

⁴³ Si veda LUCA COSSETTINI, ANGELO ORCALLI, "Da *Sortie vers la lumière du jour* a *Jour, Contre-jour*. Riscrittura di un mito dell'antico Egitto", in LUCA COSSETTINI (a cura di), *Scritture e ri-mediazioni / Écritures et re-productions*, Lucca, LIM, 2013, pp.271-308.

⁴⁴ GERARD GRISEY, "Structuration des timbres dans la musique instrumentale", p. 103.

1.2.3 La modulazione di frequenza

Come per il modulatore ad anello, anche la modulazione di frequenza è stata sviluppata nel settore delle telecomunicazioni, in particolare per la trasmissione radio. Ma per la sua applicazione nell'elaborazione dei segnali audio si attese fino al 1973, grazie al contributo di John Chowning. Dopo essersi formato presso i Laboratori Bell insieme a Max Mathews, e aver contribuito allo sviluppo del software *Music IV*, Chowning si trasferì a Stanford, dove qualche anno più tardi contribuì a fondare il CCRMA (Center for Computer Research in Music and Acoustics). Qui, teorizzò e implementò nel dominio digitale la sintesi FM. La conquista di Chowning è considerata una rivoluzione nel campo della sintesi audio digitale, in quanto consentiva di ottenere suoni estremamente complessi a partire da soli due operatori semplici (oscillatori).

La frequenza portante f_p viene modulata in frequenza da un'altra funzione sinusoidale f_m . Se la frequenza di f_m è inferiore ai 20 Hz, l'output produrrà un effetto di vibrato. Analogamente a quanto visto per il RM, con frequenze superiori alla soglia percettiva di 20 Hz si vanno a costruire bande laterali composte da somma e differenza tra f_p e f_m . La frequenza portante, in questo caso, non viene soppressa. La ricchezza della sintesi FM è legata ad un altro parametro: l'ampiezza della modulazione (indice di modulazione, i). Al crescere di i , compaiono ulteriori bande laterali che vanno a redistribuire l'energia spettrale (figura 9).

Riassumendo, l'output della sintesi FM si compone principalmente di: f_p , $f_p \pm n f_m$, dove n dipende dall'indice di modulazione i .

Le applicazioni musicali della modulazione di frequenza nel dominio dell'audio digitale hanno costellato il panorama della musica elettronica dalla metà degli anni Settanta fino ai giorni nostri. Si consideri solo ad esempio che uno dei sintetizzatori più diffusi negli anni Ottanta, il *DX7* della *Yamaha*, si fondava su sintesi FM. Sul piano compositivo, nella computer music occorre almeno citare *Stria* (1972-1977), composta dallo stesso John Chowning, in cui il viene fatto un utilizzo esclusivo della sintesi FM. Chowning ripartisce il materiale in uno spazio frequenziale non più fondato sull'intervallo di ottava bensì sul rapporto della sezione aurea ($\sim 1,618$).⁴⁵ Forte delle possibilità offerte dalla sintesi FM, il sistema di modulazione è stato anch'esso progettato per lavorare sul rapporto aureo: $f_p / f_m = \varphi$. In realtà, per *Stria* sono stati usati due segnali modulanti, il secondo dei quali presenta piccole differenze in frequenza rispetto al primo, finalizzate alla produzione di battimenti.⁴⁶

⁴⁵ Per un'analisi dettagliata di *Stria* si rimanda a LAURA ZATTRA, *Analyse de Stria de John Chowning*, reperibile all'indirizzo <<http://brahms.ircam.fr/analyses/Stria/>>, ultima visita 5 marzo 2016.

⁴⁶ CHARLES DODGE, THOMAS JERSE, *Computer Music. Synthesis, Composition, and Performance*, p. 138.

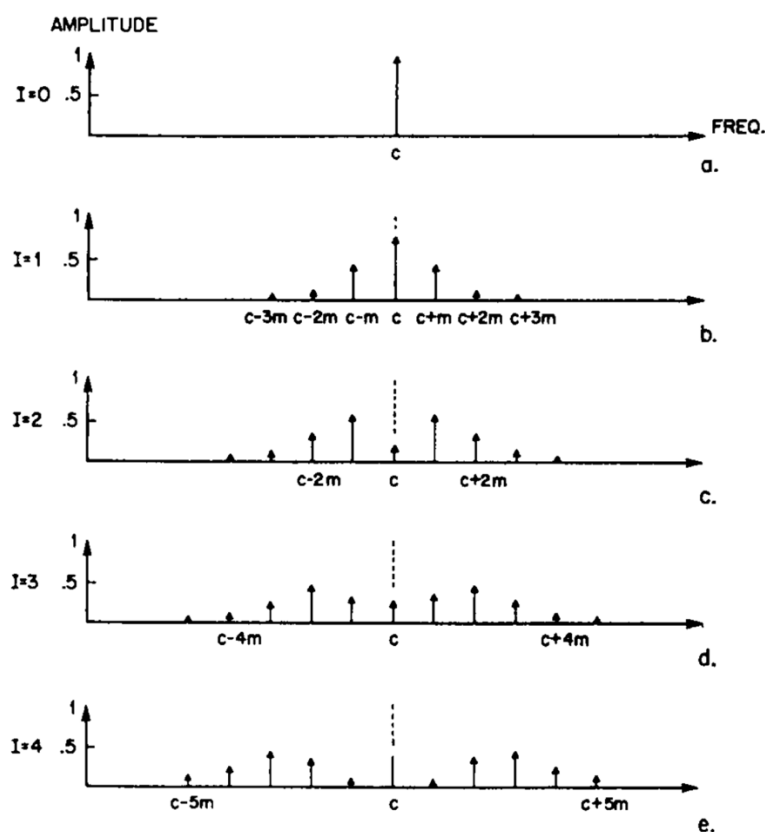


Figura 9 – Insorgenza delle bande laterali al variare progressivo dell'indice di modulazione i . c è la frequenza portante, m la modulante.⁴⁷

Il trasferimento della sintesi non lineare FM alla scrittura orchestrale è stato impiegato in modo strutturale da Tristan Murail per la composizione di *Gondwana* (1980). Il calcolo delle frequenze è qui semplificato, in quanto esiste una relazione 1:1 tra l'indice di modulazione i e n :

- per $i = 1$: (f_p) ; $(f_p \pm f_m)$
- per $i = 4$: (f_p) ; $(f_p \pm f_m)$; $(f_p \pm 2f_m)$; $(f_p \pm 3f_m)$; $(f_p \pm 4f_m)$.

La figura 10 esemplifica l'applicazione di questa tecnica. Ad ogni passo è possibile notare l'incremento dell'indice di modulazione, così come della frequenza modulante (passo di 4,87 Hz). Al contrario, la frequenza portante rimane ancorata al Do_{\sharp}^3 :

⁴⁷ JOHN CHOWNING, "The Synthesis of Complex Audio Spectra by Means of Frequency Modulation", in «Journal of the Audio Engineering Society», v. 21, n. 7, 1973, p. 527.



Figura 10 – Spettri risultanti dalla modulazione di frequenza.⁴⁸

I parametri della sintesi FM sono espressi in tabella 1:

Spettri	a	a'	b	b'	c	c'	d	d'	e-e'	f	g
f_p (Hz)	269,3	269,3	269,3	269,3	269,3	269,3	269,3	269,3	269,3	269,3	269,3
f_m (Hz)	7,88	12,75	17,92	22,49	27,36	32,23	37,1	41,97	46,84	51,71	56,58
i	2	4	6	8	10	11	12	13	14	15	16

Tabella 1 – Parametri della sintesi FM impiegati in Gondwana.

Va notato come a partire dallo spettro c' si verifica il fenomeno del 'ripiegamento': l'aumento dell'indice di modulazione produce infatti frequenze negative, ma tali frequenze sono percepite come positive; ciò che differenzia ad esempio -100 Hz e +100 Hz è la fase, che nelle frequenze negative viene ruotata di 180°.

⁴⁸ TRISTAN MURAIL, "Question de cible", in TRISTAN MURAIL, *Modèles & Artifices*, p. 71.

Murail non calcola l'ampiezza relativa delle parziali con precisione. Più semplicemente, all'incremento dell'indice di modulazione – e quindi al numero di suoni affidati all'orchestra – fa corrispondere una dinamica più contenuta. Questo motivo porta a definire la tecnica compositiva come 'simulazione' della sintesi FM.

1.2.4 Trasformazioni dello spettro

L'impiego della tecnica della sintesi strumentale ha indotto i compositori dell'itinéraire a immaginare possibili trasformazioni dello spettro armonico. L'idea è stata suggerita anche dalla constatazione che l'analisi del timbro di molti strumenti non era direttamente riconducibile ad uno spettro armonico teorico. La liuteria, spesso in modo empirico, ha da sempre lavorato dimensioni e forma degli strumenti per modellare il timbro grazie all'esaltazione o al filtraggio di alcune frequenze. Caso emblematico è quello del clarinetto, il cui lo spettro presenta solo le componenti dispari e può essere quindi considerato difettivo. La selezione di specifiche parziali, operazione associabile al filtraggio multibanda, è stata impiegata ad esempio da Gérard Grisey in *Jour, Contre-jour* o ancora in *Modulations* (1978) (figura 11).

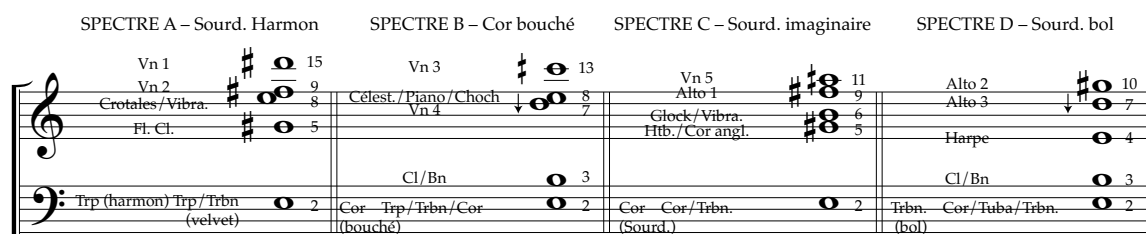


Figura 11 – Spettri difettivi utilizzati da Grisey per *Modulations*.⁴⁹

Inoltre, le caratteristiche fisiche e materiali di molti strumenti producono moti di vibrazioni che si traducono in spettri non armonici. Si pensi ad esempio alle percussioni intonate o anche alla naturale 'inarmonicità' del suono del pianoforte, attribuibile all'elevata tensione delle corde (figura 12):

⁴⁹ Trascrizione e rielaborazione di un esempio tratto da GÉRARD GRISEY, "Structuration des timbres dans la musique instrumentale", p. III.

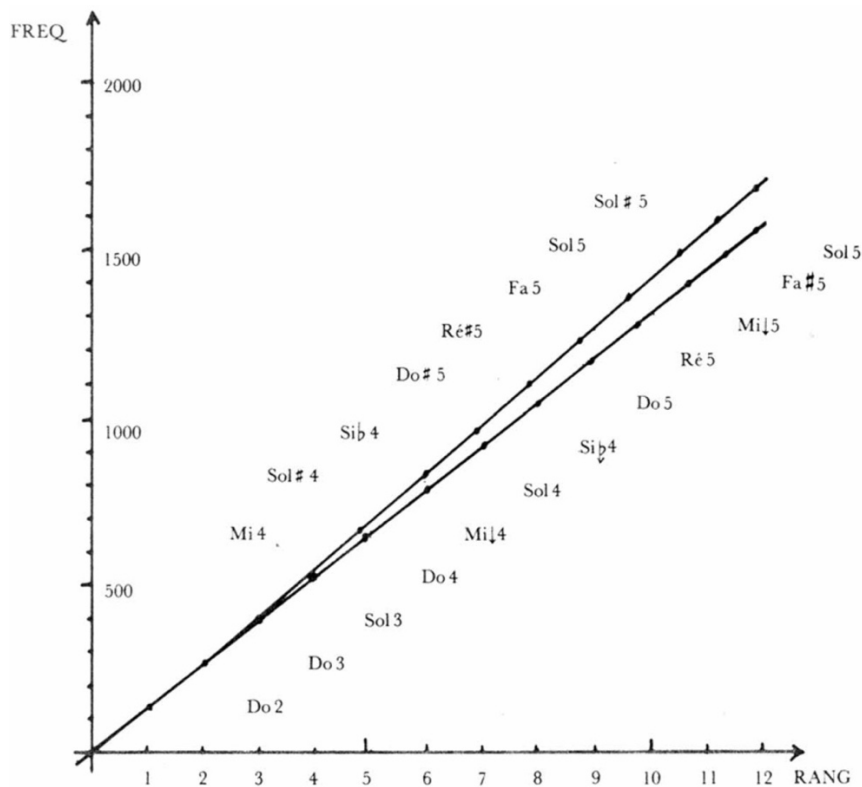


Figura 12 – Rappresentazione della distorsione spettrale di un suono di pianoforte.⁵⁰

Con le nuove scoperte della fisica acustica, che trovava anch'essa proprio nelle tecniche di analisi informatica del suono una possibilità di sviluppo impensata, le modulazioni che producevano spettri distorti come la sintesi FM o il RM non erano più solo semplici astrazioni teoriche, impiegabili per sviluppare nuovi suoni e nuovi pensieri compositivi, ma trovavano la loro controparte nella sfera dei fenomeni naturali.

Jour, Contre-jour di Gérard Grisey offre un chiaro esempio del trasferimento di tecniche di distorsione spettrale alla composizione orchestrale. Qui, l'altezza delle singole note viene derivata a partire dallo spettro armonico del *Sol* a 49Hz, le cui frequenze vengono poi approssimate al sesto e al quarto di tono, per andare incontro alle necessità della scrittura notata. Lo spettro armonico viene utilizzato una sola volta nell'opera, nella sezione centrale, tra l'altro in forma difettiva con le sole parziali dispari. Per le altre sezioni, Grisey sottopone lo spettro armonico a due diverse distorsioni che caratterizzeranno la prima e la seconda parte dell'opera. Il procedimento utilizzato per calcolare la distorsione è esclusivamente intervallare. Per la prima distorsione si procede come segue:

- Parziale 1 (fondamentale): invariata;

⁵⁰ Tristan Murail, "Spectre et lutins", in TRISTAN MURAIL, *Modèles & Artifices*, p. 39.

- parziali da 2 a 10: intervalli di 2m, 2M, 3m, 3M, 4, 5dim, 5, 6m, 6M; (aumento progressivo di un semitono);
- parziali da 11 a 12: intervalli di 6M+, 7m; (aumento progressivo di un quarto di tono);
- parziali da 13 a 24: intervalli di 6M+, 6M, 6m+, 6m, 5+, 5, 5dim+, 5 dim, 4+, 4, 3M+, 3M (diminuzione progressiva di un quarto di tono).

Per la seconda:

- Parziale 1 (fondamentale): invariata;
- parziali da 2 a 10: intervalli di 2m, 2M, 3m, 3M, 4, 5dim, 5, 6m, 6M; (aumento progressivo di un semitono);
- parziali da 11 a 20: intervalli di 6m+, 6m, 5+, 5, 5dim+, 5 dim, 4+, 4, 3M+, 3M (diminuzione progressiva di un quarto di tono).

La figura 13 mostra le curve di distorsione risultanti.

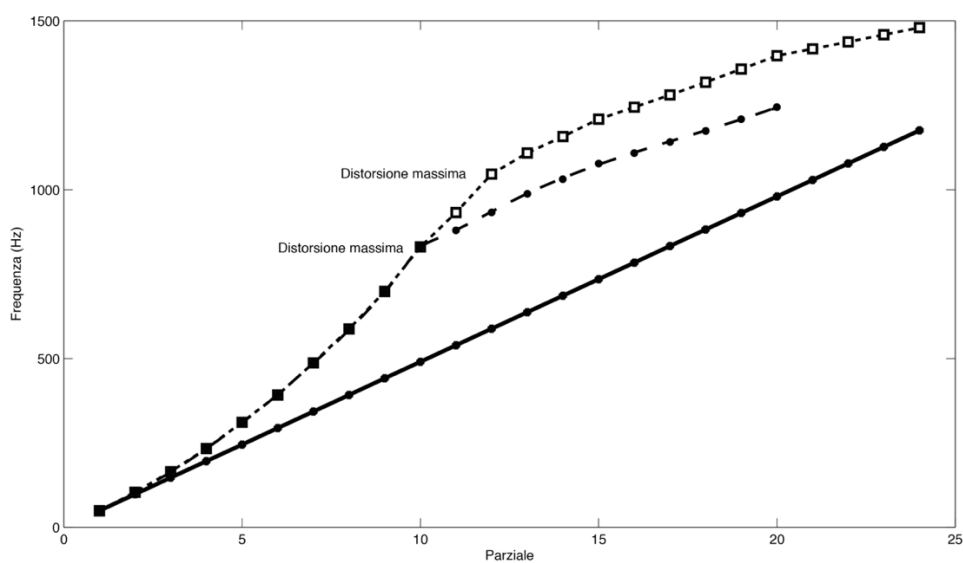


Figura 13 – Curve di distorsione degli spettri generatori (spettro armonico in linea continua; prima distorsione in linea tratteggiata breve e seconda distorsione in linea tratteggiata lunga).

Il secondo spettro distorto viene poi sottoposto a un'ulteriore elaborazione: una riflessione sull'asse definita dal $Sol\#_4$ a 415,3 Hz. Si crea così uno spettro di sub-parziali inarmoniche. In questo modo, Grisey definisce a priori il materiale armonico di base su cui costruisce l'intera opera (figura 14).



Figura 14 – Spettri in Jour, Contre-jour riportati in notazione tradizionale (approssimazione all'ottavo di tono). Per poter confrontare adeguatamente gli spettri distorti con quello armonico, quest'ultimo è stato circoscritto alle prime 24 parziali.

Le tecniche di distorsione spettrale, in Grisey ancora basate su procedimenti intervallari, verranno poi formalizzate da Tristan Murail in termini di funzioni matematiche (figura 15).

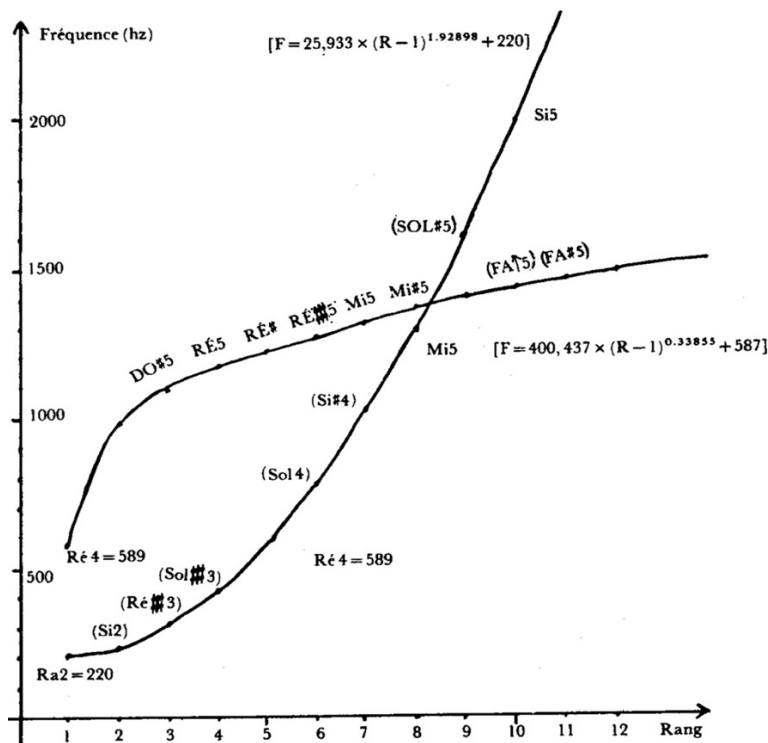


Figura 15 – Esempi di possibili distorsioni di uno spettro armonico.⁵¹

⁵¹ TRISTAN MURAIL, "Spectre et lutins", p. 40.

La distanza concettuale è marcata: ciò che per Grisey è processo di divenire del suono, per Murail diviene processo di organizzazione e trasformazione delle componenti frequenziali. Rinunciando a semplici rappresentazioni 'sonografiche' del fenomeno acustico nella sua evoluzione, Murail teorizza e mette in pratica un'elaborazione di tipo algoritmico del contenuto spettrale. Questo processo di astrazione lo porta ad allontanarsi nettamente dal primo spettralismo per approdare a una concezione della composizione da lui stesso definita 'armonia frequenziale'.⁵²

La teoria musicale aveva così posto tutte le basi necessarie per accogliere a pieno titolo i sistemi automatici di calcolo nell'atto compositivo:

Établir des relations entre des éléments revient à établir des *fonctions*, au sens mathématique. En principe, il suffirait de décrire la structure des durées et celle des composantes primaires pour *tout* décrire. C'est d'ailleurs à peu près ce que l'on fait lorsqu'on travaille sur l'ordinateur avec les programmes de synthèse classiques.⁵³

A partire dagli anni '80, grazie alle nuove tecnologie digitali l'elaborazione numerica dei dati provenienti dalle analisi sonografiche non solo ha offerto una 'visualizzazione' più accurata delle componenti microscopiche dei fenomeni acustici, ma è diventata a sua volta sorgente di possibili modelli per la creazione di oggetti sonori; l'oggetto sonoro così virtualizzato è divenuto fonte di nuove forme di organizzazione del materiale. All'IRCAM di Parigi un gruppo di ricercatori residenti, forti delle nuove possibilità offerte dal mondo digitale, assimila la lezione di Murail e ne implementa i fondamenti in ambito informatico, approdando nei primi anni Novanta alla realizzazione di *Patchwork*, un ambiente software dedicato alla composizione assistita all'elaboratore.

⁵² «Through this approach to harmony, it is possible to create harmonies (or timbres) that are completely invented, through analogies to the spectra found in nature. Most of my pieces, in fact, are built on structures that are not direct spectral observations: this is what I call 'frequencial harmony'. These harmonies are conceived outside the domain of equal temperament, equal-tempered quarter- or eighth-tones and form an unlimited harmonic realm, which happens to be contiguous to timbral space, thus placing us in a domain where harmony and timbre are more or less the same thing». TRISTAN MURAIL, "After-thoughts", in «Contemporary Music Review», v. 24, n. 2/3, 2005, p. 272.

⁵³ TRISTAN MURAIL, "Spectres et Lutins", p. 34.

CAPITOLO 2:

LA COMPOSIZIONE ASSISTITA ALL'ELABORATORE

2.1 Le origini

2.1.1 La fase pionieristica e la crisi degli anni Settanta

Le origini della composizione assistita all'elaboratore (CAO, in francese *composition assistée par ordinateur*) si possono scorgere già a metà degli anni Cinquanta, quando Richard C. Pinkerton⁵⁴ suggerisce l'idea del *Banal Tunemaker*, un algoritmo⁵⁵ fondato su un principio statistico e finalizzato alla composizione di una semplice melodia per l'infanzia: partendo dall'analisi di 39 melodie preesistenti, tutte trasposte in tonalità di Do maggiore, Pinkerton ottiene le probabilità di occorrenza di ogni singola nota e di ciascuna transizione tra le note; l'algoritmo andrà poi a comporre nota per nota la nuova melodia.

L'idea fu immediatamente ripresa e rielaborata da John F. Sowa con lo sviluppo della sua *Machine to Compose Music*:⁵⁶ il meccanismo, pur simile – ma ben più complesso – a quanto già proposto da Pinkerton, è affidato per la prima volta ad un computer. Di fatto, questo esperimento ha aperto la strada al concetto stesso di composizione assistita, ove il computer ha la funzione di svolgere calcoli dispendiosi in poco tempo.

Nel 1957 Lejaren Hiller scrive quella che possiamo considerare la prima vera composizione musicale che sfrutta il computer: *Illiad Suite*, per quartetto d'archi. Fondata sul principio che vede la creazione musicale basarsi sulla ricerca dell'ordine a partire da una moltitudine caotica di possibilità, sfrutta un sistema di regole in grado di accettare o rifiutare l'output di un generatore casuale di altezze e ritmi. Vi è possibile inoltre riscontrare l'utilizzo del calcolo delle probabilità e delle catene di Markov. Il brano fu interamente composto a computer per poi esser trascritto in notazione tradizionale per quartetto d'archi. Il lavoro di Hiller, svolto in collaborazione con Leonard Isaacson presso l'University of Illinois, può esser considerato il primo esempio di computer music fondata su processi stocastici.

⁵⁴ RICHARD C. PINKERTON, "Information Theory and Melody", in «Scientific American», v. 194/2, 1956, pp. 77-87.

⁵⁵ In informatica, un algoritmo è una serie di operazioni logiche e algebriche, espresse in linguaggio comprensibile al computer, la cui sequenza costituisce un programma.

⁵⁶ Si veda JOEL E. COHEN, "Information Theory and Music", in «Behavioral Science», v. 7/2, 1962, pp. 137-163.

Di lì alla fine degli anni Sessanta, si susseguirono diverse sperimentazioni, in generale basate su un metodo ben definito: le specifiche della composizione musicale sono definite a priori sotto forma di combinazioni simboliche non più controllabili una volta avviato il processo. È il periodo d'oro della composizione algoritmica.

In questo contesto merita sicuramente un cenno la figura di Iannis Xenakis, per il quale le possibilità computazionali costituivano la naturale estensione dei propri principi compositivi: «the naturalness with which the music of Xenakis can make use of computer comes from the fact that the language of his ideas corresponds to the language of the machines».⁵⁷ Va precisato che per Xenakis – che tra l'altro ebbe modo di conoscere pionieri della musica informatica quali Max Mathews e lo stesso Hiller – il computer era uno strumento in grado di risolvere velocemente compiti complessi e di fornire il materiale base per la composizione: «in essence, what the computer was outputting was not the composition itself but material with which Xenakis could compose. In contrast, the work of Hiller and Isaacson attempted to simulate the compositional process itself entirely, completely delegating creative decisions to the computer».⁵⁸ È lo stesso Xenakis a delineare la propria visione quando, parlando del rapporto tra macchina e creazione artistica, sintetizza:

- 1) The creative thought of man gives birth to mental mechanisms, which, in the last analysis, are merely sets of constraints and choices. This process takes place in all realms of thought, including the arts.
- 2) Some of these mechanisms can be expressed in mathematical terms.
- 3) Some of them are physically realizable: the wheel, motors, bombs, digital computers, analogue computers, etc.
- 4) Certain mental mechanisms may correspond to certain mechanisms of nature.
- 5) Certain mechanizable aspects of artistic creation may be simulated by certain physical mechanisms or machines which exist or may be created.
- 6) It happens that computers can be useful in certain ways.⁵⁹

I calcoli complessi messi in atto ad esempio per la composizione di *Metastaseis* (1955) possono quindi essere delegati alla macchina. Non è un caso che, se la composizione di

⁵⁷ GÉRARD ASSAYAG, “Computer Assisted Composition today”, in «Proceedings of 1st Symposium on Music and Computers», Corfu, 1998, disponibile online all'indirizzo <recherche.ircam.fr/equipement/repmus/RMPapers/Corfou98/>, ultima visita 5 marzo 2017.

⁵⁸ JOHN A. MAURER, “A Brief History of Algorithmic Composition”, <<https://ccrma.stanford.edu/~blackrse/algorithm.html>>, ultima visita 5 marzo 2017.

⁵⁹ IANNIS XENAKIS, *Formalized Music*, Pendragon, 1992, pp. 132-133.

Metastaseis è durata un paio d'anni, nel solo 1962 Xenakis componga *ST/4*, *ST/10* e *ST/48*, insieme ad altri lavori quali ad esempio *Stratégie* e *Polla ta Dhina*: come sottolinea Anastasia Georgaki, «the computer's ability to calculate his ideas faster was a propelling force for many of his works». ⁶⁰ *ST/4*, *ST/10* e *ST/48* sono opere composte con l'ausilio di un computer IBM 7090 capace di 500000 operazioni al secondo. La velocità di questa macchina impressionò allora Xenakis: «the long laborious calculation made by hand is reduced to nothing. The speed of the machine such as the IBM-7090 is tremendous»⁶¹. Il compositore, liberato dalle operazioni manuali, può concentrarsi sui problemi musicali.

Nei primi anni Sessanta Hiller e Robert Baker sviluppano *MUSICOMP*, (*Music Simulator Interpreter for Compositional Procedures*), forse il primo sistema informatico pensato come strumento versatile e variegato per la composizione con il computer. *MUSICOMP* non fu scritto come un unico programma, con un unico set di istruzioni, ma come una libreria di *subroutines* tra loro diversificate. L'idea di creare semplici funzioni compositive, 'impacchettate' nelle *subroutines*, e di assemblarle in un unico sistema garantì a *MUSICOMP* i caratteri di flessibilità e di generalità: il compositore, partendo da un unico sistema, poteva scegliere e utilizzare le funzioni che più aderivano alle proprie necessità e alla propria estetica, un approccio che, come si vedrà, costituirà la base dei sistemi per la CAO sviluppati negli anni '90. L'esito principale dell'utilizzo di questa tecnologia fu *Computer Cantata*,⁶² opera realizzata nel 1963 dagli stessi Hiller e Baker con intenti dimostrativi – che peraltro giustificano l'estrema varietà di tecniche compositive adoperate. Per questo lavoro, gli autori hanno fatto un ulteriore passo in avanti: *MUSICOMP*, scritto per l'IBM 7090, è stato interfacciato con un altro computer, il CSX-1, una workstation progettata per la sintesi sonora,⁶³ che in quegli anni iniziava a imporsi sulla scena musicale grazie soprattutto alle ricerche applicate condotte dai Laboratori Bell.⁶⁴ Per la prima volta, il programma gestisce sia la composizione della 'partitura' sia la generazione di suoni di sintesi.

⁶⁰ ANASTASIA GEORGAKI, "The grain of Xenakis technological thought on the computer music research of our days", *Proceedings of the International Symposium Iannis Xenakis*, University of Athens, 2005. <<http://cicm.mshparisnord.org/ColloqueXenakis/papers/Georgaki.pdf>>, ultima visita 5 marzo 2017.

⁶¹ IANNIS XENAKIS, *Formalized Music*, Pendragon, 1992, pp. 144.

⁶² Si veda LEJAREN A. HILLER, ROBERT A. BAKER, "Computer Cantata: A Study in Compositional Method", in «Perspectives of New Music», v. 3/1, 1964, pp. 62-90.

⁶³ Il computer *CSX-1* è stato progettato da J. L. Divilbiss dell'University of Illinois Coordinated Science Laboratory. La connessione tra i due computer era tutt'altro che immediata: le carte perforate con l'output dell'*IBM 7090* dovevano esser convertite in nastro perforato leggibile dal *CSX-1*.

⁶⁴ Si veda *infra*, par. 1.1.2.

Tuttavia, sarà proprio lo sviluppo della sintesi digitale del suono a costituire il fattore chiave della crisi che ha investito la CAO negli anni Settanta e nei primi anni Ottanta: molti compositori infatti, attratti dalle nuove possibilità timbriche e dal feedback acustico consentito dalla sintesi digitale, vi si dedicarono in modo esclusivo (si consolida in questo periodo la figura del compositore-ingegnere del suono). Ma, come ricorda Assayag, vanno menzionate alcune eccezioni:

Jean-Claude Risset and John Chowning for instance have maintained the tradition of the composer-researcher and have known how to gain profit from their original scientific results in order to work out a musical language integrating sound synthesis without any artistic compromise.⁶⁵

2.1.2 Dalla composizione algoritmica alla moderna CAO

Fu proprio Risset a intuire un possibile rilancio dell'utilizzo dell'informatica nel processo compositivo, in primo luogo evidenziando i limiti estetici e musicali dell'approccio puramente algoritmico, basato esclusivamente sulla formalizzazione delle idee musicali, e della complessità propria del processo compositivo. Risset propone quindi un utilizzo nuovo dell'informatica in musica:

À chaque niveau, le musicien pourra garder ou non les commandes, laisser se déployer l'efflorescence d'une écriture démultipliée ou intervenir «à la main» pour contrôler ou agencer détails ou grandes lignes. [...] il faut les développer, les diversifier, les rendre plus souples, plus commodes, plus musicales - mais non les figer dans un système final «idéel» : le musicien doit intervenir pour construire son propre monde.⁶⁶

È un passaggio concettuale determinante, che mira al superamento dei limiti della composizione algoritmica, al riscatto del 'genio' del compositore minacciato dal determinismo della macchina. Perché «ciò che costituisce l'originalità e il prezzo inestimabile del pensiero umano è il potere di non pertinenza, l'imprevedibilità vera e non più l'imprevedibilità domata all'interno di un sistema probabilistico. Alla proposizione

⁶⁵ GÉRARD ASSAYAG, "Computer Assisted Composition today".

⁶⁶ JEAN-CLAUDE RISSET, "Musique, calcul secret?", in «Critique», n. 359, Paris, 1977, pp. 423-424, citato da MIKHAIL MALT, "La composition assistée par ordinateur", <<http://musinfo.fr/archives/anciensite/texte/Malt-I.pdf>>.

binaria ‘di due cose l’una’, è il potere di rispondere ‘la terza’.⁶⁷ La fase di modellizzazione, o di formalizzazione, è quindi solo una delle tappe del processo compositivo. Scelte arbitrarie, impiego di modelli non calcolabili rientrano con vigore nella composizione di un’opera, aprendo ad un gioco di compromessi tra «*prévisible, la règle, le déterminé et le sensible, l’intuition ou la partie inconsciente du métier*».⁶⁸ Il computer può così essere utilizzato come strumento e non più come protagonista assoluto del processo compositivo.

Si afferma quindi la distinzione tra composizione algoritmica, in cui per definizione l’atto creativo è circoscritto allo sviluppo dell’algoritmo, e composizione assistita all’elaboratore, in cui l’algoritmo entra nella pratica compositiva ma non ne costituisce la totalità, perché l’idea artistica sembra nascere in una sorta di dialogo nell’interazione uomo-macchina. Quella che di primo acchito sembra una risposta convincente all’obiezione sulla rinuncia dell’io che segue la definizione del meccanismo, algoritmico o seriale che sia,⁶⁹ in realtà nasconde una rivoluzione da molti non colta, ma puntualizzata con chiarezza da Lucien Sfez che nella sua *Critique de la communication* svela il profondo paradosso: «si recupera un ‘essere’ del vero all’interno di un sistema che lo esclude, per garantire la sua credibilità».⁷⁰ Esistono infatti alcuni importanti presupposti che bisogna implicitamente condividere se si accetta l’idea che l’interattività uomo-macchina possa realizzarsi in maniera feconda nella sfera della creatività:

- 1) Per far entrare il computer intelligente nel mondo degli uomini, bisogna assegnargli un elemento indispensabile: la condivisione della creatività.

⁶⁷ ROBERT ESCARPIT, *Théorie générale de l’information et de la communication*, Parigi, Hachette, 1976, trad. it. *Teoria dell’informazione e della comunicazione*, Roma, Editori Riuniti, 1979, p. 53.

⁶⁸ MIKHAIL MALT, “La composition assistée par ordinateur”.

⁶⁹ Le radici della rinuncia del sé compositore non nascono con la composizione algoritmica, ma sono rintracciabili fin dallo sviluppo delle teorie del serialismo integrale. Si veda LUCA COSSETTINI, ANGELO ORCALLI, *L’invenzione della fonologia musicale. Saggi sulla musica sperimentale di Luciano Berio e Bruno Maderna*, Lucca LIM, 2015, p. 11 e seguenti. Già nel 1960 Krenek sosteneva che «if the succession of tones is determined by serial regulation [...] and, in addition to this, the timing of the entrance into the musical process of these tones is also predetermined by serial calculation [...], it is no longer possible to decide freely (that is, by “inspiration”) which tones should sound simultaneously at any given point. In other words, the so called harmonic aspect of the piece will be entirely the result of operations performed on premises that have nothing to do with concepts of ‘harmony’, be it on the assumption of tonality or atonality or anything else. Whatever happens at any given point is a product of the preconceived serial organization, but by the same token it is a chance occurrence because it is as such not anticipated by the mind that invented the mechanism and set it in motion» (ERNST KRENEK, “Extents and Limits of Serial Techniques”, in «The Musical Quarterly», v. 46, n. 2,», p. 228). Krenek ricorre all’analogia dell’orologiaio che costruisce un meccanismo di cui non sa predire che cosa produrrà: la libertà del compositore, in contrapposizione alla cosiddetta libertà dell’ispirazione, sta per lui nella scelta del meccanismo. Si veda, LUCA COSSETTINI, *Opere chiuse in sistemi aperti. Autopoiesi nella musica elettronica*, Lucca, LIM, 2013, p. 87.

⁷⁰ LUCIEN SFEZ, *Critica della comunicazione*, Torino, Hopefulmonster, 1995, p. 338.

- 2) Si fa così riferimento ad una definizione di *Homo "creans"*, e non solo di *Homo faber*. Si riconosce – implicitamente – che l'attività fabbricante è soltanto strumentale, esteriore, meccanicista. Si accetta bene o male la definizione di un "dentro" autonomo, di un'interiorità immanente, non suscettibile di essere analizzata in termini di perseguimento di fini prefissati. Le si concede il potere di definire la sua opera man mano, nel corso del TEMPO, per ritocchi successivi, senza che la sua finalità sia chiaramente prescritta.
- 3) Si definisce parallelamente il dialogo – la zona di operazioni alternative – come logica della creazione: in altri termini, si ammette che non esiste alcuna creazione vera e propria al di fuori della comunità linguistica, nello scambio. Meglio ancora: si fa di questa zona di dialogo la natura vera e propria della comunicazione.
- 4) Ci si avvicina così ad una analisi in termini di semiotica sociale e si accetta di integrare le teorie del soggetto nel rapporto tra uomo e macchina, non più un rapporto in cui ci si guarda in cagnesco, l'uno di fronte all'altro, ma in cui c'è un'interpenetrazione reciproca e mutuale: in questo modo, ci si allinea ad una teoria della comunicazione in termini di *Lebenswelt*, di prospettive di vita condivisa, o d'intenzionalità alla Searle, che dir si voglia.⁷¹

Già nel 1978 Barry Truax aveva intuito il problema. Nel suo articolo "Computer Music Composition: The Polyphonic POD System",⁷² suddivide la composizione musicale con il computer in tre ambiti, tra loro non facilmente separabili, impostisi per convenzione e quindi definiti arbitrariamente: sintesi del suono, composizione musicale e interazione uomo-macchina. Nel quadro teorico delineato da Truax, la CAO si configurerebbe pertanto come un processo che unisce a feedback i tre momenti (figura 16).

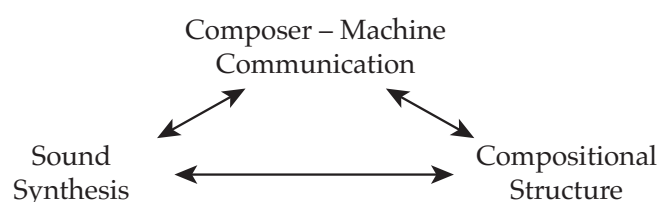


Figura 16 – Suddivisione del processo di composizione musicale con il computer secondo Truax.

Qualche anno dopo, nel 1981, Otto Laske elabora una personale visione in cui è possibile rappresentare l'interazione compositore-programma come una traiettoria che dal controllo

⁷¹ Ivi, p. 337.

⁷² Si veda BARRY TRUAX, "Computer Music Composition: The Polyphonic POD System", in «IEEE Computer», v. II n. 8, 1978, pp. 40-49.

puramente 'manuale' porta alla composizione algoritmica. In questa traiettoria, la composizione assistita all'elaboratore si colloca al centro (figura 17), una zona di grande interesse in quanto consente una flessibilità di approcci: «the powers of intuition and machine computation may be combined».⁷³

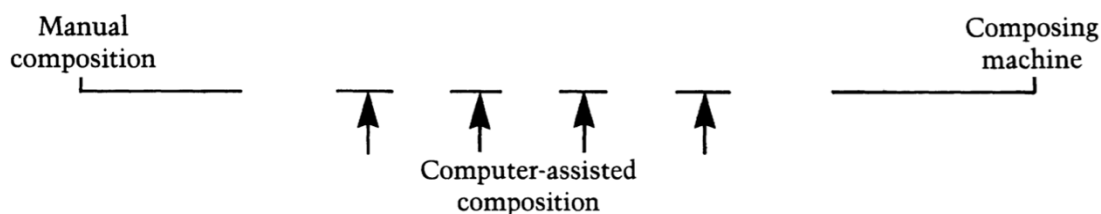


Figura 17 – «Trajectory showing degree of composer control over compositional process. Arrows show alternative approaches to computer-aided composition.»⁷⁴

L'avvento dell'informatica in musica ha avuto importanti ricadute anche sul metalinguaggio musicale, in particolare in materia di formalizzazione teorica del pensiero compositivo: se tradizionalmente i compositori si esprimono attraverso il linguaggio naturale, sovente caratterizzato da analogie e metafore, il computer necessita di un linguaggio preciso, privo di ambiguità. Si viene a creare così uno scarto tra le pratiche formaliste conosciute già dalla prima metà del XX secolo e le necessità formali imposte dall'utilizzo degli strumenti informatici per la composizione assistita: «La grande nouveauté dans cette deuxième moitié du XXe siècle, est que l'utilisation de l'ordinateur en musique ne consiste pas en la formalisation de la théorie, mais en la formalisation de la pratique, domaine traditionnel de l'expérience personnelle».⁷⁵ Inevitabilmente, questi nuovi sistemi influiscono sul pensiero compositivo, condizionandone spesso gli esiti. In altri termini, il computer porta il compositore ad immaginare strutture musicali inedite e, più in generale, a ripensare il processo compositivo. Molti importanti centri internazionali di ricerca su musica e nuove tecnologie diventano allora luoghi di aggregazione di compositori e scienziati, vere e proprie fucine della sperimentazione musicale.⁷⁶

⁷³ OTTO LASKE, "Composition Theory in Koenig's Project One and Project Two", in «Computer Music Journal», v. 5, n. 4, pp. 54 - 65, MIT-Press, 1981, p. 54.

⁷⁴ OTTO LASKE, "Composition Theory in Koenig's Project One and Project Two", p. 55.

⁷⁵ MIKHAIL MALT, "La composition assistée par ordinateur".

⁷⁶ Tra gli altri, è importante ricordare i già citati Laboratori Bell, le università di Princeton e Stanford, il Computer System Research Group dell'Università di Toronto, l'Experimentalstudio der Heirich-Strobel-Stiftung des SWF, l'Electronic Music Studio della EMS, il CSC di Padova. Si veda LAURA ZATTRA, *Studiare la computer music*, Padova, Libreria Universitaria, 2011.

2.2 La CAO all'IRCAM

È sui diversi aspetti dell'utilizzo del computer nella produzione musicale che all'IRCAM⁷⁷ di Parigi, nei primi anni Ottanta, un gruppo di ricercatori – Gérard Assayag, Pierre-François Baisnée, Pierre Cointe, Xavier Rodet – e di compositori – Jean-Baptiste Barrière, Marc-André Dalbavie, Magnus Lindberg, Claudy Malherbe, Philippe Manoury, Tristan Murail, André Riotte, Jean-Claude Risset, Kaija Saariaho⁷⁸ – avviarono una riflessione. Già nel 1984 Tristan Murail, nello scritto *Spectre et lutins*, dopo aver descritto alcuni dei principi compositivi da lui utilizzati e i pregi del computer nella gestione del materiale musicale, suggerisce lo sviluppo di un sistema di aiuto alla composizione che sia progettato ad alto livello:

Si l'on veut généraliser au niveau de la grande forme certaines idées précédemment exposées, les systèmes d'interaction vont devenir trop complexes pour être appréhendés d'un seul coup par l'esprit. Il faut donc automatiser à niveau plus élevé, créer un système d'aide à la composition (CAO = composition assistée par ordinateur). L'usage de l'ordinateur ainsi décrit diffère sensiblement des usages habituels : synthèse du son, ou au contraire utilisation automatique d'algorithmes de composition. Il s'agit au contraire de bâtir un système interactif, convivial, semblable à ce qui se fait dans d'autres domaines (dessin industriel, architecture [...]).⁷⁹

Verso la metà degli anni Ottanta, la CAO riacquistò quindi forza, anche grazie alla compresenza di più fattori tecnologici: a) la diffusione dei personal computer e l'aumento della capacità di calcolo; b) lo sviluppo di interfacce grafiche; c) la nascita di standard di intercomunicazione tra i diversi dispositivi (lo standard MIDI, *Musical Instrument Digital Interface*, è del 1983); d) il perfezionamento dei linguaggi di programmazione. Il linguaggio di programmazione, in particolare, riveste un ruolo cruciale nel processo compositivo:

In the same way that the choice of a programming language influences the programmer, it plays a role in formalization of a musical idea. Indeed, the use of a language can suggest expressions that would not be favoured in another context. [...] The use of a programming language forces the musician to reflect on the process of formalization and

⁷⁷ Institut de Recherche et Coordination Acoustique/Musique, istituzione fondata nel 1977 da Pierre Boulez. <<http://www.ircam.fr>>.

⁷⁸ MIKHAIL MALT, "La composition assistée par ordinateur".

⁷⁹ TRISTAN MURAIL, "Spectre et lutins", p. 43.

avoids him to consider the computer as a black box which imposes its choices to him. The programming language offer the composer an enormous freedom of decision, in exchange of a certain effort in formalization and design.⁸⁰

Il linguaggio di programmazione che più di ogni altro si è imposto come fondamento per la composizione assistita all'elaboratore, e che a partire dagli anni Novanta ha portato all'affermazione dei principali software per la composizione assistita (su tutti, *Patchwork* e poi *OpenMusic*) è LISP.

2.3 Il linguaggio LISP

2.3.1 Il paradigma funzionale

LISP è un linguaggio di programmazione ad alto livello basato sul paradigma funzionale: a differenza ad esempio del paradigma procedurale, ove una programmazione si attua come una sequenza di istruzioni, questo tipo di linguaggi si concentra sul concetto di funzione matematica. Il calcolo è inteso come 'valutazione' delle funzioni.⁸¹ Per usare le parole di Gregory Chaitin, LISP «è più simile alla matematica pura rispetto alla maggior parte dei linguaggi di programmazione». Basandosi sul paradigma funzionale, infatti, «invece di eseguire computazioni solo con in numeri, lavora con espressioni simboliche».⁸² In altri termini, ha una sintassi uniforme, in cui i dati e le funzioni sono rappresentati allo stesso modo. Inoltre è un linguaggio espressivo: esiste una relazione diretta tra le espressioni simboliche del linguaggio e le strutture che esso rappresenta.⁸³

Il paradigma funzionale che contraddistingue LISP si è dimostrato particolarmente adatto anche alle esigenze musicali. Per portare un esempio, la trasposizione di un accordo può essere considerata come una funzione con tre parametri in ingresso: l'accordo iniziale, l'intervallo e la direzione della trasposizione. La valutazione della funzione restituisce l'accordo trasposto. Un caso più complesso è l'interpolazione tra due accordi: qui, i parametri in ingresso sono i due accordi (quello iniziale e quello finale) e il numero di singoli

⁸⁰ GÉRARD ASSAYAG, "Computer Assisted Composition today".

⁸¹ Per una riflessione sui diversi linguaggi di programmazione si veda il volume di MAURIZIO GABBRIELLI, SIMONE MARTINI, *Programming Languages: Principles and Paradigms*, Londra, Springer, 2010.

⁸² GREGORY CHAITIN, *Alla ricerca di Omega*, Milano, Adelphi, 2007, pp. 57-58.

⁸³ GERARD ASSAYAG, "CAO : vers la partition potentielle", in *Les cahiers de l'IRCAM : la composition assistée par ordinateur*, Parigi, IRCAM, 1993, p. 14.

passi dell'interpolazione. Ad ogni passo, la funzione calcola l'accordo le cui note sono ottenute interpolando le note corrispondenti agli accordi iniziali.⁸⁴ O ancora, la distorsione di uno spettro armonico può essere rappresentato da una funzione con tre parametri in ingresso: il numero di parziali, il coefficiente di distorsione e la frequenza fondamentale. Di seguito un esempio di distorsione dello spettro in linguaggio LISP.⁸⁵

- Definizione della funzione. `part` è il numero di parziali, `freq` la frequenza fondamentale, `dist` il coefficiente di distorsione:

```
(defun dist(part freq dist)
  (let ((res))
    (dotimes (n part res)
      (setf res (cons (* freq (expt(+ n 1) dist))res)))
    (reverse res)))
```

- Valutazione della funzione: `part = 10`, `freq = 100 Hz`, `dist = 1.1`:

```
(dist 10 100 1.1)
(100.0 214.3547 334.83698 459.47937 587.3095 717.7388 850.3698
984.9155 1121.158 1258.9254)
```

2.3.2 *La nascita di Common Lisp*⁸⁶

All'inizio degli anni Ottanta, la crescita esponenziale dei dialetti di LISP aveva portato a una babele di linguaggi non standardizzati. Tra i principali, in ordine di diffusione, ricordiamo InterLisp (eseguibile su computer *PDP-10* e *Xerox Lisp*), MacLisp (su *PDP-10*), Portable Standard Lisp (multiplatforma ma poco diffuso), Zetalisp e Scheme. Oltre agli evidenti problemi di compatibilità all'interno della famiglia LISP, questi linguaggi soffrivano di basse performance e di integrazione pressoché inesistente con altri linguaggi. Tale condizione ha indotto la comunità di utilizzatori a uno sforzo comune nel tentativo di creare un linguaggio standardizzato, sviluppato secondo i principi di comunanza, portabilità, consistenza, espressività, compatibilità, efficienza, potenza e stabilità.⁸⁷ L'esito

⁸⁴ Si veda ANDREW GERZSO, "Paradigms and Computer Music", in «Leonardo Music Journal», v. 2 n. 1, 1992, p. 77.

⁸⁵ La funzione si ritrova in un file di testo elaborato da Fausto Romitelli e conservato presso l'IRCAM per il quale si rimanda all'appendice C.

⁸⁶ Per una trattazione dettagliata dell'evoluzione storica di LISP, dalla sua nascita nel 1956 al MIT ad opera di John McCarthy fino alla sua standardizzazione come Common Lisp, si rimanda a GUY L. STEELE, RICHARD P. GABRIEL, "The Evolution of Lisp", in *History of Programming languages II*, New York, 1996, pp. 233–330.

⁸⁷ GUY L. STEEL, RICHARD P. GABRIEL, "The Evolution of Lisp", pp. 255-256.

di questa riflessione fu la nascita di Common Lisp. Apparso per la prima volta nel 1984,⁸⁸ fu sottoposto ad un lungo iter di consolidamento e, infine, di standardizzazione: nel 1986 fu infatti istituito il comitato tecnico X3J13 con la missione di elevare il nuovo linguaggio a standard ANSI (*American National Standards Institute*). L'obiettivo fu raggiunto con l'approvazione dello standard X3.226/1994,⁸⁹ del dicembre 1994.

A partire dal 1984, Common Lisp godette di un grande successo, in gran parte dovuto agli sviluppi nel campo dell'intelligenza artificiale (LISP si è storicamente affermato come il linguaggio più adatto), in parte per la sua peculiarità di linguaggio 'comune', in parte per la commercializzazione di workstation sempre più potenti, con performance non distanti da quelle progettate appositamente su LISP. Inoltre, Common Lisp funzionava (e funziona tutt'ora) su ogni computer e ogni sistema operativo, abbattendo così una barriera per la sua diffusione. Negli anni è diventato un linguaggio multi-paradigma, affiancando al paradigma funzionale anche quelli procedurale e orientato ad oggetti: ha sviluppato la capacità di interpretare e di interagire con altri linguaggi di programmazione e ha integrato, a partire dal 1988, CLOS⁹⁰ (*Common Lisp Object System*), un sistema di programmazione orientata agli oggetti basato su «generic functions, multiple inheritance, declarative method combination, and a meta-object protocol».⁹¹ CLOS andrà a costituire una parte fondamentale dello standard ANSI.

Da LISP, e in particolare da Common Lisp e CLOS, negli anni sono stati sviluppati nuovi ambienti software orientati alla composizione musicale, da un lato presso il CCRMA di Stanford (*Common Music*), dall'altro, più rilevante in questa sede, presso l'IRCAM di Parigi.

⁸⁸ Data di pubblicazione del testo di riferimento: GUY L. STEELE, JR., *Common Lisp the Language*, Newton (MA), Digital Press, 1984.

⁸⁹ ANSI CL standard 1994. ANSI INCITS 226-1994, Information Technology - Programming Language - Common Lisp.

⁹⁰ Si veda SONYA E. KEENE, *Object-Oriented Programming in Common Lisp: A Programmer's Guide to CLOS*, Addison-Wesley, 1989

⁹¹ DANIEL G. BOBROW, LINDA G. DEMICHIEL et al., "Common Lisp Object System specification", in «Higher-order And Symbolic Computation», v. 1, n. 3-4, 1989, p. 248.

2.3.3 *Common Music*

Common Music è stato sviluppato sul finire degli anni Ottanta da Heinrich Traube presso il CCRMA di Stanford ed è basato su Common Lisp e CLOS. Era in grado di funzionare su diverse piattaforme quali Macintosh, SGI, NeXT, SUN e Intel. Negli anni ha implementato la possibilità di interfacciarsi con altri programmi e linguaggi: MIDI, *Csound*, *Common Lisp Music*, *Music Kit*, *C Mix*, *C Music*, *M4C*, *Mix* e *Common Music Notation*.

Concettualmente, *Common Music* considera la composizione musicale come un'attività che coinvolge tre diverse aree, separate a livello di progettazione ma in costante dialogo tra loro: i) l'area dello sviluppo delle idee musicali; ii) l'area della concretizzazione di tali idee, la loro realizzazione pratica, il 'rendering'; iii) una volta reso udibile il risultato, è coinvolta l'area della 'performance'. In *Common Music* la performance si reinserisce nel processo compositivo secondo il principio del feedback. Ciò può avvenire in modo implicito quando essa guida il compositore allo sviluppo di nuove idee oppure in modo esplicito quando investe il controllo in tempo reale del risultato acustico.⁹²

Nonostante sia stato dotato nel 1995 di un'interfaccia grafica dedicata, *Capella* (figura 18),⁹³ *Common Music* è stato concepito per essere utilizzato prevalentemente con l'interfaccia testuale. Il motivo è da ricercarsi nei principi espressi da Traube che stanno a fundamenta di *Common Music* quali portabilità e integrazione con i diversi linguaggi e software: «le texte est un format qui permet facilement les échanges entre les diverses applications et entre les diverses plateformes informatiques».⁹⁴ Per portare un esempio, l'interfaccia grafica *Capella* è stata sviluppata esclusivamente per sistemi *Macintosh* in linguaggio Macintosh Common Lisp.

⁹² Si veda HEINRICH TAUBE, "Common Music: A Music Composition Language in Common Lisp and CLOS", in *Computer Music Journal*, v. 15 n. 2, 1991, pp. 21-32 e HEINRICH TAUBE, "An Introduction to Common Music", in *Computer Music Journal*, v. 21 n. 1, 1997, pp. 29-34.

⁹³ Si veda HEINRICH TAUBE, TOBIAS KUNZE, "Capella: A Graphical Interface for Algorithmic Composition", in *Proceedings of the 1995 International Computer Music Conference*, San Francisco, International Computer Music Association, 1995, pp. 377-380.

⁹⁴ MIKHAIL MALT, "La composition assistée par ordinateur".

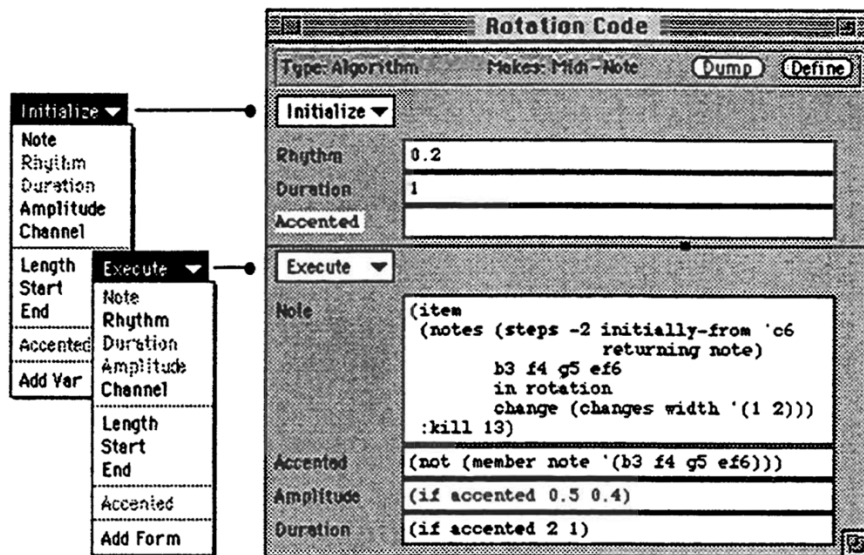


Figura 18 – Esempio di interfaccia grafica Capella. Si può notare come il codice LISP sia in buona parte ancora espresso.⁹⁵

2.3.4 Da Formes a Patchwork

Anche all'IRCAM, LISP si affermò come il linguaggio di programmazione di riferimento. Sin dal primo ambiente per la CAO lì sviluppato – *Formes* – i ricercatori lavorarono a partire da dialetti di LISP, cercando via via di adattarne il linguaggio: l'obiettivo era di fornire ai compositori un ambiente che permettesse loro di concentrarsi più sull'espressione dei problemi musicali che sul linguaggio di programmazione.⁹⁶ Nell'arco di pochi anni, presso l'IRCAM furono sviluppati diversi ambienti software orientati alla composizione assistita:

- *Formes*, concepito da Xavier Rodet e Pierre Cointe⁹⁷ tra il 1982 e il 1985. Inizialmente sviluppato su computer *PDP-10* e scritto in VLisp, era un ambiente per la sintesi e la composizione musicale basato sulla programmazione orientata agli oggetti. Gli oggetti musicali venivano strutturati e organizzati temporalmente in forma di processi;
- *PreForm*,⁹⁸ poi *PreForm/Esquisse*, fu il primo tentativo di applicare a *Formes* un'interfaccia grafica: sviluppato da Lee Boyton, Jacques Duthen e Pierre-François

⁹⁵ HEINRICH TAUBE, TOBIAS KUNZE, "Capella: A Graphical Interface for Algorithmic Composition", p. 379

⁹⁶ Si veda GERARD ASSAYAG, "CAO : vers la partition potentielle", p. 14.

⁹⁷ Si veda XAVIER RODET, PIERRE COINTE, *FORMES - composition et ordonnancement de processus*, Rapport Ircam 36/85, Parigi, IRCAM, 1985.

⁹⁸ Si veda JEAN-BAPTISTE BARRIERE, "Devenir de l'écriture musicale assistée par ordinateur : formalisme, forme, aide à la composition", in «Analyse musicale», 20, 1990, pp. 52-68.

Baisnée, con l'assistenza diretta di Tristan Murail, il sistema fornisce al compositore una serie di oggetti multidimensionali e di funzioni in grado di trasformarli (intervalli, distorsioni spettrali, interpolazioni di accordi e così via);

- *Crime*,⁹⁹ sviluppato in Le_Lisp su computer VAX da Gérard Assayag insieme a Claude Malherbe, ha portato una significativa innovazione: l'output poteva rappresentare in modo simbolico la partitura, visualizzabile pertanto in forma di notazione musicale.

Formes, *PreForm/Esquisse*, *Crime*, possono essere considerati i precursori dell'ambiente software che più di ogni altro influenzerà la CAO negli anni Novanta: *Patchwork*.

2.4 Patchwork

2.4.1 La patch: una nuova veste grafica per LISP

Lo sviluppo di *Patchwork*, iniziato da Mikael Laurson, Jacques Duthen e Camilo Rueda alla fine degli anni Ottanta e proseguito per un decennio,¹⁰⁰ ha segnato una tappa fondamentale nella storia della CAO: *Patchwork* è un sistema di programmazione visuale, che quindi permette la manipolazione grafica degli oggetti. Fornisce un'interfaccia grafica al linguaggio Common Lisp (e CLOS) su cui è fondato. Le funzioni scritte in LISP possono essere tradotte in oggetti grafici tra loro connessi che, organizzati, vanno a costituire la 'patch'. Una patch visuale vuole definire un modello computazionale in cui i singoli oggetti assumono il ruolo di una funzione.

L'esempio in figura 19 riporta una patch creata da Tristan Murail ed inserita come patch esemplificativa all'interno del software. Qui Murail ha riscritto in *Patchwork* un procedimento adottato per la composizione della decima sezione di *Désintégrations* (1983):¹⁰¹

⁹⁹ Si veda GÉRARD ASSAYAG, CAMILO RUEDA et al., "Computer-Assisted Composition at IRCAM: From PatchWork to OpenMusic", in *Computer Music Journal*, v. 23 n. 3, 1999.

¹⁰⁰ Presso l'IRCAM *Patchwork* ha poi fornito la base concettuale e linguistica per lo sviluppo di *OpenMusic*, ambiente per la composizione assistita tutt'ora in uso (si veda CARLOS AUGUSTO AGON AMADO, *OpenMusic : Un langage visuel pour la composition musicale assistée par ordinateur*, tesi di dottorato sostenuta presso l'Université Paris 6, 1998). Per una descrizione della transizione tra i due sistemi, avvenuta sul finire degli anni Novanta, si rinvia a GÉRARD ASSAYAG, CAMILO RUEDA et al., "Computer-Assisted Composition at IRCAM: From PatchWork to OpenMusic", pp. 59-72. Va segnalato che nonostante lo sviluppo di *Patchwork* entro le mura ircamiane sia stato abbandonato da tempo, esso continua a fornire la base per altri progetti (vale la pena di segnalare *PWGL*, software compatibile con le più recenti versioni di *Mac OS X*, a firma dello stesso Mikael Laurson, di Mika Kuuskankare, di Kilian Sprotte e di Vesa Norilo).

¹⁰¹ MIKHAIL MALT, CURTIS ROADS, *PatchWork – Introduction*, Parigi, IRCAM, 1996, p. 16. Si veda inoltre ANTHONY CORNICELLO, *Timbral Organization in Tristan Murail's Désintégrations and Rituals by Anthony*

data una nota fondamentale (in questo caso un Mi_1) il suo valore in midicent¹⁰² viene convertito in frequenza (modulo 'mc->f'). Il valore così ottenuto viene moltiplicato (modulo 'g*') per una lista di parziali definita dal costruttore *const*. Si viene così a creare una lista di undici frequenze che devono essere riconvertite in midicent ('f->mc') per essere visualizzate sotto forma di notazione tradizionale ('chord').

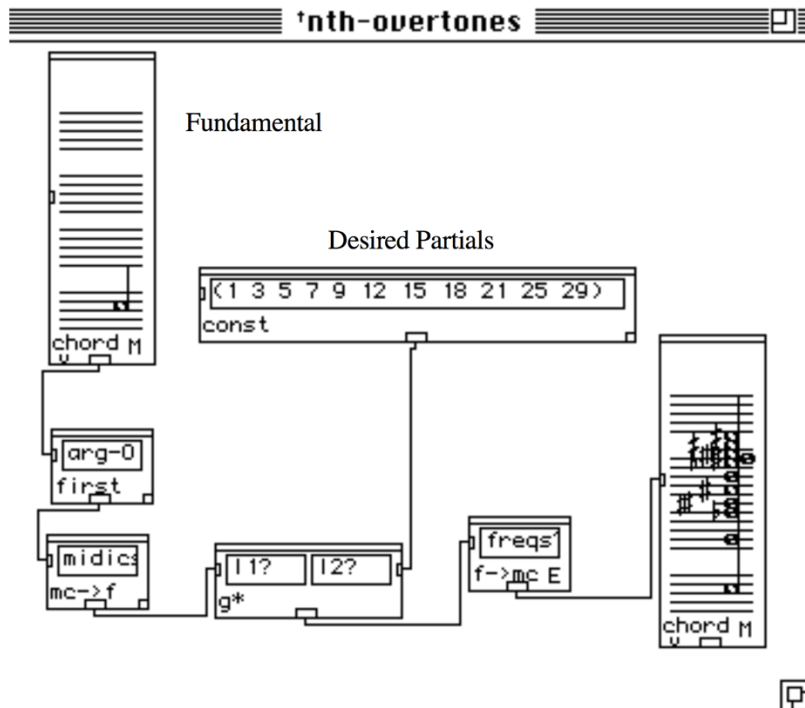


Figura 19 – Esempio di patch. Generazione di uno spettro armonico in cui sono selezionate solo alcune parziali.

Il passaggio a elaborazioni più complesse, in linea con l'armonia frequenziale proposta da Murail, è breve. Date le frequenze portanti, modulanti, e l'indice di modulazione,¹⁰³ e note le funzioni matematiche alla base della modulazione, diviene semplice costruire una patch che restituisce in notazione tradizionale il risultato della sintesi (figura 20).

Cornicello, tesi di dottorato presso la Faculty of the Graduate School of Arts and Sciences, Brandeis University, Waltham, 2000.

¹⁰² Il midicent è una particolare codifica della notazione tradizionale che fonde il protocollo MIDI e la suddivisione in cents. La codifica MIDI assegna un valore da 0 a 127 ad ogni semitono, dal Do_{-1} (8,175 Hz) al Sol_9 (12557 Hz) abbracciando così più di dieci ottave. Il cent è una misura microintervallare che corrisponde ad un centesimo di semitono (un'ottava corrisponde a 1200 cents). Il midicent può essere quindi considerato un'estensione del sistema MIDI, in cui ad esempio un Do_4 (nota MIDI 60) è rappresentata dal valore 6000, mentre un $Do_{\sharp 4}$ assumerà valore 6050.

¹⁰³ Si veda *infra*, par. 1.2.3.

Per la prima volta l'integrazione dell'interfaccia grafica è pensata fin dalle prime fasi di sviluppo. Mentre *Capella*, l'interfaccia grafica di *Common Music*, era dedicata quasi esclusivamente alla gestione delle operazioni di input e di output e alla visualizzazione delle funzioni sotto forma di grafici, in *Patchwork* l'interfaccia è utilizzata in modo aperto, flessibile e quindi programmabile dall'utente.

Ad alcune classi di oggetti è stato affiancato un editor grafico, ad esempio una BPF o *break point function* (figura 22) che permette di modificare graficamente i valori di una o più curve uno spazio bidimensionale, oppure un editor musicale per modificare i parametri di un aggregato accordale – altezza, dinamica, durata, canale, grado di approssimazione della conversione da frequenza a notazione tradizionale – (figura 23), così come un editor ritmico (figura 24) capace di gestire configurazioni ritmiche complesse.

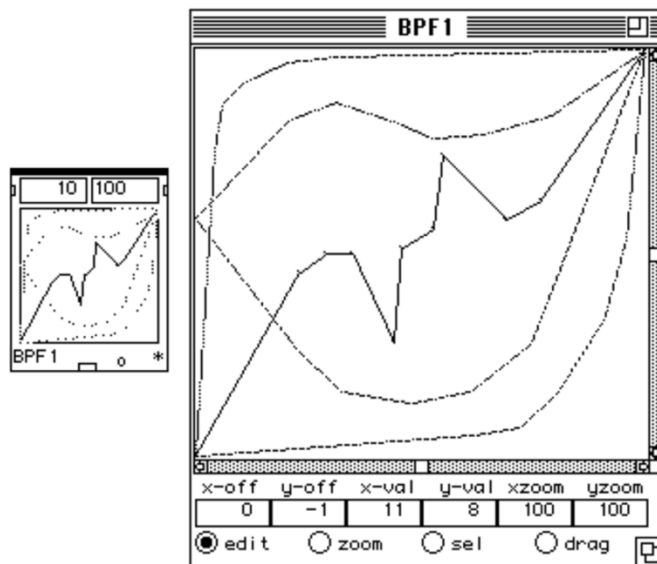


Figura 22 – 'Multi-BPF'.¹⁰⁶

¹⁰⁶ «The multi-bpf module can be used to create and edit simultaneous breakpoint functions an once, create and edit coordinate pairs (x,y), display a series of coordinate pairs, either as a BPF or as a series of points, save and load multi-bpf modules to and from a library and retrieve data concerning the points contained in the multi-bpf module.» *Ivi*, p. 97.

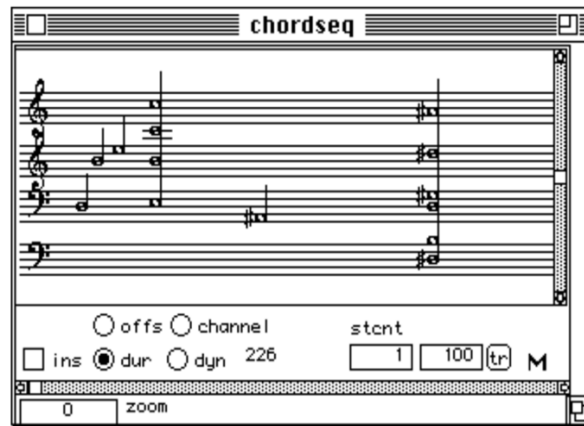


Figura 23 – Editor grafico di una sequenza accordale.¹⁰⁷

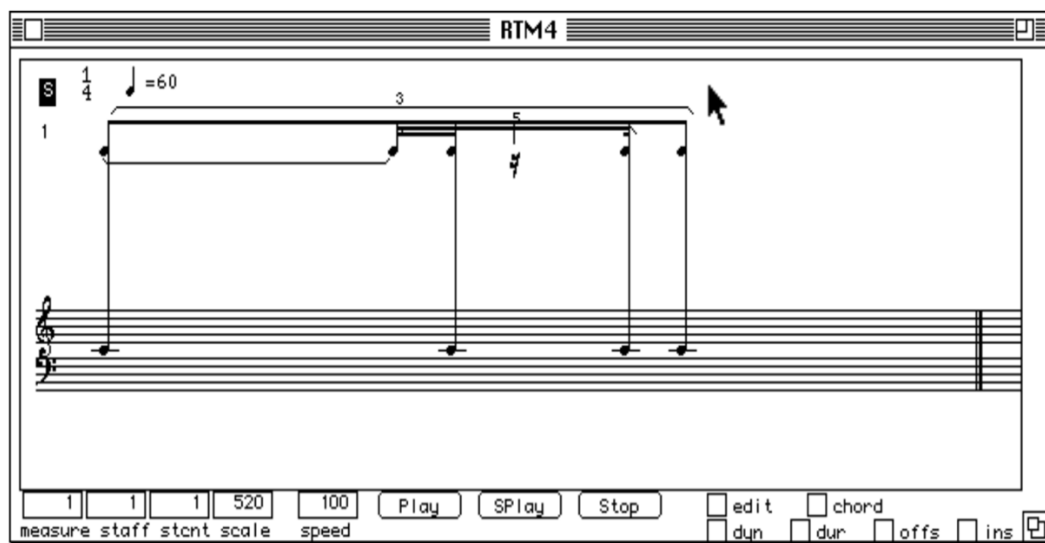


Figura 24 – RTM. Editor grafico di un pattern ritmico.¹⁰⁸

¹⁰⁷ Ivi, p. 110.

¹⁰⁸ Ivi, p. 115.

Patchwork si appoggia su di un set di funzioni base, secondo gli autori musicalmente neutre «in the sense that it does not make assumptions about which kind of musical material is to be produced with it».¹⁰⁹ A queste funzioni base si possono poi affiancare diverse librerie esterne, sviluppate sulla base di specifiche esigenze musicali.¹¹⁰ I compositori avevano quindi la possibilità di programmare gli strumenti più utili al proprio lavoro, sia sotto forma di oggetti che di patch, organizzati in una libreria personale.

Le potenzialità di *Patchwork* risiedono nella sua estrema flessibilità e nella capacità di interfacciarsi con numerosi software di analisi e di sintesi del suono:

il permet de programmer des algorithmes, ou « patches », destinés à créer et à manipuler le matériau musical représenté par des notes, des rythmes, des enveloppes ou des structures formelles. Il permet également de contrôler différents programmes de synthèse sonore. Il peut s'agir de synthétiseurs MIDI ou de logiciels de synthèse comme *Csound* et *Chant*. *Patchwork* peut communiquer avec de nombreux programmes – d'analyse, de traitement, de synthèse, de séquences, d'édition de partition – et sous différentes formes – fichiers texte, MIDI, binaire, etc.¹¹¹

La combinazione di semplicità della programmazione, potenza dell'interfaccia grafica e capacità di comunicare con sistemi diversi contribuirono alla diffusione di *Patchwork* a livello europeo.¹¹²

¹⁰⁹ GÉRARD ASSAYAG, CAMILO RUEDA, et al., "Computer-Assisted Composition at IRCAM: From PatchWork to OpenMusic", p. 59.

¹¹⁰ Negli anni, alcune di queste librerie sono state integrate direttamente nell'ambiente *Patchwork*. È il caso ad esempio di «*Esquisse*, conceived by the composer Tristan Murail, for spectral music operations; *Chaos/Alea*, by Mikhail Malt, for the definition of musical structures through stochastic manipulations and dynamic models; *Kant*, by Gerard Assayag, Carlos Agon, Joshua Fineberg, and Camilo Rueda (1994), for intelligent rhythm transcription; *PWConstraints*, by Mikael Laurson, for rule-based programming; *Situation*, by Camilo Rueda and Antoine Bonnet, for constraint-based generation of harmonic and rhythmic structures; and *Repmus* by Gerard Assayag and Claudy Malherbe for harmonic and rhythmic structure generation and processing» (GÉRARD ASSAYAG, CAMILO RUEDA, et al., "Computer-Assisted Composition at IRCAM: From PatchWork to OpenMusic", p. 62).

¹¹¹ LAURENT POTTIER, *Le contrôle de la synthèse sonore informatique*, pp. 89-90.

¹¹² Molti compositori si confrontarono con *Patchwork*. Solo per citarne alcuni, Brian Ferneyhough, Gérard Grisey, Claudy Malherbe, Tristan Murail, Kaija Saariaho.

In questa sede è opportuno ricordare tre librerie in particolare, *Csound/Edit-sco*,¹¹³ *PW-Chant*¹¹⁴ e *SpData*,¹¹⁵ che permettono a *Patchwork* di dialogare con differenti sistemi di sintesi sonora, *Csound* e *Chant*, e di analisi, come ad esempio *AudioSculpt*.

2.4.2 Il controllo di Csound in Patchwork

2.4.2.1 Csound

Csound è uno dei più diffusi sistemi di sintesi sonora digitale. Esso rientra nella categoria di software per la sintesi basati sulla nozione di separazione tra generazione del suono e segnali di controllo, declinati come 'orchestra' e 'partitura', e sulle connessioni tra i diversi *unit generators* (UG), i moduli usati per costruire vari algoritmi di generazione sonora: l'uscita di un UG può essere connessa all'ingresso di qualsiasi altro UG. Esempi di UG sono generatori di segnale – oscillatori, generatori di rumore, generatori di impulsi – ma anche processori di segnale – filtri, riverberi, echi – e operatori quali moltiplicatori o addizionatori.

L'origine di questa tipologia di sintesi è da ricondursi al linguaggio *Music III*, sviluppato nel 1960 dal team guidato da Max Mathews presso il Laboratori Bell. *Music III* fu seguito da *Music IV*, scritto in assembly BEFAP, e soprattutto da *Music V*, punto di approdo della ricerca di Max Mathews. *Music V* fu scritto in linguaggio Fortran, un linguaggio 'standard' che ne ha favorito una diffusione capillare. Da allora, costanti perfezionamenti hanno portato a un numero considerevole di riscritture su piattaforme e linguaggi differenti: per citarne alcune, *Music 4BF*, *Music 7*, *Music 11*, *MUS10*, *Cmusic*, *Common Lisp Music* e *Csound* (tabella 2).

¹¹³ Si veda MIKHAÏL MALT, LAURENT POTTIER, *Patchwork. Csound/Edit-sco. Library of Modules for Generating Csound scores. Reference*, IRCAM – Centre Georges Pompidou, 1996.

¹¹⁴ Francisco Iovino, Mikael Laurson, Laurent Pottier, *PW-Chant – Reference*, Parigi, IRCAM, 1996.

¹¹⁵ Si veda LAURENT POTTIER, *Patchwork. SpData*, Parigi, IRCAM, 1997.

2.4 Patchwork

Program	Date introduced	Author(s)	Original host computer	Language	Location
Music III	1960	M. Mathews	IBM 7090	Assembler	Bell Laboratories, Murray Hill
Music IV	1963	M. Mathews J. Miller	IBM 7094	Macro assembler	Bell Laboratories, Murray Hill
Music IVB	1965	G. Winham H. Howe	IBM 7094	Macro assembler	Princeton University, Princeton
Music V	1966	M. Mathews J. Miller	GE 645	Fortran IV	Bell Laboratories, Murray Hill
MUS10	1966	J. Chowning D. Poole L. Smith	DEC PDP-10	PDP-10 assembler	Stanford University, Stanford
MUSIGOL (Music-Algol)	1966	D. MacInnes W. Wulf P. Davis	Burroughs 5500	Burroughs Algol	University of Virginia, Richmond
Music 4BF	1967	H. Howe G. Winham	IBM 360	Fortran II and BAL assembler	Princeton University, Princeton
Music 360	1969	B. Vercoe	IBM 360	BAL assembler	Princeton University, Princeton
Music 7	1969	H. Howe	Xerox XDS Sigma 7	Assembler	Queens College CUNY, Flushing
TEMPO	1970	J. Clough	IBM 360	BAL assembler	Oberlin Conservatory, Oberlin
B6700 Music V	1973	B. Leibig	Burroughs B6700	Fortran and Algol	University of California at San Diego (UCSD), La Jolla
Music II	1973	B. Vercoe S. Haflich R. Hale C. Howe	DEC PDP-11	Macro-11 assembler	MIT, Cambridge
MUSCMP	1978	Tovar	Foonly 2 (DEC PDP-10 clone)	FAIL assembler	Stanford University, Stanford
Cmusic	1980	F. R. Moore D. G. Loy	DEC VAX-11	C	UCSD, La Jolla
MIX	1982	P. Lansky	IBM 370	IBM 360 assembler	Princeton University, Princeton
Cmix	1984	P. Lansky	DEC PDP-11	C	Princeton University, Princeton
Music 4C	1985	S. Aurenz J. Beauchamp R. Maher C. Goudeseune	DEC VAX-11 (later ported other UNIX computers)	C	University of Illinois at Urbana-Champaign
Csound	1986	B. Vercoe R. Kartstens	DEC VAX-11 (later ported to IBM PC, Apple Macintosh etc.)	C	MIT, Cambridge
Music 4C	1988	G. Gerrard	Apple Macintosh	C	University of Melbourne, Parkville
Common Lisp Music	1991	W. Schottstaedt	NeXT	Common Lisp	Stanford University, Stanford

Tabella 2 – Software di sintesi basati su UG.¹¹⁶

¹¹⁶ CURTIS ROADS, *The Computer Music Tutorial*, Cambridge-Londra, The MIT Press, 1996, p. 789-790.

Nonostante modifiche anche sostanziali a livello di programmazione – dal linguaggio Fortran si è passati ad esempio a C oppure a Common Lisp – il principio di funzionamento che contraddistingue tutta la famiglia *Music-N* è rimasto pressoché inalterato.

Così come gli altri software derivati dalla famiglia *Music-N*, la sintesi sonora in *Csound* richiede la scrittura di due documenti:¹¹⁷ l'*orchestra* e la 'partitura' o *score*. L'*orchestra* descrive gli 'strumenti' da utilizzare (oscillatori, filtri...) e la natura dei parametri per il loro controllo. Ad esempio:

```
instr 1
a1  oscil p4,  p5,  p6
out  a1
endin
```

definisce un oscillatore governato dai parametri *p4* (ampiezza), *p5* (frequenza) e *p6* (dimensione e tipo di tabella che andrà a leggere).

Lo *score* invece include i valori destinati al controllo degli strumenti definiti dall'*orchestra*. Si compone di: a) una prima dichiarazione che descrive dimensione e tipo di tabella dell'oscillatore, ad esempio:

```
f1  0    1024 10    1
```

descrive una tabella che si attiva all'istante 0, di grandezza 1024, descritta come somma di segnali sinusoidali (*subroutine* GEN10) e con ampiezza delle parziali successive alla prima pari a 0. b) la dichiarazione della partitura, ossia il comportamento di ogni singolo strumento. Nel caso di una sintesi additiva:

```
;p1  p2    p3    p4    p5    p6
i1   0    3    3000 100    1
i1   0    3    3000 214    1
i1   0    3    3000 335    1
```

dove *p1* richiama lo strumento *instr1*, *p2* indica l'istante iniziale, *p3* la durata, *p4* l'ampiezza, *p5* la frequenza e *p6* la tabella dell'oscillatore. In questo caso viene prodotto un segnale costituito da una somma di tre sinusoidi tra loro in rapporto non armonico.

¹¹⁷ Nelle moderne versioni di *Csound* i file *orchestra* (.orc) e *score* (.sco) sono accorpati all'interno di un unico file di testo con estensione .csd.

2.4.2.2 Csound/Edit-sco

La libreria *Csound/Edit-sco*,¹¹⁸ sviluppata da Laurent Pottier e Mikhail Malt presso l'IRCAM e introdotta nel 1993, fornisce una serie di moduli a *Patchwork* finalizzati alla generazione degli *score* per *Csound*. Essa permette di sfruttare la capacità algoritmica e grafica di *Patchwork* per manipolare i dati destinati alla sintesi (altezze, ampiezze, durate, funzioni e tabelle). I parametri per la sintesi possono essere ottenuti con modalità differenti, sia utilizzando l'interfaccia grafica ad oggetti, sia mediante i moduli e gli editor grafici per la notazione tradizionale.

L'esempio riportato in figura 25 mostra una patch in *Patchwork* che va a scrivere un file *score* per *Csound*. In questo caso, lo *score* andrà a controllare un oscillatore ad involuppo variabile:

```
instr 1
kenvamp   linseg      0, p6, p4, p3-2*p6, p4, p6, 0
ason oscili   kenvamp, p5, 1
          out        ason
endin
```

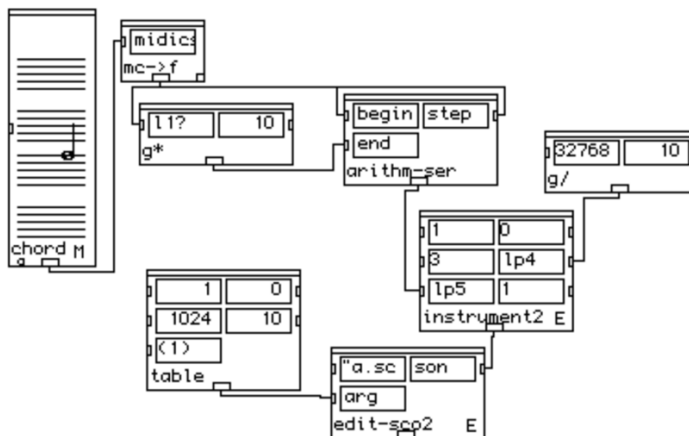


Figura 25 – Patch per la scrittura dello *score* di *Csound* per generare uno spettro armonico mediante sintesi additiva.

La patch è così descritta da Laurent Pottier:

On part d'une note, La2, dont on calcule la fréquence, 220 Hz, avec l'objet *mc->f*, puis on calcule les dix premiers multiples de cette fréquence avec une fonction qui calcule des séries arithmétiques.

¹¹⁸ Si veda MIKHAIL MALT, LAURENT POTTIER, *PW Csound/Edit-sco – Library of Modules for Generating Csound scores – reference*, Parigi, Ircam, 1996.

Le module *instrument2* est un module extensible dont chaque entrée correspond à une colonne dans la partition. L'entrée n°1 indique donc le numéro de l'instrument (i), l'entrée n°2 donne la date du début du son (temps = 0"), l'entrée n°3 donne la durée du son (3") qui sera la durée de chacun des partiels que l'on va additionner. Pour avoir des durées différentes pour chaque partiel, il faut donner une liste de valeurs, au lieu de donner une valeur unique. L'entrée n°4 donne l'amplitude des partiels (ici, l'amplitude maximum de l'onde divisée par le nombre de partiels), l'entrée n°5 donne la liste des fréquences et l'entrée n°6 donne le numéro de la table utilisée pour donner la forme d'onde de chaque partiels.

Le module *edit-sco2* produit le fichier partition en regroupant les données de type *note* et les données de type *table*. Les données de type *table* correspondent ici à la description d'une table sinusoïdale.¹¹⁹

L'output della patch è il seguente :

```
f1 0 1024 10 1
i1 0 3 3276.0 220 1
i1 0 3 3276.0 440 1
i1 0 3 3276.0 660 1
i1 0 3 3276.0 880 1
i1 0 3 3276.0 1100 1
i1 0 3 3276.0 1320 1
i1 0 3 3276.0 1540 1
i1 0 3 3276.0 1760 1
i1 0 3 3276.0 1980 1
i1 0 3 3276.0 2200 1
e
```

dove *f1* definisce la tabella – in questo caso l'oscillatore genera sinusoidi – mentre le righe di codice *i1* definiscono i parametri delle singole parziali, tra loro in rapporto armonico.

2.4.3 Il controllo di Chant in Patchwork

2.4.3.1 Chant

Il sintetizzatore *Chant*, concepito dall'équipe guidata da Xavier Rodet all'IRCAM già alla fine degli anni Settanta, ha innegabilmente rivestito un ruolo fondamentale nella sintesi

¹¹⁹ LAURENT POTTIER, *Le contrôle de la synthèse sonore informatique*, p. 96.

della voce.¹²⁰ Ma la sua concezione, fondata sulla sintesi mediante funzione d'onda formantica (FOF)¹²¹ e quindi sul principio della risposta all'impulso, permette agevolmente di applicare il modello alla generazione di altre tipologie di suoni, ad esempio suoni percussivi, dai piatti alle campane tubolari agli archi pizzicati. Ne è riprova l'utilizzo estensivo che Jean-Baptiste Barrière fa della sintesi in *Chant* nella sua opera *Hybris* (1988), un brano fondato sui processi di interpolazione di timbri sintetizzati secondo la seguente metodologia:

Modélisation (modèle arbitraire ou issu de l'analyse d'instruments traditionnels), hybridation (mélange « statique » de deux modèles), interpolation ou transition (mélange « dynamique » de deux modèles), extrapolation (transformation préservant la reconnaissance de la référence) et abstraction (« aliénation » de la référence).¹²²

Questo tipo di analisi, trattamento e sintesi del suono va sotto il nome di *modèles de resonance*.¹²³ L'impiego di tali modelli permette a Barrière di sintetizzare ed elaborare suoni quali contrabbasso, tam-tam e piastre metalliche.¹²⁴

¹²⁰ Si veda XAVIER RODET, YVES POTARD, JEAN-BAPTISTE BARRIÈRE, "The CHANT Project: From the Synthesis of the Singing Voice to the Synthesis in General", in «Computer Music Journal», v. 8 n. 3, 1984, pp. 15-31.

¹²¹ La sintesi per funzione d'onda formantica è una tecnica sviluppata da Xavier Rodet e originariamente pensata per la sintesi di suoni vocali. La FOF assume che il tratto vocale possa essere modellizzato come un risonatore complesso in risposta a un impulso. La prima versione della sintesi FOF è assimilabile al principio della sintesi sottrattiva: dato un segnale, sia esso un treno di impulsi o un rumore a banda larga, viene veicolato attraverso un banco di filtri che ne seleziona le bande frequenziali corrispondenti alle formanti del suono da sintetizzare. Una versione successiva realizza invece un approccio additivo: generatori di sinusoidi di breve durata, minuziosamente controllati in ampiezza, sono impiegati al posto dei filtri. Il segnale prodotto da un generatore FOF sinusoidale equivale al segnale prodotto dalla risposta all'impulso di un filtro centrato sul picco formantico desiderato. Dato un treno di impulsi, un singolo generatore FOF produce per ogni picco un'intera sequenza di sinusoidi. L'utilizzo di segnali sinusoidali in vece dei filtri comporta grandi vantaggi in termini di potenza di calcolo richiesta.

¹²² Testo tratto dal programma di sala di *Hybris*, scritto in occasione del concerto svoltosi presso l'*Espace de projection* all'IRCAM il 28 giugno 1990. <<http://brahms.ircam.fr/works/work/6634/#program>>, ultima visita 5 marzo 2017.

¹²³ Si veda PIERRE-FRANÇOIS BAISNEE, YVES POTARD, JEAN-BAPTISTE BARRIÈRE, "Méthodologies de synthèse du timbre : l'exemple des modèles de resonance", in Jean-Baptiste Barrière (a cura di), *Le timbre, métaphore pour la composition*, Parigi, Christian Bourgois, 1990, pp. 135-163.

¹²⁴ LAURENT POTTIER, *Le contrôle de la synthèse sonore informatique*, pp. 141-145.

2.4.3.2 *PW-Chant*

Chant, scritto originariamente in linguaggio SAIL su computer *PDP-10* da Xavier Rodet, Yves Potard e Conrad Cummings,¹²⁵ beneficiò di diverse riscritture. Nel 1989 una nuova versione, scritta in linguaggio C e portata su ambiente *Macintosh*, contemplò la possibilità di essere controllata da linguaggio LISP. Così come per *Csound*, nei primi anni Novanta è stata quindi sviluppata una libreria che consentisse a *Patchwork* di comunicare con il sintetizzatore digitale *Chant: PW-Chant*. In questo caso, però, la patch non produce un file esterno come nel caso di *Csound* bensì assume il ruolo di interfaccia diretta per la generazione del suono. L'esito dell'operazione è il file sonoro salvato su disco.

Per potersi interfacciare con *Chant*, la patch prevede due fasi: la creazione dello strumento e, in seguito, la definizione delle 'note' che andranno a controllare lo strumento stesso. La patch è costituita da almeno due diverse finestre: la finestra che porta il nome della patch – dove vengono definite le 'note' solitamente rappresentate attraverso il modulo 'Chordseq' – e la finestra del modulo strumentale. Come si può notare in figura 26, il modulo strumentale 'reson' viene rappresentato nella finestra principale sotto forma di astrazione (il modulo 'absout' della finestra 'reson' genera un output grafico nel modulo 'reson' della finestra principale). La figura 26 riporta un semplice esempio di sintesi FOF di un suono di campana tubolare – si può notare il richiamo al modello di risonanza 'tubul' nella finestra dello strumento. La frequenza fondamentale della sintesi è impostata a 4800 midicents, equivalenti a un Do_3 a 130,81 Hz, la durata a 2,75 secondi.

¹²⁵ XAVIER RODET, YVES POTARD, JEAN-BAPTISTE BARRIÈRE, "The CHANT Project: From the Synthesis of the Singing Voice to the Synthesis in General", p. 30.

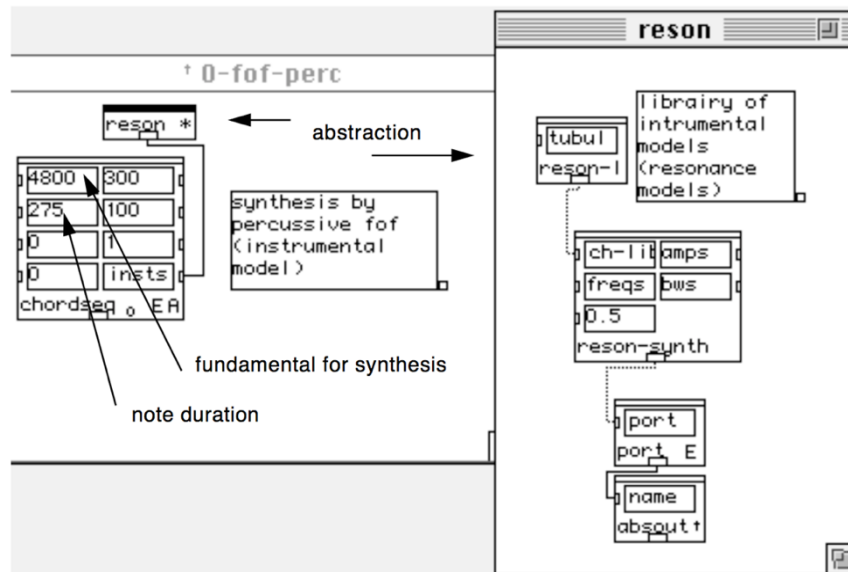


Figura 26 – Esempio di patch per la sintesi FOF di un suono di campana tubolare.¹²⁶

2.4.4 SP-Data

Da segnalare, infine, lo sviluppo di *SpData*,¹²⁷ rilasciato ufficialmente nel 1996 e probabilmente già in fase sperimentale da qualche anno.¹²⁸ *SpData* è una libreria di *Patchwork* destinata alla lettura, alla scrittura, alla creazione, alla trasformazione e allo studio dei dati numerici provenienti da diversi software dedicati all'analisi spettrale del suono – su tutti, *AudioSculpt*.¹²⁹

Le analisi spettrali, siano esse statiche – una singola FFT – oppure dinamiche – come ad esempio l'analisi tempo-frequenza con *AudioSculpt* –, possono essere quindi manipolate secondo il paradigma della programmazione orientata agli oggetti.

Nel caso di *AudioSculpt* l'analisi dinamica è costituita da una successione di FFT effettuate con finestra costante sul segnale. Si riporta a titolo di esempio l'analisi della

¹²⁶ FRANCISCO IOVINO, MIKAEL LAURSON, et al., *PW-Chant – Tutorial*, Parigi, IRCAM, 1996, p. 11.

¹²⁷ Si veda LAURENT POTTIER, *Patchwork. SpData*, Parigi, IRCAM, 1997.

¹²⁸ Nelle patch utilizzate per la generazione della parte elettronica di *EnTrance* di Fausto Romitelli è possibile rilevare l'utilizzo di moduli *SpData*. Le patch, conservate presso gli archivi del dipartimento di produzione dell'IRCAM, sono datate fine 1995.

¹²⁹ *AudioSculpt* è un software di visualizzazione, analisi ed elaborazione del suono sviluppato da Philippe Depalle e Chris Rogers all'IRCAM. Già a metà degli anni Novanta il suono poteva essere visualizzato sia in tempo/ampiezza sia come sonogramma. Le funzionalità di sintesi e di elaborazione del suono – compressione o espansione temporale, trasposizione, *cross-synthesis* – derivano dal programma *SVP*, a sua volta basato sul principio del *phase vocoder*, del quale originariamente *AudioSculpt* doveva costituire l'interfaccia grafica. Si veda PETER HANAPPEUNDER, MARIE-HÉLÈNE SERRA, *AudioSculpt User's manual*, Parigi, IRCAM, 1996.

prima finestra di un suono di violoncello con fondamentale a 63,849 Hz.¹³⁰ L'output del software è strutturato nel seguente modo: ciascuna riga, fatta eccezione la prima in cui sono indicati il numero di parziali e il segmento temporale analizzato, rappresenta ogni singola parziale; ogni parziale è quindi definita da quattro diversi parametri: il numero, la frequenza, l'ampiezza e la fase.

30	0.200000		
1	63.849	0.015965	-2.641355
2	127.44	0.185832	-0.269728
3	190.93	0.251104	2.841070
4	254.36	0.135406	0.862897
5	317.74	0.126968	1.172801
6	381.08	0.017780	-1.353609
7	444.39	0.030557	-2.482815
8	507.67	0.031252	-1.038302
9	570.94	0.083474	1.778481
10	634.18	0.056069	0.993245
11	697.4	0.023534	1.674749
12	760.6	0.018169	1.511033
13	823.79	0.016025	-2.451984
14	886.97	0.013155	2.604112
15	950.13	0.015403	2.021511
16	1013.3	0.006045	-1.149924
17	1076.4	0.005200	-0.870073
18	1139.5	0.013674	2.621776
19	1202.7	0.035349	-0.328225
20	1265.8	0.021674	-0.296857
21	1328.9	0.003601	-1.438232
22	1392.0	0.006043	-2.923688
23	1455.0	0.004217	1.409028
24	1518.1	0.003553	2.755301
25	1581.2	0.017199	-0.708678
26	1644.2	0.016684	0.002948
27	1707.3	0.005724	-0.905953
28	1770.3	0.007363	-2.697055
29	1833.4	0.008193	-2.943879
30	1896.4	0.002509	1.603571

In questo caso, la libreria *SpData* contiene l'oggetto 'addsyn-read' che è in grado di interpretare i dati dell'analisi – solitamente salvati in un file di testo con estensione '.format' – e di renderli fruibili all'interno di *Patchwork* (figura 27).

¹³⁰ Questa analisi è stata rinvenuta tra il materiale preparatorio utilizzato per la parte elettronica di *EnTrance*.

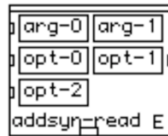


Figura 27 – Modulo ‘addsyn-read’. Parametri in ingresso, in ordine dall’alto al basso: 1) nome del file contenente l’analisi; 2) tipo di codifica del testo, ASCII o binario; 3) inizio della lettura del file, opzionale; 4) fine della lettura del file, opzionale; 5) numero di parziali da considerare, opzionale.

SpData è inoltre in grado di manipolare dati provenienti dall’analisi per *modèles de résonance*, una tipologia di analisi che vuole individuare i modi di risonanza di uno strumento studiando il rapporto tra diverse FFT di dimensione via via crescente. Come ricorda Pottier,¹³¹ l’analisi per *modèles de résonance* consente di descrivere una funzione di trasferimento di un filtro, costituito da un insieme di risonatori elementari, che si avvicina al comportamento di uno strumento acustico.

L’impiego di finestre di analisi di diverse dimensioni (da 2 millisecondi a 1,2 secondi) permette di individuare sia le risonanze lunghe sia quelle più brevi, che si producono nelle primissime fasi del suono. In quest’ultimo caso, la risonanza è rilevata dalle FFT più ‘strette’ e quindi, secondo il principio di indeterminazione, meno precise nel dominio delle frequenze. Viceversa, una risonanza lunga sarà captata da una finestra di analisi più ‘larga’, e quindi sarà descritta in termini frequenziali con più precisione. Ne consegue che il valore della larghezza di banda restituito dall’analisi sarà inversamente proporzionale alla durata della risonanza: più larga la banda, più breve sarà la risonanza.

Possiamo quindi affermare che l’analisi per *modèles de résonances*, a differenza della DFT (*Discrete Fourier Transform*), integra le informazioni temporali all’interno di un’unica serie di dati. Di seguito un esempio di analisi di un suono di gong:¹³² le colonne indicano rispettivamente numero del risonatore, la frequenza centrale, l’ampiezza e la larghezza di banda.

No	freq	amp	
1	176.206	-39.176	0.281
2	228.341	-26.994	0.341
3	238.027	-39.681	0.542
4	243.154	-21.850	9.36
5	254.890	-35.917	0.169
6	485.443	-30.525	0.273
7	555.061	-38.792	0.321

¹³¹ LAURENT POTTIER, *Patchwork. SpData*, Parigi, IRCAM, 1997, p. 8.

¹³² Questa analisi è stata rinvenuta tra il materiale preparatorio utilizzato per la parte elettronica di *EnTrance*. Sono state riportate solo le prime 30 risonanze contro le 101 dell’analisi.

8	575.968	-37.681	0.0604
9	638.998	-33.910	0.261
10	672.193	-37.909	0.353
11	702.549	-33.163	1.48
12	709.759	-34.749	1.5
13	710.500	-34.974	0.218
14	925.981	-34.352	0.618
15	1030.423	-39.769	0.35
16	1105.017	-39.118	0.371
17	1106.057	-35.744	0.347
18	1109.618	-23.379	7.45
19	1186.349	-37.485	0.483
20	1195.337	-30.824	0.926
21	1212.596	-36.837	0.276
22	1214.951	-33.790	0.362
23	1238.119	-32.111	0.827
24	1239.957	-33.148	0.484
25	1257.367	-39.146	0.39
26	1266.351	-33.476	0.482
27	1276.613	-37.616	0.337
28	1311.747	-36.124	0.446
29	1321.248	-28.722	0.341
30	1331.389	-33.557	0.295

Così come esiste il modulo `addsyn-read` per l'analisi con *AudioSculpt*, per acquisire questo tipo di dati – solitamente salvati con estensione '.ll' – è stato progettato il modulo 'llmod-read'. Il modulo contempla in ingresso esclusivamente il nome del file. Inoltre, i dati contenuti nel file devono essere espressi come liste in linguaggio LISP sulla base della seguente sintassi:

```
(setq l-freqf '(n n n n n))
(setq l-amplf '(n n n n n))
(setq l-bandf '(n n n n n))
```

dove `freqf` definisce la frequenza, `amplf` l'ampiezza e `bandf` la larghezza di banda.

Fausto Romitelli inizia la sua esperienza all'IRCAM in questo periodo di grande fermento, in cui *équipes* di sviluppatori fornivano costantemente nuovi strumenti informatici ad uso, pratico e artistico, dei compositori. Nei prossimi capitoli si affronterà nel dettaglio questa esperienza, nel percorso che porterà Romitelli dal *Cursus international de composition et d'informatique musicale* alla sua permanenza all'IRCAM come *Compositeur en recherche*, fino alla composizione dell'opera che può esser considerata il punto d'approdo di questa fase: *EnTrance* (1995)

CAPITOLO 3: FAUSTO ROMITELLI ALL'IRCAM

3.1 La partecipazione al *Cursus international de composition et d'informatique musicale*

3.1.1 Verso la Francia

Fausto Romitelli, compositore nato a Gorizia nel 1963, dopo essersi diplomato in composizione al conservatorio Giuseppe Verdi di Milano sotto la guida di Umberto Rotondi, frequentò alla fine degli anni Ottanta corsi di composizione con Franco Donatoni, all'Accademia Chigiana di Siena così come alla Civica scuola di musica di Milano. Tra i motivi che spinsero il compositore in Francia a partire dal 1990, l'interesse verso le tecniche compositive sviluppate dai compositori del gruppo dell'*Itinéraire* riveste certamente un ruolo centrale. Altri motivi, come sottolinea Alessandro Arbo, sono da ricercarsi «nelle maggiori opportunità offerte alla creazione musicale dalle politiche culturali di questo paese»,¹³³ che aveva portato molti giovani compositori a lasciare l'Italia. Infine, va segnalato l'interesse verso l'impiego dell'informatica nell'analisi e nella sintesi del suono. Romitelli considera infatti le nuove tecnologie come fondamento irrinunciabile nel panorama musicale contemporaneo. Nel 2000 scriverà:

Le nuove tecnologie hanno sconvolto le basi del pensiero musicale: sono il punto di arrivo di un lungo processo verso il controllo assoluto del suono e l'emancipazione del rumore. Se la fabbricazione e la pratica degli strumenti dell'orchestra si sono sviluppate nella necessità di rendere il suono il più armonico possibile e di ridurre al minimo le componenti di rumore, le nuove tecnologie musicali hanno aperto, invece, le porte della percezione all'universo inarmonico e ci hanno fornito gli strumenti per esplorare questo mondo inaudito. Le tecnologie non hanno generato un nuovo linguaggio, ma hanno suggerito ai compositori nuove interpretazioni di uno stesso principio: comporre 'il suono' anziché 'con il suono'. Il problema principale è quello di rendere personale l'uso delle

¹³³ ALESSANDRO ARBO, "Risonanze e anamorfofi. Note sulla poetica di Fausto Romitelli", in Amalia Collisani, Gabriele Garilli, Gaetano Mercadante (a cura di), *Italia/Francia. Musica e cultura nella seconda metà del XX secolo*, Palermo, L'Epos, 2009, p. 328.

macchine, funzionale cioè alle proprie necessità creative, affinché esse non divengano un ulteriore strumento d'omologazione imponendo sonorità e trattamenti standardizzati.¹³⁴

La ricchezza formativa rappresentata dal *Cursus international de composition et d'informatique musicale* dell'IRCAM non poteva lasciare indifferente il compositore. La sua partecipazione al *cursus* 1990-1991 segna l'inizio di un percorso quinquennale che, anche grazie all'incontro con Hugues Dufourt e Gérard Grisey, segnerà una svolta nella maturazione del suo linguaggio compositivo.

3.1.2 L'istituzione del Cursus

Il ritmo frenetico nel susseguirsi di nuovi strumenti informatici, come suggerisce Omer Corlaix,¹³⁵ ha portato ad uno stravolgimento delle dinamiche di trasmissione del sapere. Il rapporto maestro-allievo, su cui si era fino ad allora basato l'insegnamento della musica, era stato messo in secondo piano a favore di una didattica, idealmente neutra, incentrata sulle specificità e le caratteristiche del sistema in uso. L'istituzione all'IRCAM nel 1990 del *Cursus international de composition et d'informatique musicale* muove dalla volontà di recuperare quegli aspetti più prettamente pedagogici, declinati attorno al rapporto maestro-allievo, e di orientarli all'insegnamento delle nuove tecnologie e al loro utilizzo nel contesto di un'opera. In questa didattica rinnovata, i compositori hanno potuto passare da un tipo di apprendimento finalizzato alla comprensione del sistema in sé al suo utilizzo in funzione dell'idea musicale. Si voleva così rafforzare la figura del compositore-ricercatore che nei decenni precedenti aveva lasciato spazio a quella del compositore-ingegnere.

Nel 1990 il *cursus*, destinato ai compositori già in possesso di diploma di conservatorio o comunque ottenuto presso istituzioni di alta formazione musicale (università, *hautes écoles* ecc.), era strutturato in modo intensivo. Da ottobre a maggio venivano trattate sia da un punto di vista teorico che pratico le seguenti tematiche: acustica, psicoacustica, scienze cognitive, elaborazione digitale del segnale, formalizzazione delle strutture musicali, iniziazione all'informatica e alla programmazione, analisi, elaborazione e sintesi del suono, analisi e composizione assistita al computer, composizione. Giugno e luglio erano dedicati alla realizzazione di un brano personale che poi sarebbe stato presentato in forma di

¹³⁴ FAUSTO ROMITELLI, "Il compositore come virus", in Alessandro Arbo (a cura di), *Il corpo elettrico. Viaggio nel suono di Fausto Romitelli*, Monfalcone, Teatro comunale di Monfalcone, 2003, p. 82.

¹³⁵ OMER CORLAIX, "Le Cursus de composition et d'informatique musicale à l'Ircam", in «Résonance», n. 14, octobre 1998, p. 12.

concerto la primavera seguente. Il corso, tutt'ora attivo con cadenza annuale, si è nel tempo affermato come realtà d'eccellenza nella formazione musicale e vi si iscrivono musicisti provenienti da tutto il mondo.

3.2 *Natura morta con fiamme*

Esito del periodo di formazione di Romitelli al *cursus* fu la composizione di *Natura morta con fiamme* (1991), per quartetto d'archi amplificato ed elettronica. Il brano è stato eseguito per la prima volta presso l'Espace de projection dell'IRCAM il 19 marzo 1992.

Il brano è composto da sette sezioni nettamente separate, l'ultima delle quali non prevede elettronica. La diffusione del suono avviene per mezzo di otto altoparlanti, disposti intorno alla sala, adibiti all'amplificazione del quartetto d'archi. Altri due altoparlanti riproducono la parte elettronica.

Il suono degli archi è snaturato dall'amplificazione. Questa amplificazione è ottenuta sia attraverso microfoni panoramici – che captano l'intero quartetto – sia attraverso microfoni a contatto posizionati vicino al ponticello di ogni strumento. Nel secondo caso, il suono catturato risulta molto più 'rumoroso', meno 'puro', più 'distorto' di quanto ottenibile con un'amplificazione ordinaria. Il microfono al ponticello, infatti, sacrifica le risonanze della cassa acustica a favore dei transitori di attacco.

3.2.1 *Spazializzazione*

Nel programma di sala di *Natura morta con fiamme* si legge:

Si le timbre intègre lui-même les notions traditionnelles d'harmonie et de rythme, l'espace est la mise en perspective du timbre : l'écriture établit un système de relations-oppositions entre localisation et diffusion, surface et transparence ; c'est de la même façon que médium coloré et objet s'influencent mutuellement, l'un changeant la perception de l'autre.¹³⁶

¹³⁶ FAUSTO ROMITELLI, programma di sala di *Natura morta con fiamme*, <<http://brahms.ircam.fr/works/work/11521/#program>>, ultima visita 5 marzo 2017.

La scrittura musicale, come anche dichiarato da Romitelli nella presentazione della prima esecuzione,¹³⁷ è pensata per lo spazio, uno spazio frammentato, esplosivo, in cui i suoni che passano attraverso i diversi altoparlanti creano una sorta di moto vorticoso nella sala.

Ad ogni sezione dell'opera corrisponde una differente configurazione spaziale: ogni strumento è veicolato di volta in volta verso altoparlanti differenti grazie all'impiego di una matrice di spazializzazione. Ciò consente a Romitelli di costruire delle 'immagini' sonore differenti, che possono essere localizzate in modo puntuale – nel caso di un suono veicolato su di un singolo altoparlante – oppure diffuso – nel caso di due o più altoparlanti. La staticità della spazializzazione interna alle singole sezioni è compensata, come aveva già fatto Marco Stroppa in *Spirali*, dal movimento illusorio ottenuto attraverso l'artificio della scrittura musicale: «se facciamo suonare la stessa nota, alla stessa altezza, ad ogni strumento in successione, sentiremo la stessa nota che si muove, donando così l'illusione di uno stesso suono che si muove nello spazio. Noi sappiamo però che si tratta di quattro suoni diversi, suonati da strumenti diversi, che sono indirizzati nello spazio della sala».¹³⁸ Questa tecnica è storicamente consolidata, a maggior ragione se escludiamo l'ausilio dell'amplificazione a favore della ricerca sulla disposizione delle sorgenti acustiche: Romitelli stesso porta come esempi i cori battenti nella musica di Andrea e Giovanni Gabrieli e, più recentemente, *Gruppen* di Karlheinz Stockhausen o *Samuel Beckett: What is the Word* di György Kurtág.

Nel caso di *Natura morta con fiamme*, data una figura musicale affidata ad esempio al violino II, essa può passare successivamente al violino I, poi al violoncello e alla viola. Ampiezza, frequenza e durata sono qui leggermente differenti così da creare una sorta di eco, l'illusione di un singolo suono che rimbalza all'interno della sala e che, attraversando lo spazio, modifica le sue proprietà costitutive. La traiettoria di questo percorso è rappresentata in figura 28.

¹³⁷ FAUSTO ROMITELLI, presentazione di *Natura morta con fiamme*, <http://medias.ircam.fr/embed/media/x371b59_presentation-de-loeuvre-natura-morta-con>, ultima visita 5 marzo 2017.

¹³⁸ FAUSTO ROMITELLI, presentazione di *Natura morta con fiamme*, in occasione del festival Agora 2000. Trascrizione e traduzione dello scrivente. <http://medias.ircam.fr/x28breo_natura-morta-con-fiamme-presentation>, ultima visita 5 marzo 2017.

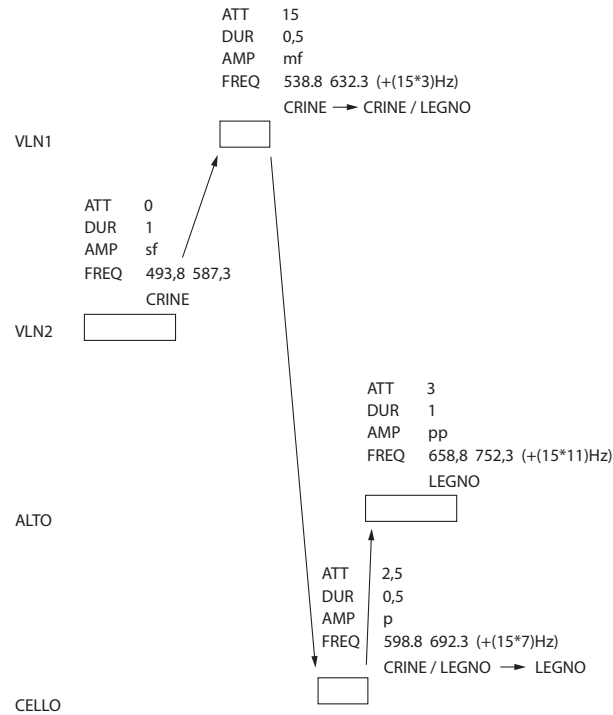


Figura 28 – Spazializzazione e trasformazione attraverso la scrittura musicale. Sezione F.¹³⁹

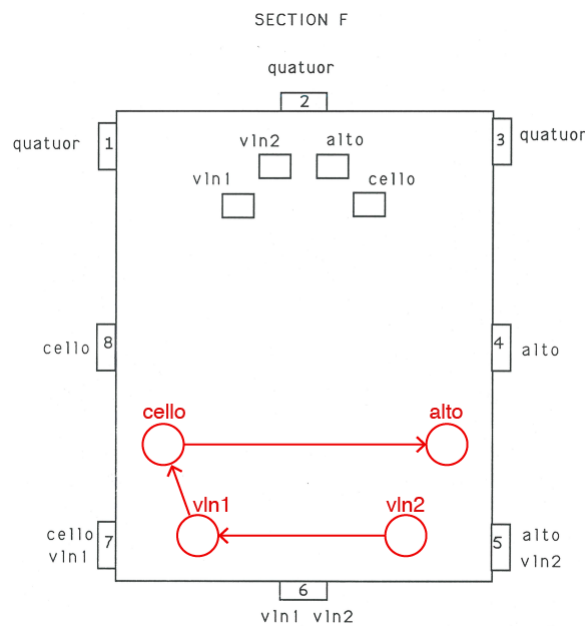


Figura 29 – Schema di diffusione per la sezione F. In rosso, la simulazione della traiettoria spaziale descritta in figura 28.¹⁴⁰

¹³⁹ Esempio tratto da *Pertinence du timbre*, uno scritto inedito riportato in appendice B.

¹⁴⁰ Schema di diffusione conservato presso l'IRCAM.

3.2.2 Elettronica

L'elettronica è realizzata interamente con suoni di sintesi, ossia ogni suono è stato generato a computer. Romitelli evita espressamente¹⁴¹ di elaborare suoni concreti, di fatto rinunciando ad ogni possibile fusione tra parte elettronica e parte strumentale. Il contrasto le dimensioni è dirimpante: l'elettronica, artificiale, è molto lontana da qualsiasi indice percettivo che possa richiamare una dimensione acustica. Più che l'integrazione, Romitelli ricerca tutte le possibili collisioni tra le parti. La stessa diffusione dell'elettronica all'interno della sala riceve un trattamento diverso da quello previsto per il quartetto, quasi a volerne marcare la distanza anche in termini spaziali: la sua diffusione, fissa, attraverso i soli due altoparlanti sul fronte del palco si contrappone al moto vorticoso della parte strumentale.

L'utilizzo di suoni artificiali, sintetici, così diversi dai suoni acustici degli strumenti, è giustificato dal compositore in questi termini:

Credo che tutta la mia generazione abbia un'esperienza dell'ascolto completamente mutata. Vale a dire che l'ambiente che ci circonda è costituito, a livello sonoro, da suoni che non sono quasi mai suoni naturali: i rumori della televisione, della radio, sono suoni filtrati, ossia mediati dagli altoparlanti, sono distorti. La nostra esperienza di suono non è quindi più legata al segnale 'naturale' ma piuttosto al segnale 'distorto'. Per me, ad esempio, il suono elettronico, il suono artificiale, da un certo punto di vista risulta più naturale di quello di un quartetto d'archi, e quindi è più artificiale lavorare con strumenti acustici rispetto all'elettronica.¹⁴²

3.2.3 Il programma WORKLISP

Lavorare con l'elettronica impone di interfacciarsi con sistemi di sintesi e di controllo del suono. In *Natura morta con fiamme* i suoni sono stati ottenuti per mezzo di *Csound*. *Csound* permette la sintesi a partire da operatori semplici, quali ad esempio oscillatori o filtri, a loro volta descrivibili per mezzo di più parametri (inizio, fine, evoluzione dell'ampiezza e della frequenza nel tempo, fase, larghezza di banda ecc.). Per generare suoni complessi è quindi necessario gestire una notevole quantità di dati. Declinando il principio di «rendere personale l'uso delle macchine, funzionale cioè alle proprie necessità creative», espresso anni più tardi nello scritto *Il compositore come virus*, Romitelli ha superato questa difficoltà

¹⁴¹ FAUSTO ROMITELLI, presentazione di *Natura morta con fiamme*, in occasione del festival Agora 2000.

¹⁴² FAUSTO ROMITELLI, presentazione di *Natura morta con fiamme*, in occasione del festival Agora 2000. Trascrizione e traduzione dello scrivente.

sviluppando un programma di controllo scritto in LISP da lui chiamato *WORKKLISP*. Attraverso la definizione di funzioni specifiche, il programma permette di ridurre sensibilmente la quantità di dati da manipolare e al tempo stesso di creare delle categorie sulla base del tipo di suono da ottenere.

Una versione del programma è stata rinvenuta presso gli archivi dell'IRCAM, conservata da Laurent Pottier all'interno del materiale utilizzato per un'altra composizione di Romitelli, *EnTrance* (1995), di cui è stato assistente musicale. Questo programma ha sicuramente subito delle modifiche tra il 1991 e il 1995 anche per mano dello stesso Pottier,¹⁴³ e quindi non può porsi come fonte autentica. Ma anche ammesso di ritrovare una versione di *WORKKLISP* non modificata, il solo programma non consentirebbe di far luce sull'intero processo creativo. A tale scopo, si dovrebbe risalire ai parametri della sintesi utilizzati per generare i singoli suoni. Inoltre, in partitura si registra l'assenza di una qualsivoglia forma simbolica di notazione della parte elettronica. Senza tali dati, lo studio deve pertanto ancorarsi ad approcci differenti, come già proposto ad esempio da Eric Maestri che ha utilizzato il metodo dell'analisi aurale.¹⁴⁴ Ciononostante, *WORKKLISP* rappresenta una fonte preziosa per indagare il *modus operandi* di Romitelli per la composizione della parte elettronica di *Natura morta con fiamme* e, più in generale, per la ricerca sulla sintesi in *Csound*.

Il programma è costituito da due file di testo: *lispdef* e *WORKKLISP*. Il primo conserva un gran numero di funzioni in LISP e costituisce le fondamenta del programma; il secondo è invece progettato per 'valutare' le funzioni definite da *lispdef* e generare i file *orchestra* per *Csound*.

Nel file *WORKKLISP* è possibile riconoscere almeno otto diverse tipologie di sintesi:

- Sintesi additiva semplice;
- sintesi additiva con formanti;
- sintesi sottrattiva da rumore;
- sintesi sottrattiva da fonte periodica;
- filtraggio da un suono dato;

¹⁴³ I metadati dei file riportano come data di creazione novembre 1991 e come data di ultima modifica luglio 1995. Ai commenti in lingua italiana si alternano commenti in lingua francese. In qualche caso, inoltre, i percorsi di salvataggio dei file per *Csound* fanno riferimento al computer personale di Pottier.

¹⁴⁴ ERIC MAESTRI "Natura morta con Fiamme de Fausto Romitelli. La musique mixte et l'analyse aurale", in *Corpus et méthodes : traductions théoriques de l'hétérogénéité musicale*, atti del colloquio internazionale Journées d'Analyse Musicale 2013 de la SFAM, Université de Rennes II, Sampzon, Delatour, 2013. <https://www.academia.edu/4460470/_Natura_morta_con_fiamme_de_Fausto_Romitelli._La_musique_mixte_et_la_analyse_aurale_>, ultima visita 5 marzo 2017.

- sintesi additiva con modulatore d'ampiezza;
- modulazione di frequenza;
- modulazione di frequenza con formanti.

Di queste, almeno due (sintesi sottrattiva da rumore e filtraggio da un suono dato) sono state modificate da Pottier.

Di seguito si riportano alcuni esempi, commentati, del funzionamento del programma. In LISP, il simbolo 'punto e virgola' indica un commento e pertanto l'intera riga non viene processata.

3.2.3.1 Esempio 1. *lispdef*: definizione di uno spettro armonico con n parziali e un fattore r:

```
;;; Definizione della funzione spectrum.
;;; Argomenti della funzione:
;;; f0 = frequenza fondamentale
;;; r = fattore moltiplicativo
;;; nmax = numero di parziali
(defun spectrum (f0 r nmax)

;;; inizializzazione la variabile locale l
(let ((l nil))

;;; indicizzazione di i tramite iterazione, da nmax a 1.
(do ((i nmax (- i 1)))
    ((<= i 0)l)

;;; costruzione della lista l con il calcolo dello spettro
(setq l(cons(* f0 r i)l))))
```

La valutazione della funzione seguente definirebbe uno spettro armonico a partire dal La a 110 Hz se r fosse pari a 1, ma in questo caso lo spettro è trasposto di un'ottava all'acuto grazie al fattore $r = 2$. L'output della funzione è quindi uno spettro armonico con dieci parziali e con fondamentale sul La a 220 Hz:

```
(spectrum 110 2 10)

(220 440 660 880 1100 1320 1540 1760 1980 2200)
```


3.2.3.2 Esempio 2. *lispdef*: interpolazione complessa tra due liste di valori.¹⁴⁵

In questo esempio, la sintassi non prevede l'utilizzo di operatori base, fatta eccezione per la dichiarazione della funzione (*defun*). Essa piuttosto richiama altre funzioni da cui dipende, con i rispettivi argomenti. Le funzioni *popcopy*, *qinterpxx1*, *scopy*, *interplus0* sono riportate in appendice C.1.

```
;;; Definizione della funzione interpmax0
;;; Argomenti della funzione:
;;; rk = passi interpolazione tra i valori della singola serie.
;;; qk = tipo di interpolazione interna alla serie.
;;;      Se qk = 1, interpolazione lineare.
;;; r = passi interpolazioni tra le due serie
;;; q = tipo di interpolazione tra le serie.
;;;      Se q = 1, interpolazione lineare.
(defun interpmax0 (list1 list2 rk qk r q)
  (popcopy(qinterpxx1
    (scopy(interplus0 list1 rk qk)
      (interplus0 list2 rk qk))
    r q)))
```

Si considerino ad esempio le seguenti due liste da interpolare

```
a1 (100 200 300 400 500)
a2 (110 215 320 425 530)
```

e la valutazione della funzione

```
(interpmax0 a1 a2 1 1 2 1)
```

In questo caso, la valutazione restituirà due diverse interpolazioni: una tra gli elementi delle singole liste (*rk* = 1, un passo) e una tra le due liste (*r* = 2, due passi). Le interpolazioni sono lineari (*qk* = 1, *q* = 1).

```
((100 150 200 250 300 350 400 450 500)
 (310/3 925/6 205 1535/6 920/3 715/2 1225/3 2755/6 510)
 (320/3 475/3 210 785/3 940/3 365 1250/3 1405/3 520)
 (110 325/2 215 535/2 320 745/2 425 955/2 530))
```

¹⁴⁵ Interpolazioni di questo tipo sono frequentemente utilizzate da Romitelli, sia in termini frequenziali tra spettri diversi, come lo stesso compositore afferma nella presentazione dell'opera nel 2000, sia tra involucri e traiettorie delle diverse parziali di un suono.

3.2.3.3 Esempio 3. Interpolazione complessa: creazione di più involucri¹⁴⁶

Questa funzione è più complessa, in quanto è stata progettata appositamente per essere valutata dal file *WORKLISP*: essa dà la possibilità di generare i parametri per l'involucro di n componenti frequenziali, ad esempio per la sintesi additiva. Lo script genera 20 diversi involucri per *Csound*.

In questo caso, l'argomento *k* comunica a *Csound* che si tratta di un involucro (*kenv*), *q* indica il tipo di 'curva' (esponenziale, *expseg*), mentre le restanti sette liste definiscono i parametri delle interpolazioni da affidare a ciascuna parziale.

- Definizione della funzione *iperqenvelope*:

```
(defun iperqenvelope (k q list1 list2 list3 list4 list5 list6 list7
nom-fichier)
(list
  (iperqenv k q list1 list2 list3 list4 list5 list6 list7 nom-
fichier)
  (endinterpline0 k
    (car(reverse list1)) q
    (molt list2 (car(reverse list3)))
    0 1
    (molt list5 (car(reverse list6)))
    (car(reverse list7)) nom-fichier))
```

- *WORKLISP*. Valutazione della funzione *iperqenvelope*:

```
(iperqenvelope
;;; involucro: k
'kenv
;;; tipo curva: q
'expseg
;;; numero sezioni parziali: list1
'(1 10 20)
;;; modello involucro ampiezza generale: list2
'(1 20 1)
;;; rapporto involucro parziali/involucro generale: list3
'(.5 1 .2)
;;; tipo di interpolazione tra le parziali: list4
'(1 1 1)
;;; modello evoluzione temporale generale: list5
'(0 2 4)
;;; rapporto evoluzione parziali/evoluzione generale: list6
'(1 .8 .6)
;;; tipo di interpolazione nel tempo: list7
'(1 1 1)
```

¹⁴⁶ Come nel caso precedente, anche qui la funzione dipende da altre funzioni riportate in appendice C.

nom-fichier)

- Istruzioni per *Csound* generate da *WORKLISP*:

```
KENV1 EXPSEG 0.5, 2.0, 10.0, 2.0, 0.5,.01,0.00001
KENV2 EXPSEG 0.555556, 1.955556, 11.11111, 1.955556, 0.555556,.01,
0.00001
KENV3 EXPSEG 0.611111, 1.911111, 12.22222, 1.911111, 0.611111,.01,
0.00001
KENV4 EXPSEG 0.666667, 1.866667, 13.33333, 1.866667, 0.666667,.01,
0.00001
KENV5 EXPSEG 0.722222, 1.822222, 14.44444, 1.822222, 0.722222,.01,
0.00001
KENV6 EXPSEG 0.777778, 1.777778, 15.55556, 1.777778, 0.777778,.01,
0.00001
KENV7 EXPSEG 0.833333, 1.733333, 16.66667, 1.733333, 0.833333,.01,
0.00001
KENV8 EXPSEG 0.888889, 1.688889, 17.77778, 1.688889, 0.888889,.01,
0.00001
KENV9 EXPSEG 0.944444, 1.644444, 18.88889, 1.644444, 0.944444,.01,
0.00001
KENV10 EXPSEG 1.0, 1.6, 20.0, 1.6, 1.0,.01,0.00001
KENV11 EXPSEG 0.92, 1.56, 18.4, 1.56, 0.92,.01,0.00001
KENV12 EXPSEG 0.84, 1.52, 16.8, 1.52, 0.84,.01,0.00001
KENV13 EXPSEG 0.76, 1.48, 15.2, 1.48, 0.76,.01,0.00001
KENV14 EXPSEG 0.68, 1.44, 13.6, 1.44, 0.68,.01,0.00001
KENV15 EXPSEG 0.6, 1.4, 12.0, 1.4, 0.6,.01,0.00001
KENV16 EXPSEG 0.52, 1.36, 10.4, 1.36, 0.52,.01,0.00001
KENV17 EXPSEG 0.44, 1.32, 8.8, 1.32, 0.44,.01,0.00001
KENV18 EXPSEG 0.36, 1.28, 7.2, 1.28, 0.36,.01,0.00001
KENV19 EXPSEG 0.28, 1.24, 5.6, 1.24, 0.28,.01,0.00001
KENV20 EXPSEG 0.2, 1.2, 4.0, 1.2, 0.2,.01,0.00001
```

3.2.3.4 Esempio 4. *WORKLISP*: strumento FM

Si riporta lo strumento FM, basato sulla modulazione di frequenza. Dopo le dichiarazioni iniziali, vengono valutate le seguenti funzioni:

- **ENTER** imposta frequenza di campionamento per segnali audio (**sr**) e segnali di controllo (**kr**), così come il numero di canali audio (**nchnls**);
- **NINSTR** va a scrivere il numero dello strumento e il valore massimo d'ampiezza;
- **IPERQENVELOPE**, **LINE** e **SUPERQENVELOPE** controllano i diversi parametri nel tempo, rispettivamente: inviluppo, indice di modulazione, e scostamento in frequenza;
- **FOSCILI** gestisce il rapporto tra frequenza portante e frequenza modulante;
- **EXIT** istruisce sul segnale in uscita (**OUT**) del sintetizzatore.

In questo esempio si sono conservati i commenti redatti con ogni probabilità da Romitelli.

```
(with-open-file
(nom-fichier "Macintosh HD:Users:FR:getStarted:lorc.orc"
:direction :output :if-exists :overwrite
:if-does-not-exist :create)
;;;;;;;;;;;;;
(ENTER 48000 1000 1
nom-fichier)

(NINSTR
;;; ninstr
1
;;; max ampiezza
100
;;;-nparz.
20
nom-fichier)

(IPERQENVELOPE
'kenv
;;; tipo curva
'expseg
;;; numero sezioni parziali
'(1 2 3)
;;;-modello env amp
'(1 20 1)
;;; rapp sez parz con env amp
'(.5 1 .2)
;;; q verticale amp
'(1 1 1)
;;; mod env time
'(0 6 12)
;;; rapp sez parz con mod env time
'(1 .8 .6)
;;; q verticale time
'(1 1 1)
nom-fichier)

(LINE
'kindex 1
;;; tipo curva
'expseg
;;; env indice fm
'(10 1 40)
;;; env time indice.
'(0 0.5 12)
nom-fichier)

(SUPERQENVELOPE
'kport
;;; tipo curva
'expseg
;;; numero sezioni parziali
```

```
'(1 2 )
;;; modello env freq
'(100 261 261)
;;; rapp sez parz con env freq
'(1.028 1.414)
;;; q verticale freq
'(1 1 )
;;; mod env time
'(0 6 12)
  nom-fichier)

(FOSCILI
;;; ratio portante/modulante
'(1 1.12 )
nom-fichier )

(EXIT
;;; tipo out
'out
;;; numero out
'(3 )
;;; evaluate.
  nom-fichier))
```

L'output, il file *orchestra* di *Csound*, è il seguente:

```
sr = 48000
kr = 1000
ksmps = 48
nchnls = 1

instr 1
iscaling=32767/(100*20)

KENV1  EXPSEG    0.5,    6.0,    10.0,    6.0,    0.5,.01,0.00001
KENV2  EXPSEG    1.0,    4.8,    20.0,    4.8,    1.0,.01,0.00001
KENV3  EXPSEG    0.2,    3.6,    4.0,    3.6,    0.2,.01,0.00001

KINDEX1 EXPSEG    10.0,    0.5,    1.0,    11.5,    40.0
KPORT1  EXPSEG    102.8,    6.0, 268.308,    6.0, 268.308
KPORT2  EXPSEG    141.4,    6.0, 369.054,    6.0, 369.054

a1 foscili kenv1*iscaling,1,kport1,kport1*    1.0,kindex1,1
a2 foscili kenv2*iscaling,1,kport2,kport2*    1.12,kindex1,1
aout1=0.0+a1+a2+a3

asortiel=0.0+aout1
OUT  (0.0+asortiel)
endin
```

Il suono può essere sintetizzato (figura 30) anche con le più recenti versioni di *Csound*,¹⁴⁷ a patto di aggiungere il seguente *score*:

```
f 1 0 1024 10 1      ;definizione della tabella dell'oscillatore
i 1 0 12             ;numero, inizio e durata dello strumento
e                   ;chiusura del processo
```

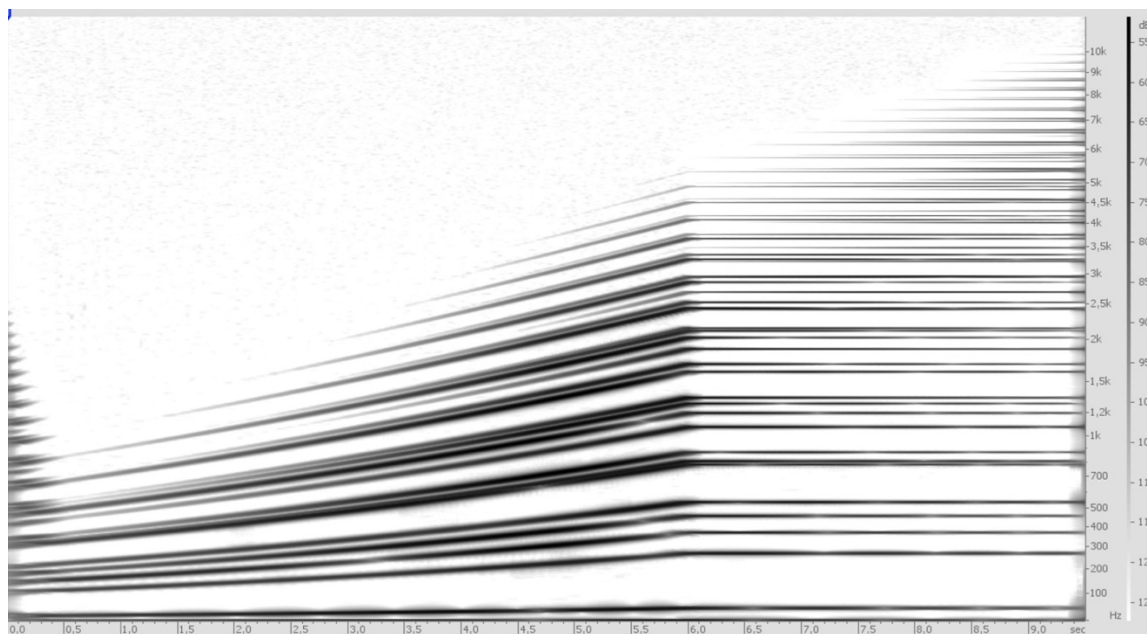


Figura 30 – Sonogramma del suono ottenuto per modulazione di frequenza con Csound. Si notino: a) il repentino decremento dell'indice di modulazione, seguito da un suo progressivo incremento; b) la progressiva variazione in frequenza che poi si annulla a 0:06. Analisi effettuata con iZotope RX 4, multiresolution, con finestra base di 4096 punti.

¹⁴⁷ Le moderne versioni di *Csound* sono case-sensitive. Per la sintesi del suono si sono dovuti riscrivere in carattere minuscolo i comandi originariamente in maiuscolo.

3.2.4 Patchwork?

Il programma di sala della prima esecuzione dell'opera, sempre conservato all'IRCAM, riporta un'informazione aggiuntiva, non rientrata nel testo poi utilizzato dall'editore Ricordi:

Les sons de la bande digitale ont été entièrement réalisés par le programme de synthèse Csound sous le contrôle d'un programme en Lisp écrit par le compositeur et de l'environnement d'aide à la composition "PatchWork".¹⁴⁸

In assenza di altre fonti, non è possibile stabilire con certezza quale fosse il reale impiego di *Patchwork* nella composizione. Considerato però che tutte le informazioni inerenti la parte elettronica, così come le testimonianze orali del compositore,¹⁴⁹ parlano esclusivamente del programma *WORKLISP*, e che il programma è effettivamente in grado di generare gli strumenti per *Csound*, è lecito supporre che *Patchwork*, ancora in fase sperimentale, possa esser stato impiegato da Romitelli non tanto per l'elettronica ma piuttosto come aiuto per composizione della parte strumentale. Lo stretto legame che intercorre tra LISP e *Patchwork* garantiva una transizione pressoché immediata tra i due sistemi. Inoltre *Patchwork* riusciva a tradurre e visualizzare in notazione tradizionale valori frequenziali, facilitando la fase di scrittura della partitura.

3.3 Nuove prospettive compositive

3.3.1 Verso un'organizzazione del timbro

La riflessione sul timbro che ha contraddistinto gli ultimi decenni del Novecento ha portato da un lato alla maturazione dell'estetica spettrale ascrivibile ai compositori del gruppo

¹⁴⁸ FAUSTO ROMITELLI, programma di sala di *Natura morta con fiamme*, Parigi, IRCAM, 1992. Il programma è redatto in occasione della prima esecuzione assoluta dell'opera, il 19 marzo 1992, nell'ambito dell'Atelier Pédagogique all'IRCAM. L'informazione è inoltre presente nel contributo di ERIC MAESTRI "Natura morta con Fiamme de Fausto Romitelli. La musique mixte et l'analyse aurale", al quale si rinvia per un'analisi aurale della parte elettronica.

¹⁴⁹ FAUSTO ROMITELLI, presentazione di *Natura morta con fiamme*, in occasione del festival Agora 2000,

dell'itinéraire, dall'altro a una sua integrazione all'interno del discorso musicale proposta nel 1985 da Boulez.¹⁵⁰

Per Boulez, infatti, il timbro acquista valenza estetica in quanto legato alla costituzione dell'oggetto musicale. Così come il suono di per sé, seppur contraddistinto da una propria identità, non rappresenta un fenomeno estetico se estrapolato dal linguaggio musicale, anche il timbro deve essere strutturato attraverso un sistema di relazioni per poter essere organizzato nella pratica compositiva:

The functional possibilities of timbre only seem valid if they are linked to language and to the articulation of a discourse through structural relationships; timbre both explains and masks at the same time. Without musical discourse it is nothing, but it can also form the entire discourse on its own. [...] I would underline that timbre, through composition, should integrate itself totally with musical language in a multidimensional world where its specificity will be the measure of its importance.¹⁵¹

La necessità di individuare relazioni strutturali nell'organizzazione del timbro ha spinto i compositori a ricercare una nuova categorizzazione in grado di ristabilire quel sistema gerarchico che era stato abolito dal serialismo.¹⁵² A suo modo, l'insegnamento di Boulez fu assimilato anche da Fausto Romitelli, che nei primi anni Novanta elaborerà una personale riflessione sul timbro.¹⁵³

¹⁵⁰ L'articolo fu presentato nel 1985 in un seminario tenutosi all'IRCAM. È stato poi ripreso e tradotto in inglese in PIERRE BOULEZ, "Timbre and composition - timbre and language", in «Contemporary Music Review», v. 2 n. 1, 1987, pp. 161-171.

¹⁵¹ PIERRE BOULEZ, "Timbre and composition - timbre and language", pp. 170-171.

¹⁵² Per Kaija Saariaho, ad esempio, la storica contrapposizione tra consonanza e dissonanza viene declinata nel campo timbrico, delimitato da una personale interpretazione di concetti quali 'suono' e 'rumore': «the 'noise' in itself can actually manifest itself in different ways – soft, harsh etc. In a general way, the concept of "noise" signifies to me utterances such as breathing, the sound of a flute in a low register or a string instrument playing 'sul ponticello'. By contrast, a pure sound would be more akin to the ringing of a bell or a human voice singing in the Western tradition. The sound/noise axis exists as an abstraction which can be applied on different scales: it might be conveyed with a single violin bow, or by using all the instruments of an orchestra». KAIJA SAARIAHO, "Timbre and harmony: Interpolations of timbral structures", in «Contemporary Music Review», v. 2 n. 1, 1987, p. 94.

¹⁵³ Tra i primi esiti della ricerca sul timbro promossa dall'IRCAM va segnalata la raccolta di saggi del 1991 *Le timbre: métaphore pour la composition*, curata da Jean-Baptiste Barrière ed edita da Bourgois / IRCAM. Il testo, come sottolinea Arbo, era considerato da Romitelli una lettura fondamentale (ALESSANDRO ARBO, "Risonanze e anamorfofi. Note sulla poetica di Fausto Romitelli", p. 329).

3.3.2 Una personale riflessione sul timbro. L'esperienza come compositeur en recherche

Dopo aver seguito il *cursus*, Romitelli ritornò all'IRCAM dal 1993 fino alla fine del 1995 come *compositeur en recherche* all'interno dell'*équipe de representation musicale* guidata da Gérard Assayag. In questo periodo, conduce una personale ricerca su possibili criteri di organizzazione del materiale musicale. Romitelli mira a comprendere quel sistema di opposizioni che tradizionalmente esiste tra la concorrenza di entità simultanee e la concatenazione di entità successive, letto alla luce delle nuove scoperte in campo acustico e della riflessione sul fenomeno timbrico che in quegli anni si sviluppava all'IRCAM anche sulla spinta delle riflessioni teoriche di Boulez.

Già nella presentazione de *La sabbia del tempo* (1991-1992) e di *Les idoles du soleil* (1992), Romitelli scrive:

Gli oggetti evolvono in uno spazio di transizione tra armonia e timbro: le strutture timbriche sono legate ai processi di fusione percettiva di tutte le componenti in singole immagini sintetiche, quelle armoniche sono legate ai processi di separazione (parsing processes) che permettono la riconoscibilità di ogni singola componente intervallare. I processi di fusione e fissione percettiva sono legati all'interazione complessa di fattori quali l'armonicità, la sincronia nell'attacco, la coordinazione nella modulazione d'ampiezza e frequenza, la densità spettrale ecc.¹⁵⁴

In queste poche righe è possibile scorgere, da un lato, concetti quali 'soglie percettive', 'armonicità', 'modulazione', 'densità spettrale', che mettono in evidenza l'interesse nutrito verso il panorama della musica spettrale e verso le tecniche compositive sdoganate dai compositori dell'*Itinéraire*; dall'altro, 'strutture timbriche', 'oggetti', mostrano come Romitelli abbia accolto la riflessione maturata all'IRCAM.

Uno scritto inedito dal titolo *Pertinence du timbre*,¹⁵⁵ redatto tra il 1992 e il 1995 e conservato presso l'archivio privato di Laurent Pottier, fa chiarezza sullo sviluppo del pensiero compositivo che Romitelli maturava in quegli anni, dopo la composizione di *Natura morta con fiamme*. Vi si riportano alcuni esempi delle nuove tecniche compositive elaborate in particolare per la composizione di *Mediterraneo I – Les idoles du soleil* (1992).

¹⁵⁴ Presentazione della *Sabbia del tempo*, in ALESSANDRO ARBO, "Risonanze e anamorfofi. Note sulla poetica di Fausto Romitelli", pp. 341-342.

¹⁵⁵ Lo scritto è riportato in appendice B.

Nello scritto, Romitelli utilizza insistentemente il termine ‘aggregato’ per definire un gruppo accordale. La definizione di un insieme discreto di aggregati fornisce la base per una possibile categorizzazione timbrica. L'esempio portato da Romitelli è il seguente: il materiale di base è costituito da tre aggregati (figura 31), che differiscono per numero e conformazione degli intervalli, così come per densità e registro:

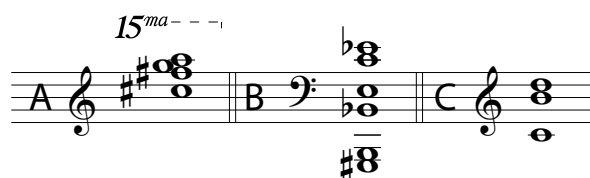


Figura 31 – I tre aggregati base.

Ogni aggregato è manipolato attingendo da un repertorio di regole combinatorie che gli consente di generare gli insiemi di oggetti A, B e C. Come si può notare in figura 31, le regole impiegate per trasformare gli aggregati base sono di volta in volta diverse:

- L'aggregato A viene sottoposto a un'elaborazione che in dodici passi lo porta dal sovracuto verso il grave. Da A₁ a A₃, così come da A₅ a A₆, ad ogni passo viene condotta al grave solamente una delle quattro note dell'aggregato, mentre le altre tre rimangono fisse; la transizione tra A₃ e A₄ è invece contraddistinta da una trasposizione di una nona minore di tutte le note dell'aggregato: A₃ e A₄ mantengono così invariate le relazioni intervallari interne; tra A₄ e A₅ assistiamo ad una compressione dei rapporti intervallari; infine, le transizioni viste finora sono ripresentate, in ordine invertito, da A₇ a A₁₂;
- le operazioni svolte sull'aggregato B conducono alla generazione di altri quattro aggregati, ognuno dei quali è calcolato a partire da B₁. Il procedimento si basa sull'inversione di un numero di intervalli via via crescente fino ad ottenere una completa riflessione in B₅;
- l'aggregato C genera dodici oggetti ottenuti esclusivamente tramite inversione degli intervalli e trasposizione.

Vengono a definirsi così tre distinte categorie, che comprendono l'aggregato iniziale e le sue trasformazioni combinatorie (figura 32).

The figure displays three systems of musical notation, labeled A, B, and C, each containing several aggregates. System A includes aggregates A1 through A12, with fingerings such as 2, 1, 5, 6, 8, 13, 5, 2, 6, 5, 8, 6, 13, 8, 6, 5, 1, 2. System B includes aggregates B1 through B5, with fingerings like 8, 6, 11, 3, 8, 11, 3, 8, 11, 3, 8, 11, 3, 8, 11, 3. System C includes aggregates C1 through C12, with fingerings such as 3, 11, 3, 11, 3, 11, 3, 11, 3, 11, 3, 11, 3, 11, 3, 11, 3. Lines connect notes between aggregates, and some aggregates have specific markings like 'A10' or 'C6'.

Figura 32 – Regole per la trasformazione degli aggregati.

Una volta esaurito il calcolo combinatorio per la generazione del nuovo materiale, Romitelli si interroga sui possibili percorsi tra le diverse categorie. Viene definito il seguente sistema di regole:

- La transizione tra due aggregati è possibile se ci sono almeno due intervalli consecutivi in comune, anche in inversione;
- analogamente, la transizione è possibile in presenza di almeno due altezze assolute in comune;
- ciascun aggregato definito dagli insiemi A, B e C non può legarsi a più di due altri aggregati. Nel caso in cui siano possibili più collegamenti, va rispettato l'ordine di apparizione del calcolo combinatorio;
- ogni insieme deve essere letto da sinistra a destra.

Esemplificando, B₁ può legarsi ad altri quattro aggregati (figura 33): A₃, A₄ (intervalli consecutivi in comune, in questo caso in inversione), così come a C₂ e C₁₀ (due note in comune).

Intervalli consecutivi Intervalli consecutivi Note in comune Note in comune

B1 A3 B1 A4 B1 C2 B1 C10

8 6 11 3 6 8 13 3 8 6 11 3 6 8 13 3

Figura 33 – Possibili collegamenti tra B_1 e gli aggregati degli insiemi A e C .

Se le regole imposte consentono di individuare oggetti sonori tra loro più o meno simili – percettivamente, il legame per note comuni è più forte rispetto a quello per intervalli comuni –, è altresì vero che per ampliare le possibilità compositive è necessario operare anche per contrasto. Regole per elementi in comune, per contrasto, o ancora situazioni intermedie permettono di strutturare la transizione tra gli aggregati delle diverse categorie. In questo modo, si viene a creare un sistema gerarchico che permette al compositore di realizzare una ‘curva di tensione’ nel dominio armonico.

Se finora le relazioni si sono ancorate ad un modello quasi esclusivamente intervallare, Romitelli muove un passo nella direzione dello spazio del timbro:

Nous allons les considérer pas plus comme des produits d’une logique combinatoire, des objets discrets constitués à partir de l’articulation réglée d’éléments abstraits et relationnels, mais comme des ensembles de fréquences qui tendent à la fusion perceptive dans des images synthétiques: des timbre placés à l’intérieur d’un espace multidimensionnel comme produit d’une transition entre modèles opposés.¹⁵⁶

Gli aggregati di ciascuna categoria sono quindi analizzati alla ricerca di un fondamentale implicito di cui le note dell’‘accordo’ possono essere considerate parziali. Per portare un esempio, le note di A_6 corrispondono alle parziali 5, 7, 8 e 11 di uno spettro armonico sul S_{11} (figura 34). Il risultato dell’operazione, applicata alle categorie A , B e C , è esposto in figura 35.

¹⁵⁶ *Infra*, appendice B. La multidimensionalità dello spazio timbrico di cui parla Romitelli si ancora ad un concetto espresso da Jean-Baptiste Barrière e che il compositore cita: “Un espace de timbres est déterminé à partir d’un choix de modèles. Ceux-ci sont distribués en fonction d’une catégorisation psychologique arbitraire projeté dans un espace à x dimensions”.

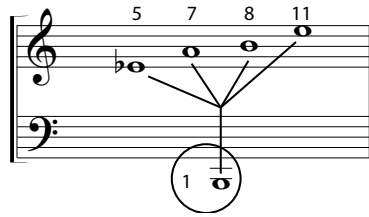


Figura 34 – Fondamentale implicito dell'aggregato A6.

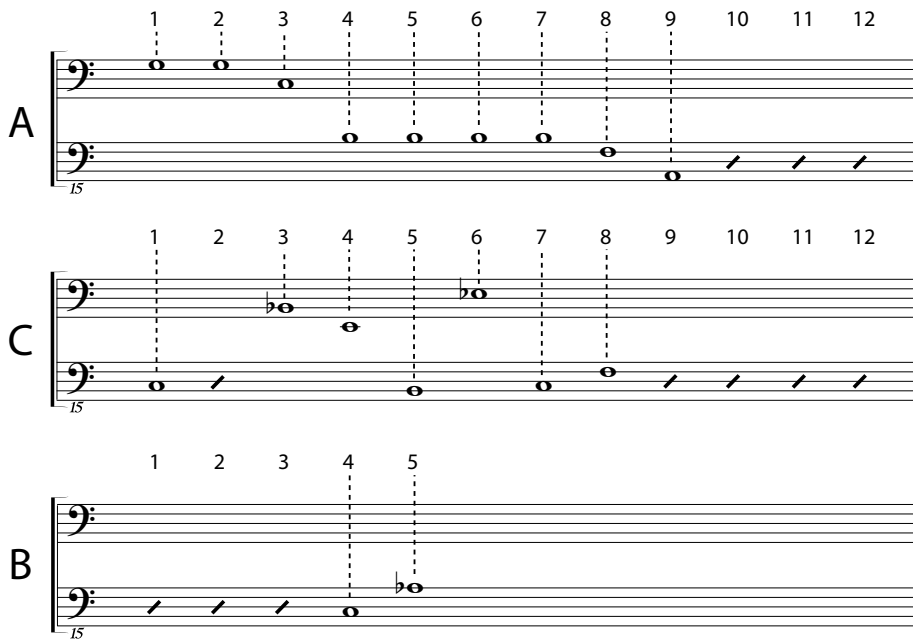


Figura 35 – Fondamentali impliciti degli aggregati.

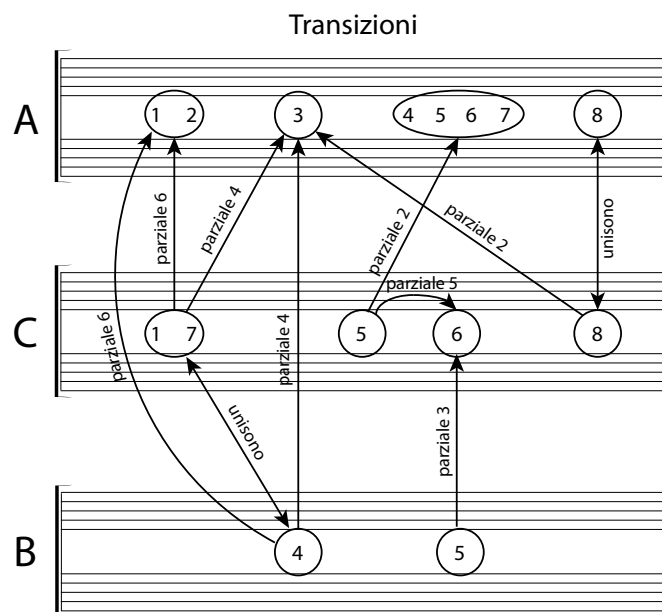


Figura 36 – Possibili transizioni individuate su base spettrale.

Ulteriori transizioni, di conseguenza, possono essere individuate su base spettrale: è possibile infatti tessere nuove relazioni quando le frequenze fondamentali implicite di due o più aggregati fanno parte di uno stesso spettro armonico immaginario (figura 36).

La categorizzazione del timbro ha indotto Romitelli a sviluppare un nuovo sistema in grado di guidare la formalizzazione delle strutture musicali. Il mezzo designato è inevitabilmente l'informatica musicale: «l'objectif est alors de déterminer des représentations informatiques dans lesquelles cette opposition peut être pensée et contrôlée.»¹⁵⁷

Come ausilio alla composizione, Romitelli sviluppa un software prototipale che gli permette di costruire lo spazio timbrico mettendo in relazione i diversi aggregati sonori e gli spettri. Il programma considerava quindi sia le relazioni intervallari, verticali o orizzontali, sia la struttura spettrale dei suoni:

A partir des paramètres fixés de l'un ou de l'autre type, le système construit un réseau de contraintes qu'il cherche à résoudre en produisant tous les objets compatibles (i.e. moments spectraux contenant les bons intervalles). Une interface *Patchwork* permet de diriger les solutions vers une représentation en notation musicale ou vers la génération de script de synthèse pour *Csound*.¹⁵⁸

Grazie alla CAO, la riflessione teorico musicale di Romitelli si traduce in codice LISP. Il file *lispdef*,¹⁵⁹ creato originariamente per la composizione di *Natura morta con fiamme*, si stratifica negli anni con nuove funzioni volte all'implementazione nell'ambiente informatico delle strategie compositive. I commenti 'autografi' sono illuminanti, ad esempio:

```

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;; DATA UNA LISTA DI LISTE D'INTERVALLI, LI TROVA ALL'INTERNO
;;; DI UNO SPETTRO, NELL'ORDINE STABILITO DALLA LISTA,
;;; E LI FA APPARIRE CANCELLANDO LE NOTE DELLO SPETTRO
;;; ESTRANEE AGLI INTERVALLI
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;; DATA UNA LISTA DI INTERVALLI, LI INTEGRA IN SPETTRI DISTORTI
;;; ABBASSANDO IL COEFF.; E' POSSIBILE DEFINIRE
;;; QUANTI DEGLI INTERVALLI DELLA LISTA DEVONO
;;; ESSERE OBBLIGATORIAMENTE PRESENTI NELLO
;;; SPETTRO (NI) E IN QUALE DENSITA' MINIMA (NN ) RISPETTO
;;; AL NUMERO COMPLESSIVO DEGLI INTERVALLI DELLO SPETTRO

```

¹⁵⁷ Rapport IRCAM 1993, p. 46.

¹⁵⁸ *Ibidem*

¹⁵⁹ Si veda *infra*, appendice C.

Vi si ritrovano: a) il legame tra intervalli e spettri; b) la gerarchia tra aggregati intervallari e la definizione dei percorsi possibili; c) la definizione delle regole di transizione. Tutti questi elementi, come si vedrà, saranno alla base della scrittura di *EnTrance*.

3.4 La collaborazione con Laurent Pottier

Nei primi anni Novanta, all'IRCAM si sviluppa la figura dell'assistente musicale, poi affermatosi come RIM (*réalisateur en informatique musicale*).¹⁶⁰ La sua formazione si differenzia da quella dell'assistente puramente tecnico: alle competenze più marcatamente tecnologiche – dallo sviluppo e utilizzo di software alle tecniche di sintesi ed elaborazione del suono – affianca una solida preparazione musicale e musicologica in grado di consentire un dialogo proficuo con i compositori, spesso inesperti sul fronte delle nuove tecnologie. Una professionalità vasta e difficilmente circoscrivibile in un ambito definito che, nel quadro di una spiccata missione pedagogica, spazia dalla composizione all'interpretazione, dalla ricerca applicata alle tecnologie audio alla preservazione delle opere elettroniche e miste.

Nella composizione di un'opera musicale, il RIM riveste allora il fondamentale ruolo di intermediario tra l'idea compositiva e la sua concretizzazione, in un processo che va dalla fase di gestazione a quella di composizione, fino all'esecuzione in concerto.

Laurent Pottier è stato il *réalisateur en informatique musicale* che affiancò Fausto Romitelli nella composizione della parte elettronica di *EnTrance* (1995). Proveniente da studi scientifici, Pottier ha insegnato informatica e computer music all'Università di Montpellier. Nel 1992 è approdato all'IRCAM per seguire il master "Musicologie du XXème siècle", fondato da Hugues Dufourt tre anni prima, e conseguire il titolo di *diplôme d'études approfondies*. Durante il master ha avuto l'occasione di studiare con Jean-Claude Risset e Hugues Dufourt e ha potuto approfondire le tematiche inerenti la *computer music*, in particolare la sintesi del suono con *Csound* e il sistema di aiuto alla composizione *Patchwork*, all'epoca in fase di sviluppo. Inoltre, si è confrontato con la musica di Marco Stroppa e quella di Fausto Romitelli, anche se limitatamente alla parte elettronica di *Natura morta con fiamme*.¹⁶¹

¹⁶⁰ La figura del *réalisateur en informatique musicale* ha assunto via via sempre più rilievo, soprattutto in terra francese. Attualmente è attivo, presso l'Università di Saint-Étienne, un master dedicato: il Master II Pro RIM.

¹⁶¹ La testimonianza di Pottier conferma che il programma *WORKLISP* è stato utilizzato esclusivamente per la generazione della parte elettronica di *Natura morta con fiamme*, e non è stato impiegato in *EnTrance*.

Una volta terminato il master, fino al 1995 ha insegnato al dipartimento di pedagogia dell'IRCAM *Patchwork*, *Csound*, *Max* e altri sistemi per la sintesi e il controllo del suono. In particolare, i compositori a cui Pottier insegnava dimostravano un crescente interesse verso *Patchwork*, flessibile, semplice e intuitivo nella sua interfaccia visuale. In quegli anni ha sviluppato in particolare due librerie per *Patchwork*, *SPdata* e *Csound/Edit-sco*. Poi, nel 1995, si è trasferito nel dipartimento di produzione per lavorare alla composizione di *EnTrance*.

Tra le ragioni che spinsero Pottier a lavorare con Romitelli ci furono sicuramente delle motivazioni legate alle rispettive inclinazioni musicali, così come fattori di ordine tecnologico: l'interesse di Romitelli verso un suono distorto, non convenzionale, proveniente anche da strumenti impiegati nella tradizione rock, incuriosì Pottier. Il suo background musicale era affine: lui stesso, proveniente dall'ambiente della musica *popular*, in passato aveva fatto uso di sintetizzatori, tastiere, così come di strumenti per la manipolazione e la distorsione del suono. Inoltre, li accomunava l'interesse verso la sintesi con *Csound* e il suo controllo parametrico in LISP (e ovviamente *Patchwork*)

Si gettarono così le basi per una collaborazione semestrale, dal luglio 1995 al gennaio del 1996, quando *EnTrance* fu eseguita per la prima volta.

CAPITOLO 4: *ENTRANCE*

Nata nel solco dell'estetica spettrale, *EnTrance* di Fausto Romitelli si colloca nel contesto teorico-tecnologico fin qui delineato. Quando Romitelli si stabilì presso l'IRCAM come *compositeur en recherche*, dal 1993 al 1995, lo sviluppo di *Patchwork* era già ad uno stadio avanzato. Il compositore poté quindi avvalersene per la scrittura dell'opera.

EnTrance è un'opera 'mista' per soprano, ensemble e elettronica.¹⁶² Composta nel corso del 1995, è stata eseguita per la prima volta venerdì 26 gennaio 1996 all'Espace de projection dell'IRCAM dall'Ensemble InterContemporaine e dal soprano Françoise Kubler sotto la direzione Ed Spanjaard.

Alla voce e all'ensemble di legni, ottoni, archi, percussioni e pianoforte, si affiancano dispositivi elettronici che comprendono: un sintetizzatore, un campionatore e una parte elettronica digitale quadrifonica creata in studio da mandare in riproduzione in concerto. La natura eterogenea dell'organico ci induce quindi a parlare di un'organologia allargata.¹⁶³

Nel dettaglio:

- **Legni:** flauto (anche flauto basso), oboe (anche corno inglese), clarinetto in *Sib* (anche clarinetto basso), fagotto;
- **armonica a bocca** suonata da Fg, Cr2, Tbn (modello diatonico da 3,5 cm di lunghezza);
- **ottoni:** due corni, tromba in *Do*, trombone;
- soprano;
- **percussioni intonate:** vibrafono, marimba, glockenspiel, tre gong;

¹⁶² FAUSTO ROMITELLI, *EnTrance*, per soprano, sedici esecutori ed elettronica, Parigi, Ricordi, 1996.

¹⁶³ Come afferma Stiegler, «nous vivons, en ce début de XXIème siècle, un véritable bouillonnement organologique, où instruments acoustiques, technologies analogiques et technologies numériques issues de l'informatique forment un système de plus en plus intégré». BERNARD STIEGLER, "Bouillonnements organologiques et enseignement musical", in «Les dossiers de l'ingénierie éducative», 43, Paris, CNDP, 2003, pp. II.

- **percussioni non intonate:** grancassa, tam tam, bidone metallico, congas di due dimensioni, piatti cinesi di tre dimensioni, piatto splash, piatto ride;
- **tastiere:**
 - pianoforte;
 - sintetizzatore *Yamaha SY99* (anche pianoforte acustico). Per l'*SY99* sono stati progettati due suoni creati ad hoc: a) archi filtrati; b) archi filtrati trasposti di $\frac{1}{4}$ di tono all'acuto;
 - tastiera master keyboard che controlla un campionatore *Digidesign SampleCell*. Il campionatore gestisce cinque diversi suoni: a) suono sintetico di archi filtrati; b) suono di archi filtrati, trasposto di $\frac{1}{4}$ di tono all'acuto; c) chitarra elettrica distorta, nota singola; d) chitarra elettrica distorta, triade maggiore; e) pianoforte;
- **archi:** due violini, viola, violoncello, contrabbasso a 5 corde;
- **parte elettronica su quattro tracce:** i suoni sono notati in partitura, ma la simbologia adottata è approssimativa e puramente funzionale al sincrono tra parte elettronica e ensemble. Sono presenti indicazioni sulle componenti frequenziali dei suoni in forma di notazione tradizionale (approssimazione al semitono), e un'indicazione di massima sul tipo di involuppo del suono, percussivo o meno.

4.1 Lo studio delle fonti

Lo studio di schizzi e abbozzi compositivi costituisce una risorsa preziosa per la ricostruzione della genesi dell'opera. Tuttavia, l'analisi di questi avantesti, seppur essenziale, non è sufficiente quando si ha a che fare con la musica mista, che coinvolge altre forme di scrittura. A partire dagli anni Ottanta compositori e musicisti si sono trovati ad agire in un plesso di apparati elettronici e acustici, frutto di un singolare connubio tra ricerca tecnologica e sperimentazione artistica. Studiare questo repertorio significa indagare i sistemi di sintesi, le tecniche di produzione e riproduzione del suono, i software per la composizione assistita e la nuova organologia. L'utilizzo di una determinata tecnologia porta infatti il compositore ad elaborare una personale forma di scrittura, che l'analista deve indagare: «there exists a specific *écriture* adapted to the electronic production environment, which, while the technology itself is universal and shared by many, is uniquely inscribed in the work».¹⁶⁴

¹⁶⁴ MARC BATTIER, *Electroacoustic music studies and the danger of loss*, in «Organised Sound», v. 9 n. 1, 2004, p. 50.

Nel caso di *EnTrance*, in particolare, le patch informatiche, le librerie e gli stessi software impiegati assumono un ruolo centrale nella comprensione del processo creativo. Sappiamo che già nel 1991, per la composizione di *Natura morta con fiamme*, Romitelli aveva sperimentato la sintesi sonora in *Csound* e il suo controllo in LISP. Sappiamo altresì che *EnTrance* fu composta durante il suo soggiorno all'IRCAM come *compositeur en recherche*, e che tra i suoi interessi le tecniche di composizione spettrale rivestivano un ruolo importante. Inoltre, la testimonianza di Gérard Assayag, direttore dell'*équipe de représentation musicale*, rivela che Romitelli sviluppò una sua personale libreria di funzioni in *Patchwork*.¹⁶⁵ L'indagine deve quindi prendere le mosse da fonti eterogenee, conservate oggi presso l'archivio dell'editore Ricordi, l'ente di produzione (IRCAM), e custodite dai collaboratori di Romitelli, in particolar modo da Laurent Pottier.

4.1.1 Le fonti Ricordi

L'editore è in possesso dei seguenti materiali:

- La partitura depositata dall'autore;
- due CD-ROM che tramandano una versione da concerto postuma, realizzata da Denis Lorrain nel 2005. La versione comprende: a) la parte elettronica incapsulata in un file 'aiff' quadrifonico con una patch in *Max-MSP* per la riproduzione delle singole sezioni; b) i suoni e le istruzioni per il campionatore, convertite da formato *Samplecell* al formato nativo per il campionatore *Akai S5000*; c) la patch per il sintetizzatore SY99, interfacciabile via midi in *Max-MSP*.

4.1.2 Le fonti IRCAM

La ricerca condotta negli archivi della produzione dell'IRCAM ha portato alla luce diverse tipologie di supporti:

¹⁶⁵ Comunicazione privata. Purtroppo non è stato possibile risalire alla libreria originale. Tuttavia, come si vedrà, sono stati individuati documenti che vanno a confermare l'utilizzo di *Patchwork* da parte del compositore.

- Quattro partiture cartacee;
- nove CD-ROM;
- due floppy disk;
- una cassetta S-VHS;
- un disco *SyQuest* EZ 135.

Le fonti cartacee comprendono due partiture complete e due incomplete, che testimoniano le due ultime fasi redazionali: il completamento delle parti di tastiera e la notazione sommaria della parte elettronica.

Il riconoscimento delle altre fonti non è agevole: da un lato perché conservate su supporti ormai difficilmente leggibili a causa dell'obsolescenza degli apparati di riproduzione, dall'altro, nel caso di fonti informatiche, in quanto legate ad un preciso ambiente hardware/software. I documenti, infatti, siano essi sonori o informatici, non esistono in forma di dati direttamente accessibili, come avviene per i documenti testuali, ma sono indissolubilmente legati a un sistema che ne permetta la lettura. Se quindi l'accesso alle partiture è immediato, ciò non si può dire per i restanti materiali. Ammesso che i supporti non si siano deteriorati nel tempo e siano pertanto leggibili, accedere alle informazioni contenute ad esempio in un floppy o, ancor più, in un disco *SyQuest* è tutt'altro che agevole. Anche il supporto a noi più familiare, il CD-ROM, è ormai in fase di abbandono (e con lui, i lettori ottici).

L'allestimento di un sistema tecnologico compatibile con i diversi supporti e formati ha reso possibile l'accesso alle diverse fonti qui descritte. Fa eccezione il disco *SyQuest*, per il quale non è stato individuato un lettore compatibile. Le indicazioni riportate sul disco fanno comunque supporre che possa contenere le patch per il campionatore Akai S5000, ad esso interfacciabile via protocollo SCSI.

La maggior parte dei materiali conservati su CD-ROM tramanda la versione postuma del 2005. Accanto a queste fonti 'recenti', sono state individuate anche fonti risalenti all'epoca della composizione di *EnTrance*:

- La cassetta S-VHS, registrata in formato audio digitale ADAT. Contiene la parte elettronica su quattro tracce e una *click-track* per il direttore;
- due floppy, tra loro copie, che contengono la patch per i suoni del sintetizzatore *Yamaha SY99*, che di fabbrica incorpora un lettore dedicato;
- un CD-ROM in due copie, creato nel febbraio del 1996 e contenente materiale eterogeneo.

Il CD è stato masterizzato con File System Mac OS standard (HFS), con tutta probabilità su workstation *Power Macintosh* e sistema operativo *Apple System 7*. Per la sua lettura è stata allestita una postazione *PowerMac G4* con *Mac OS 9.2* che ha assicurato una piena retro-compatibilità. Il contenuto del documento è prezioso: non solo contiene i file utilizzati per il concerto del 1996, ma tramanda anche una mole di materiali preparatori, in particolare suoni, analisi, patch e codici di programmazione utilizzati per la composizione della parte elettronica. La struttura del disco è la seguente:

- *C O N C E R T *: conserva le fonti per l'esecuzione in concerto, tra cui:
 - la parte elettronica organizzata in sezioni tramite una sessione in *Digital Performer*;
 - i file per il campionatore *Samplecell*, comprensivi di una patch scritta in *Max-MSP* per controllare via master keyboard i diversi suoni richiesti al tastierista;
 - la patch in formato *Galaxy* per il sintetizzatore *SY99*;
- 'Analyses': conserva il risultato di analisi spettrali svolte su suoni concreti quali violoncello, chitarra elettrica e gong;
- 'AudioSVP': contiene suoni generati per cross-synthesis con *Audiosculpt-AudioSPV*;
- 'Csound-1', 'Csound-2' e 'Csound-3': contengono suoni creati con Csound, spesso accompagnati dai file *orchestra* e *score* per la loro generazione;
- 'Lisp': contiene il programma *WORKKLISP* con *lispdef*;
- 'PW': *Patchwork*. Insieme alle patch utilizzate per il controllo della sintesi in *Csound* e *Chant*, nella sottocartella 'Data' sono conservate alcune patch che riportano informazioni preziose sulla struttura armonica delle singole sezioni;
- 'Sons': la cartella contiene una raccolta di campioni di strumenti acustici ed elettrici, in alcuni casi confluiti nella parte elettronica. Troviamo suoni ottenuti con flauto basso, chitarra elettrica basso elettrico distorti, piatti, gong, corno, tromba e trombone. Per quanto concerne la registrazione della chitarra elettrica, qui troviamo i suoni poi affidati al campionatore *Samplecell*.

Per poter accedere alle informazioni contenute nel disco, la workstation è stata configurata con i seguenti software: *Digital Performer 2.7*, *Soft SampleCell Editor*, *Max 3.5.8*, *Galaxy 2.5.4*, *Csound 1.0*, *Patchwork 2.7* con le librerie *Csound/Edit-sco* e *SpData*.¹⁶⁶

¹⁶⁶ Per un approfondimento sui software impiegati si veda *infra*, par. A.2.3.

4.1.3 Le fonti dell'archivio privato di Laurent Pottier

Laurent Pottier preserva una mole considerevole di materiali cartacei:

- Una partitura, che testimonia il penultimo stadio della stesura di *EnTrance*. Tutte le parti strumentali sono complete; le annotazioni sulla parte elettronica sono però assenti;
- schizzi e abbozzi compositivi per mano di Romitelli;
- schizzi e abbozzi per la composizione della parte elettronica per mano di Pottier;
- stampe di screenshot che testimoniano alcune fasi di lavorazione della parte elettronica;
- codice LISP;
- corrispondenza. In particolare, un foglio inviato via fax da Romitelli a Pottier con precise indicazioni sui suoni da generare con Csound.

4.2 Analisi dell'opera

4.2.1 Il motivo ispiratore

L'idea da cui muove il compositore, come suggerisce il titolo, è quella di un rituale finalizzato a indurre uno stato di trance. Il testo di riferimento per la parte di soprano è un mantra di quindici sillabe proveniente dal *Bardo Thodol*, il libro dei morti tibetano: *om a yu vse sa ra ha ra ka ra re sva re hûm p'at*. Nella parte vocale, il compositore mette in campo tecniche di inspirazione/espirazione accompagnate da un movimento rotatorio del capo, quasi ieratico:

La voix fait alterner inspiration, dans un microphone situé à gauche du visage, et expiration, dans un autre situé à droite. Ce mouvement, inspiration-expiration de paroles sacrées monosyllabiques, son accélération et les rotations de la tête dont il s'accompagne, rappellent les rites de certaines musiques traditionnelles qui favorisent l'entrée en transe en s'appuyant aussi sur des facteurs d'ordre physiologique, telle l'hyper ventilation durant l'accélération de la respiration, autrement dit une oxygénation maximale du cerveau.¹⁶⁷

¹⁶⁷ FAUSTO ROMITELLI, programma di sala della prima esecuzione di *EnTrance*, il 26 gennaio 1996. <<http://brahms.ircam.fr/works/work/11518/>>, ultima visita 5 marzo 2017.

Ad enfatizzare questo rituale, due microfoni posti ai lati del volto a distanza ravvicinata amplificano e diffondono la voce in modi differenti: l'espiazione è diffusa frontalmente e viene trattata con un riverbero ordinario; l'inspirazione, al contrario, è veicolata in fondo alla sala e trattata con una riverberazione lunga, innaturale, e con un leggero effetto di eco.¹⁶⁸

La voce costituisce il fulcro e il motore della composizione, in quanto l'evoluzione del ritmo respiratorio scandisce il ritmo armonico della parte strumentale, così come le trasformazioni di densità e volume.

4.2.2 *Articolazione formale*

L'opera, come dichiara Romitelli, è articolata in modo ciclico e prevede la ripetizione di tre stati:

- 1) Respiration lente et régulière, polarisation harmonique, registres instrumentaux gelés, fusion : situation homéostatique, immobile, suspendue. Souffle.
- 2) Respiration accélérée, temps d'inspiration toujours plus bref, crescendo dynamique et agogique, croissance exponentielle de la densité et des volumes, accélération du rythme harmonique et de distorsion, accentuation des transitoires d'attaque au point de produire une perturbation et une distorsion dans la perception des hauteurs.
- 3) Articulation rapide et violente, fin du mouvement inspiration-expiration dans la voix, nouvelle polarisation et régularité extrême du rythme harmonique devenant presque hypnotique, aucune fusion des instruments mais un unique geste furieux.¹⁶⁹

Le fasi del ciclo respiratorio si ripresentano per tre volte all'interno dell'opera, rendendola di fatto tripartita. Una coda chiude il brano (figura 37).

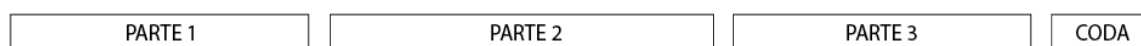


Figura 37 – Macroforma.

¹⁶⁸ L'amplificazione dell'ensemble strumentale è invece di tipo ordinario, per cui i livelli dinamici ed eventuali riverberazioni vanno decisi in base all'acustica della sala, che comunque non deve essere troppo 'secca'. La parte elettronica, registrata su quattro tracce, può essere diffusa secondo diverse configurazioni: a) diffusione stereofonica: tracce 1 e 3 a sinistra, tracce 2 e 4 a destra; b) diffusione quadrifonica: traccia 1, fronte sinistra; traccia 2, fronte destra; traccia 3, retro sinistra; traccia 4, retro destra; c) diffusione esafonica: traccia 1, fronte sinistra e retro sinistra; traccia 2, fronte destra e retro destra; traccia 3, centro sinistra; traccia 4, centro destra. La configurazione a sei altoparlanti fu quella adottata per la prima esecuzione dell'opera.

¹⁶⁹ FAUSTO ROMITELLI, programma di sala della prima esecuzione di *EnTrance*.

I tre stati risultano chiaramente riconoscibili nelle prime due parti dell'opera, in cui la linearità del processo non lascia margini di dubbio. La terza parte, al contrario, risulta di più articolata e di difficile comprensione. Verrà analizzata in seguito.

Seguendo la categorizzazione proposta da Laurent Pottier,¹⁷⁰ assegniamo ai tre stati descritti da Romitelli le lettere **A**, **B** e **C**, per indicare rispettivamente staticità, accelerazione e articolazione rapida e violenta.

Le sezioni di tipo **A** sono a loro volta divisibili in due sotto sezioni. La prima, **A'**, ha funzione di preludio: il materiale armonico si esaurisce in poche note, i registri strumentali sono fissi, la stasi è totale. Gli archi contribuiscono con suoni ottenuti tramite la tecnica degli armonici, le tastiere creano un tappeto con un suono di archi filtrati;¹⁷¹ i legni, senza imboccatura, filtrano il respiro degli strumentisti. L'intervento della parte elettronica scandisce l'inizio della sottosezione successiva, **A''** (battuta 8 in figura 39).

Si avvia quindi un processo di trasformazione timbrica costante e graduale, in cui i diversi aggregati accordali sono percettivamente molto simili gli uni agli altri. Nonostante la densità sonora aumenti leggermente (figura 38), intervengano gli ottoni e i legni e gli archi ritornino gradualmente a tecniche esecutive ordinarie, la sensazione di immobilità permane.



Figura 38 – Graduale aumento della densità armonica nella sezione **A''** della prima parte dell'opera.

Le sezioni di tipo **B** rappresentano l'accelerazione del ritmo respiratorio che accompagna l'accelerazione del ritmo armonico. Le fasi di inspirazione, brevissime, sono seguite da un'articolazione vocale forte, decisa, che innesca rapidi movimenti nell'ensemble strumentale, quali arpeggi e glissandi. A queste fasi, brevi e con intensità elevata, si alternano altre fasi in cui i singoli suoni, privati del loro transitorio d'attacco, tendono alla fusione timbrica su dinamiche più contenute – emblematica in tal senso l'indicazione per il soprano 'fondendo con l'ensemble'. La dinamica oscilla tra lo sforzato e il pianissimo, la densità sonora è elevata. Gli aggregati armonici sono nettamente differenziati e inseriti in un processo di compressione temporale. La distorsione, intesa sia armonicamente, sia come sinonimo di saturazione – nella parte elettronica troviamo infatti suoni di chitarra elettrica e basso elettrico distorti – è impiegata in modo sistematico.

¹⁷⁰ Si veda LAURENT POTTIER, *Le contrôle de la synthèse sonore informatique*.

¹⁷¹ Per il dettaglio del campionatore e del sintetizzatore si veda *infra*, par. A.3.

L'articolazione 'rapida e violenta' di cui parla Romitelli è riconoscibile nelle sezioni di tipo C. La componente ritmica viene esaltata a scapito della densità sonora, che subisce una rarefazione. Il ritmo armonico è nuovamente annullato, i timbri dei singoli strumenti emergono distintamente mentre nella parte vocale, ove presente, singole cellule melodiche vengono reiterate in modo quasi 'compulsivo'. A contrastare la rapida articolazione dell'ensemble, contribuiscono suoni di sintesi registrati nella parte elettronica e il suono di chitarra elettrica distorta affidato al campionatore. Quest'ultimo in particolare è dirompente nell'equilibrio generale della sezione: le singole note sono da riprodursi ad elevata intensità e hanno una durata di diversi secondi (figura 41). Il carattere violento delle sezioni C è marcato da indicazioni agogiche quali 'furioso', 'con estrema violenza', 'molto ritmato', 'martellato', 'sempre *f*', 'brutale'.

Come accennato, le prime due parti presentano in successione le sezioni A (A' + A"), B e C e sono organizzate come da figura 40:

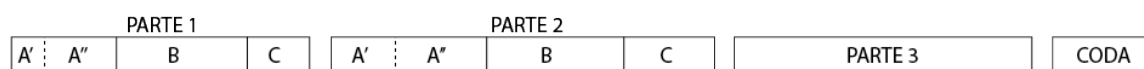


Figura 40 – Dettaglio dell'articolazione interna alle prime due parti.

La terza parte merita invece un approfondimento. Qui Romitelli trasfigura la scansione lineare delle fasi descritte in precedenza.

La parte si apre a b. 335, come ci si aspetterebbe, con una sezione di 11 battute (38 secondi) che è possibile ricondurre sicuramente ad A'. Fanno seguito 6 battute (circa 21 secondi) in cui, contro ogni previsione, non compare A". Al posto della trasformazione timbrica appare un momento di stasi: soprano, fiati ed elettronica tacciono – l'unico strumento aerofono impiegato è l'armonica a bocca – e gli archi eseguono suoni armonici sul *ppp*, così come gli 'archi filtrati' della tastiera che suona in un registro sovracuto. Si tratta a tutti gli effetti di una pausa all'interno processo, che chiameremo R.

131

fl *legato*
sempre f

ob *legato*
f *(f)*

cl *legato*
sempre f

fg *legato*
sempre f

cr I *p* *(pp)* *p* *p*

cr II *(pp)* *p* *p*

tr *(pp)* *(pp)*

tne *(pp)* *p*

sopr

perc *to* *(f)*
molto pp *molto sfz*

tast I

tast II *Metal tenuto sino alla fine del Suono*
(effetto all'indietro successivo)

vn I *sempre f* *(f)*

vn II *sempre f*

vla *sempre f*

vc *sempre f*

cb

el

Figura 41. Fausto Romitelli, EnTrance, batt. 131-140.

A b. 352 interviene l'elettronica e la densità strumentale aumenta leggermente, facendo presupporre l'inizio della parte **B** per la durata di 30 battute (56 secondi). Come nelle altre sezioni **B**, essa è caratterizzata da tempi di inspirazione molto brevi per il soprano, da un'accelerazione progressiva del ritmo armonico e da elementi di distorsione spettrale che richiamano quanto si può riscontrare ad esempio in **1B**. Tuttavia, il movimento melodico della voce è pressoché nullo, la densità sonora è contenuta, i legni rimangono senza imboccatura e la dinamica oscilla intorno al *pp*, caratteristiche tipiche delle sezioni di tipo **A**. È possibile quindi considerare questa sezione come un'ibridazione tra le due tipologie descritte, pertanto si indicherà con **AB** (figura 42).

Un repentino crescendo a b. 381 innesca quella che è a tutti gli effetti la sezione **C**. Internamente, possiamo dividerla in **C'** e **C''**. La separazione è marcata a b. 412 quando il soprano, dopo aver ossessivamente insistito sulle corde di Sol \flat e La in **C'**, tace lasciando all'ensemble il compito di condurre la seconda parte della sezione, **C''**. A turbare l'unità di **C''**, Romitelli inserisce inoltre due battute (7 secondi) in cui lo smorzamento globale dà l'illusione di fine sezione. Ma immediatamente un altro crescendo riporta in auge tutta la potenza di **C''**, che perdura per altre 10 battute (10 secondi). Le due battute descritte possono essere considerate alla stregua di un respiro interno alla sezione, indicato quindi con il simbolo '.

Il ciclo respiratorio potrebbe quindi considerarsi concluso, ma a b. 441 compare una nuova sezione. L'assenza di tecniche di respirazione, per la voce (che tace da qui fino alla fine dell'opera) così come per i legni, fuga il dubbio che si possa trattare dell'inizio di un nuovo ciclo. Partendo da un pianissimo, in una situazione iniziale di stasi e di estrema rarefazione, una sequenza di aggregati accordali tra loro molto simili si trasforma gradualmente in un crescendo dinamico e in un'accelerazione costante, resa ancor più evidente dall'impiego di scale ascendenti affidate al corno. L'accelerazione del ritmo armonico può suggerire un'affinità a quanto visto in **AB**, ma l'assenza del respiro e il costante incremento dinamico rendono di fatto la sezione un caso a sé: sarà pertanto chiamata **D** (figura 43).

The image displays a page of a musical score for Fausto Romitelli's *EnTrance*, measures 361-370. The score is arranged in a standard orchestral format with multiple staves. The instruments listed on the left are: fl (flute), ob (oboe), cl (clarinet), ARM fg (bassoon), cr I (cor I), cr II (cor II), tr (trumpet), tne (trombone), sopr (soprano), perc (percussion), tast I (piano I), tast II (piano II), vn I (violin I), vn II (violin II), vla (viola), vc (violoncello), cb (contrabbasso), and el (electric bass). The score includes various musical notations such as notes, rests, dynamics (e.g., *pp*, *ppp*), articulation marks, and performance instructions like "altari sempre ppp" and "dist. 2.96". The vocal line (sopr) includes lyrics in Italian: "ju se so ho", "ho se hum po-)", and "ju". The percussion part includes markings for "VIB", "MARE", "Gong", and "MAP". The electric bass part includes markings for "dist. 2.96" and "dist. 2.1".

Figura 42. Fausto Romitelli, *EnTrance*, batt. 361-370.

La sezione trova la sua giustificazione dopo 21 battute (38 secondi): il crescendo fino al *ff* porta, a b. 462, ad un'ultima sezione di tipo C (25 secondi). L'articolazione è qui esasperata, ostinati ritmici sempre più insistenti non lasciano spazio ad alcun movimento armonico. In chiusura di sezione, in corrispondenza di una corona, la parte elettronica introduce un effetto di 'vento' – rumore filtrato – e un ostinato in forma di rapida concatenazione di arpeggi in dissolvenza.¹⁷²

La terza parte è dunque unica nel suo genere, sia per l'introduzione di nuove sezioni sia per la frammentarietà di C. È in questo contesto che le parole di Romitelli assumono un significato pregnante:

On cherchera en vain dans cette pièce élégance et harmonie de proportions, équilibre formel et transformations graduelles et linéaires. En revanche, j'ai exhibé l'aspect obsessionnel et violent, répétitif et visionnaire, oscillant entre une extrême densité et une extrême raréfaction.¹⁷³

Chiude l'opera una coda dove la dinamica oscilla tra il *p* e il *ppppp* e la fusione tra l'ensemble e l'elettronica è totale (bb. 483–520 / 84 secondi). Qui una sequenza di accordi viene condotta progressivamente verso il grave fino ad arrivare alla scordatura delle ultime due corde del contrabbasso, in un processo che porta poco a poco verso il silenzio coronato nell'ultima battuta (figura 44).

The image shows a musical score for the finale of EnTrance. It consists of four staves: viola (via), violin (vc), double bass (cb), and electronics (el). The score is heavily annotated with handwritten notes and symbols. Key annotations include:

- 'pendenzia al nulla' above the violin staff.
- 'poco... poco... verso il silenzio' with arrows indicating a gradual decrease in volume across the double bass staff.
- 'scordare la IV' and 'V. scordare la V.' indicating string detuning in the double bass.
- '* impercettibile il cambio di arco' in the double bass staff.
- 'silenzio' at the end of the piece.
- Various dynamic markings like *ppppp* and *p* are present.

Figura 44. Fausto Romitelli, EnTrance. Finale (estratto).

¹⁷² L'arpeggio finale è ottenuto con tutta probabilità tramite un *sequencer*, in un procedimento analogo a quanto già sperimentato nel 1973 dai Pink Floyd per il brano *On the Run*.

¹⁷³ FAUSTO ROMITELLI, programma di sala per *EnTrance*.

Schematizzando, la struttura dell'intera opera può esser rappresentata come in figura 45.

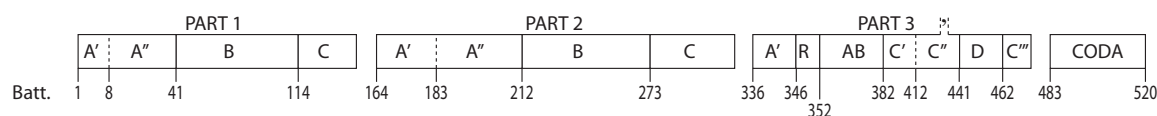


Figura 45 – Articolazione interna alle sezioni in EnTrance.

4.2.3 Analisi della sezione 1B

Le fonti informatiche, in particolare le patch di *Patchwork*, offrono all'analista un'opportunità eccezionale: ripercorrere il processo creativo e indagare così nel dettaglio la struttura armonica dell'opera.

Si prenda ad esempio la sezione 1B. L'analisi dei file conservati nella cartella 'PW/divers PW/Data' del CD IRCAM mostra come la componente armonica sia calcolata a partire da una linea melodica precisa (figura 46):



Figura 46 – Melodia di base per la sezione 1B con le rispettive frequenze.

Secondo un procedimento prettamente spettrale, ciascuna nota assume il ruolo di fondamentale di uno spettro costituito da 30 parziali che chiameremo S. Gli spettri non sono però armonici, ma vengono distorti con coefficienti diversi (figura 47), in perfetta analogia con i procedimenti descritti da Murail per illustrare la sua idea di armonia frequenziale.¹⁷⁴

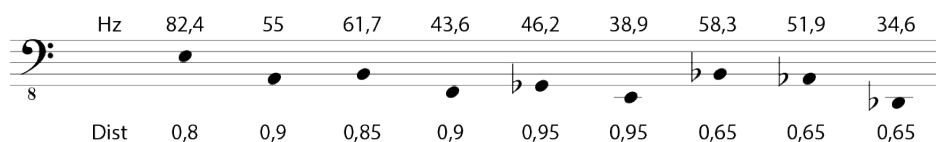


Figura 47 – La stessa melodia con il dettaglio dei coefficienti di distorsione.

¹⁷⁴ Si veda *infra*, par. 1.2.4.

Sul piano formale l'articolazione interna alla sezione consiste di nove sotto-sezioni, pari al numero degli spettri distorti.

Per il calcolo delle frequenze che costituiscono gli spettri, Romitelli ha probabilmente utilizzato la seguente funzione, scritta in linguaggio LISP:

```
;;; Definizione della funzione. part è il numero di
;;; parziali, freq la frequenza fondamentale,
;;; dist il coefficiente di distorsione:
```

```
(defun dist(part freq dist)
  (let ((res))
    (dotimes (n part res)
      (setf res (cons (* freq (expt(+ n 1) dist))res)))
    (reverse res)))
```

Una volta valutata la funzione per ogni nota con i diversi parametri, i dati sono stati raccolti in un unico file di testo ('datachdsharmo dists'), dal quale si riportano i primi due spettri con le rispettive 30 parziali:

```
(defvar *distchds* ())      ;;; inizializzazione della variabile
(setf *distchds*' (        ;;; assegnazione della variabile
  ("Mil-0.8")
  (82.4068892282175      143.47872767421208      198.45430487658496
  249.81097439559304    298.63418729776294      345.52901379770356
  390.87865236297296    434.94616895527435      477.92255590401237
  519.9523199430956     561.1483217911464      601.6009551934901
  641.3841186639341     680.559261990032      719.1782217192105
  757.2852647750104     794.9185970819026     832.1115005083225
  868.8932052694906     905.2895700271768     941.3236195697617
  977.0159752559147     1012.3852035090151    1047.4481008463588
  1082.2199291616728    1116.7146115841522    1150.944896780484
  1184.9224977636338    1218.658209932158     1252.1620120559269)
)

(("1a0-0.9")
(55.000000000000014    102.63362906904884    147.8331458737258
191.52112392514735    234.11847869321585    275.86640468571926
316.92093278732216    357.39105439674364    397.3570730713144
436.88052909835505    476.0100325290473     514.7849136568403
553.2376228386881     591.395371998381      629.281294770708
666.9152892645753     704.3146447549119     741.4945171920764
778.4682963951607     815.2478940176426     851.8439724942805
888.2661293042202     924.5230469088051     960.6226159745179
996.5720375568317     1032.3779085357864    1068.0462935874239
1103.5827862333454    1138.9925609568722    1174.2804179561383)
)))
```

In figura 48, si riportano i primi due spettri trascritti in notazione musicale.

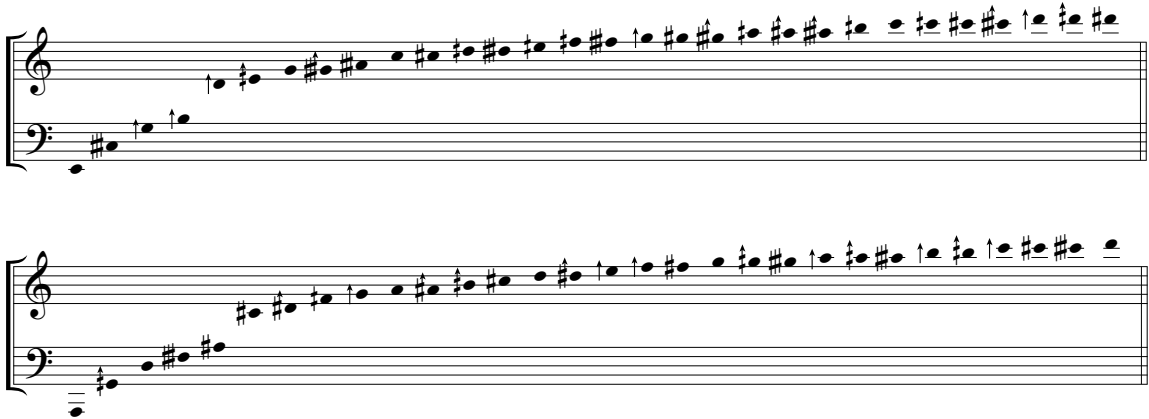


Figura 48 – Spettri distorti S^1 a partire dal Mi_1 (S^1 : distorsione 0.8) e dal La_0 (S^2 : distorsione 0.9). Le frequenze sono state approssimate in notazione tradizionale con una risoluzione di un ottavo di tono.

Gli spettri ottenuti sono quindi orchestrati in partitura, in un'operazione che produce un effetto di fusione timbrica e che richiama la tecnica spettrale della sintesi strumentale. Nel caso dello spettro S^1 , sul Mi_1 , le parziali sono così assegnate. Legni: parziali 4, 9, 15, 19; ottoni: parziali 2, 3, 4, 5, 7, 8, 11, 17; voce: parziale 7; percussioni: parziali 10, 18; tastiera (suono di archi filtrati): parziali 13, 14, 20, 21, 23; archi: parziali 1, 2, 3, 6, 8, 12, 16, 25. In figura 57 a pagina III sono riportate le prime 10 battute della sezione 1B (bb. 41-50). L'orchestrazione sopra descritta è riscontrabile nelle battute 46-50, ed è indicata con il colore blu.

Per comprendere il modo in cui Romitelli ha invece operato nelle prime battute dell'esempio, viene in aiuto una patch di *Patchwork*: 'chds-secT' (figura 49).

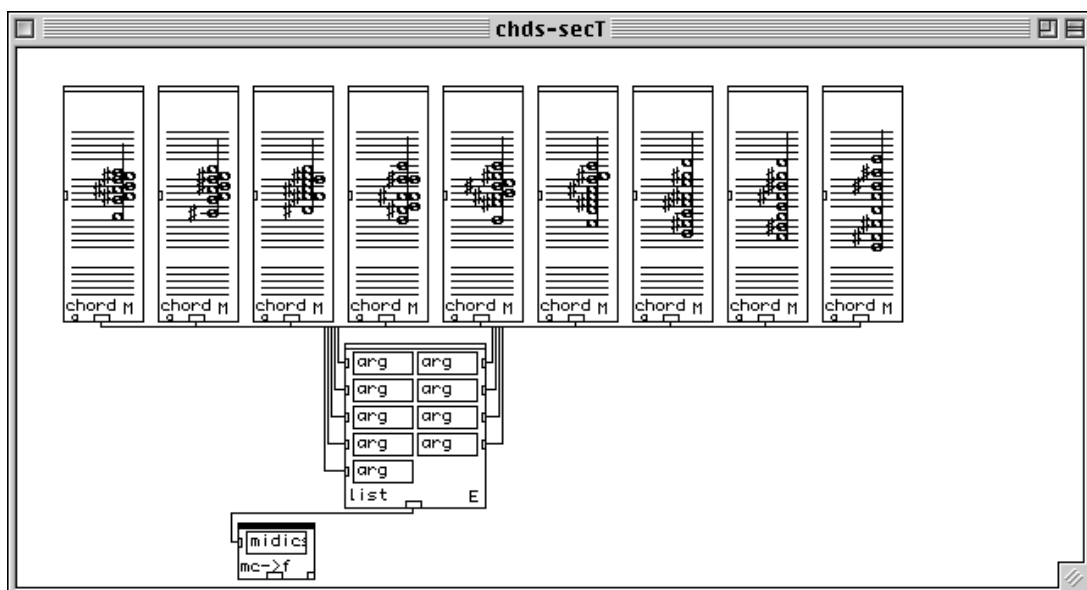


Figura 49 – Patch con gli aggregati T.

Questa patch, nella sua semplicità – si tratta infatti di una semplice concatenazione di accordi – fornisce il materiale armonico di base per le battute 41-45.

A questo punto dell'analisi, il legame tra gli aggregati definiti dalla patch 'chds-secT' e gli spettri non è chiaro. Tuttalpiù, è possibile constatare l'esistenza di alcune note comuni. Per comprendere le tecniche utilizzate da Romitelli per derivare gli aggregati dagli spettri di base è necessario ricorrere agli schizzi compositivi cartacei, conservati presso l'archivio privato di Laurent Pottier.

Prendiamo ad esempio la seconda sotto-sezione, quella definita dalla distorsione dello spettro sul La_0 (S^2). Il compositore sceglie alcune parziali tratte dallo spettro di riferimento: parziali 6, 9, 10, 12, 15, approssimate al semitono (figura 50, passo 1). Tra le prime due note, il $Do\#$ e il Sol , pone un asse di simmetria che divide a metà il tritono; l'asse è dunque sul Mi . A questo punto, opera una riflessione degli intervalli (passo 2). Infine, traspone le nuove note di due ottave all'acuto (passo 3).

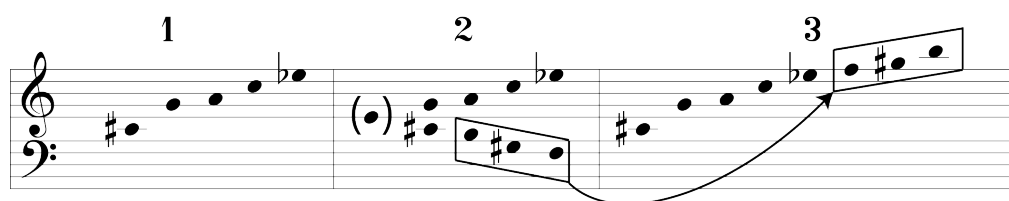


Figura 50 – Procedimento di creazione dell'aggregato a partire da S^2 . Passo 1: scelta delle parziali; passo 2: riflessione intervallare sul Mi_4 ; passo 3: trasposizione all'acuto delle note ottenute.

Questo sistema non è applicato rigidamente per la costruzione di tutti i nove aggregati che compaiono nella sezione **1B**. Si notano infatti una certa libertà nel posizionamento degli assi di simmetria e interventi di 'selezione', 'scarto' e 'aggiunta' di note. Un esempio illuminante è fornito dall'aggregato nella prima sottosezione (S^1 , spettro sul Mi_1): qui, una volta scelte le parziali (4, 7, 9, 11 e 12) viene adottato un procedimento analogo a quanto visto in precedenza fino all'ultimo passo, dove Romitelli decide di rimuovere il Mi_5 per aggiungere un La_5 (figura 51). L'aggregato, così rivisto, è orchestrato nelle parti in rosso di figura 57.

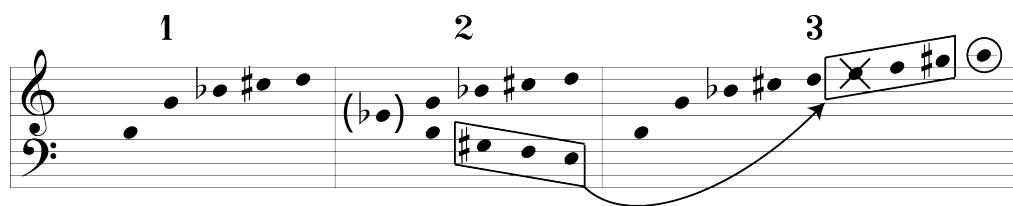


Figura 51 – Creazione dell'aggregato a partire dallo *S*. Si noti nel passo 3 la scelta arbitraria di sostituire il *Mi*₅ con il *La*₅.

I nove aggregati, a cui è stata assegnata la lettera **T**,¹⁷⁵ sono stati ottenuti con tecniche combinatorie¹⁷⁶ che sembrano rispondere alla necessità sentita da Boulez di strutturare il timbro all'interno del discorso musicale.¹⁷⁷ Sono direttamente collegati ai rispettivi spettri distorti *S*, li anticipano, ma l'idea compositiva che li genera è ben lontana dall'estetica spettrale.

Possiamo quindi considerare le nove sotto-sezioni di **iB** come a loro volta bipartite: l'aggregato **T** ottenuto attraverso procedimenti intervallari e lo spettro *S*. La transizione tra le due situazioni armoniche è ammorbidita dalle note in comune, affidate principalmente agli ottoni, come si può notare nell'esempio riportato in figura 57.

Rimangono da chiarire altri due aspetti. L'articolazione nelle battute 44-47 di marimba, tastiera II e violini, evidenziati in verde in figura 57, ripropone l'aggregato **T**. I singoli suoni sono però esposti sia all'altezza definita dall'aggregato **T** sia trasposti all'acuto fino anche a tre ottave di distanza. I rapporti intervallari tra le note non vengono perciò mantenuti.

Un discorso a sé va fatto invece per gli arpeggi e i glissandi presentati alle battute 41 e 43 ed evidenziati in giallo nell'esempio in figura 57: qui il materiale base, nuovamente tratto dall'aggregato **T**, viene sottoposto ad un ulteriore procedimento combinatorio. Ponendo un asse di simmetria sul primo intervallo dell'aggregato, Romitelli ottiene per inversione una nuova riserva di suoni (figura 52).

¹⁷⁵ Va precisato che la lettera **T** viene inoltre utilizzata da Romitelli per indicare l'intera sezione della parte elettronica corrispondente a **iB**.

¹⁷⁶ L'impiego di tecniche combinatorie non stupisce se si considera la formazione di Romitelli sotto la guida di Franco Donatoni.

¹⁷⁷ Si veda *infra*, par. 3.3.I.

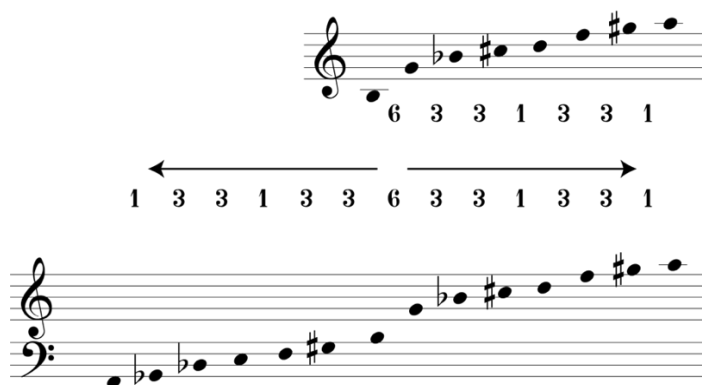


Figura 52 – Inversione intervallare dell'aggregato T.

A battuta 43, gli arpeggi affidati ai legni attingono direttamente da questo materiale, come si può notare in figura 53; gli ottoni e gli archi, invece, producono glissandi che trovano origine e destinazione nelle note appena individuate.

Figura 53 – Dettaglio della sorgente degli arpeggi dei legni in battuta 43: flauto basso (rosso); oboe (blu); clarinetto basso (giallo); fagotto (verde).

Il procedimento adottato per la composizione degli arpeggi in battuta 41 è simile a quanto appena visto, fatta salva una trasposizione all'acuto. La trasposizione è operata in modo da raggiungere, sull'ultima nota dell'arpeggio, il Fa. In figura 54, un'immagine esplicativa che mette in relazione la sezione dei legni e degli archi con la riserva di suoni individuata per inversione intervallare.

Figure 54 shows a musical score for woodwinds and strings. The woodwinds include Flute (fl), Oboe (ob), Clarinet Bass (cl), and Bassoon (fg). The strings include Violin I (vn I), Violin II (vn II), Viola (vla), and Violoncello (vc). The score is in 2/4 time with a tempo of quarter note = 60. The woodwinds and strings are playing arpeggiated figures. Below the score, a diagram shows a scale with three overlapping colored regions: a green region for Transposition: -2, a red region for Transposition: 0, and a yellow region for Transposition: +8.

Figura 54 – Dettaglio degli arpeggi in battuta 41: oboe e violino I (giallo) sono trasposti all'acuto di un intervallo di sesta minore; clarinetto basso e viola (rosso) non subiscono trasposizione; fagotto e violoncello (verde) sono trasposti al grave di un intervallo di seconda maggiore.

Anche la parte di soprano che innesca l'intera sezione 1B è riconducibile a una trasformazione dell'aggregato T. Se infatti eliminiamo dall'aggregato T la nota più grave, il Si₃, e trasponiamo all'ottava inferiore le due note più acute, Sol#₅ e La₅, compattando quindi il materiale melodico all'interno dell'ottava, otteniamo esattamente la parte del soprano (figure 55 e 56).

Figure 55 shows a combinatorial procedure for the soprano scale. It consists of two staves of musical notation. The top staff shows a sequence of notes: C#4, D#4, E4, F#4, G4, A4, B4. The bottom staff shows a sequence of notes: C#3, D#3, E3, F#3, G3, A3, B3. An arrow points from the bottom staff to the top staff, indicating the transformation process.

Figura 55 – Procedimento combinatorio per la scala affidata al soprano.

Figure 56 shows the soprano part for measures 40-41. The notation is in 2/4 time. Measure 40 starts with a quarter rest, followed by a quarter note G#4. Measure 41 starts with a quarter note A4, followed by a quarter note B4, a quarter note C#5, and a quarter note D5. A triplet bracket is over the last three notes. The lyrics 'p'a' and 't a' are written below the notes.

Figura 56 – Soprano. Battute 40-41.

The image shows a page of a musical score for Fausto Romitelli's *EnTrance*, measures 41-50. The score is heavily annotated with color-coded regions and markings. The instruments listed on the left are: fl (flute), ob (oboe), cl (clarinet), fg (bassoon), cr I (coro I), cr II (coro II), tr (trumpet), tne (trombone), sopr (soprano), perc (percussion), tast I (piano I), tast II (piano II), vn I (violin I), vn II (violin II), via (viola), vc (cello), cb (contrabbasso), and el (elettrodomestico). The tempo is marked as $\text{♩} = 60$. The score is annotated with color-coded regions: blue for the S spectrum, red for aggregate T, green for octave transpositions of T, and yellow for arpeggiated and glissando passages derived from the inverted T aggregate. The score includes various musical notations such as dynamics (pp, p, f, sf), articulation (acc, sfz), and performance instructions like 'Pianoforte' and 'fondendosi con l'ensemble'. There are also handwritten notes and markings like 'MAR', 'VIB', 'arco', 'pizz.', and 'dist. R.B.'.

Figura 57 – Fausto Romitelli, *EnTrance*, bb. 41-50. In blu, l'orchestrazione dello spettro S; in rosso, l'aggregato T; in verde, trasposizioni di ottava delle note dell'aggregato T; in giallo, arpeggi e glissandi derivati dall'inversione dell'aggregato T.

Riassumendo, possiamo affermare che l'intera sezione **1B** è composta di nove sotto-sezioni, una per ciascuna nota della melodia germinale. Ciascuna sottosezione è suddivisa in due: l'aggregato **T** con le sue trasformazioni seguito dallo spettro distorto (**S**) da cui è stato generato. Alle nove sottosezioni si aggiunge, in chiusura, una codetta. La componente armonica si inserisce in un processo di compressione temporale, quindi di accelerazione, come si può notare in figura 58.

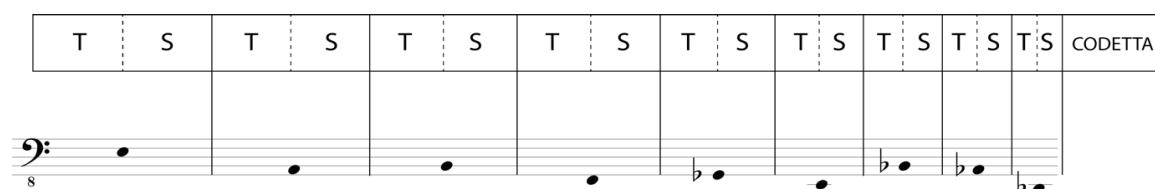


Figura 58 – Articolazione formale della sezione **1B**. Si noti l'alternanza di aggregati (**T**) e spettri distorti (**S**) a partire dalla melodia riportata in basso.

4.2.4 La parte elettronica della sezione **1B**

Vanno a comporre la parte elettronica della sezione **1B** suoni di tipo percussivo (pizzicato) e suoni ad inviluppo a doppia forcilla (crescendo-diminuendo), sia di sintesi, sia concreti (nello specifico, basso elettrico distorto). Un importante contributo alla comprensione della natura di questi suoni viene dalla tesi dottorale di Laurent Pottier in cui si dedica ampio spazio alla trattazione delle varie tecniche di sintesi da lui adottate assieme a Romitelli.¹⁷⁸ Senza tale studio la ricostruzione del processo creativo e l'analisi dei rapporti tra elettronica ed ensemble strumentale risulterebbe tutt'altro che agevole. Nella composizione della parte elettronica sono state impiegate le seguenti tipologie di suono:

- Campionamento;¹⁷⁹
- cross synthesis (*Audiosculpt-SVP*);
- filtraggio (*Csound*);
- sintesi per forma d'onda variabile (*Csound*);
- sintesi additiva (*Csound*);
- sintesi FOF (*Chant + Csound*).

¹⁷⁸ Si veda LAURENT POTTIER, *Le contrôle de la synthèse sonore informatique*.

¹⁷⁹ I suoni campionati provengono da strumenti acustici (flauto contrabbasso), da strumenti elettrificati (chitarra elettrica, basso elettrico), così come da sintetizzatori. In questo caso, sono stati prelevati suoni sintetici di campana da *Yamaha SY77* e da *Yamaha DX7*.

Nella sezione **1B** sono state utilizzate tre tecniche: campionamento, sintesi FOF (funzione d'onda formantica) e sintesi per forma d'onda variabile. I parametri della sintesi sonora sono stati elaborati in ambiente *Patchwork*.

4.2.4.1 Campionamento

I campioni utilizzati nella parte in analisi derivano dalla registrazione di un basso elettrico suonato da Kasper T. Toeplitz. Lo strumento, trattato con distorsori, è suonato tradizionalmente con le dita oppure con plettri e archetti. Le frequenze fondamentali dei suoni poi montati nella parte elettronica della sezione **1B** corrispondono alle frequenze fondamentali degli spettri **S**.

4.2.4.2 Sintesi FOF

Questo tipo di sintesi ha generato i suoni di tipo percussivo chiamati 'datachantfof-cb'. Secondo Pottier, la sintesi FOF impiegata in *EnTrance* si fonda su di un modello di risonanza di un contrabbasso pizzicato. Il modello, sviluppato originariamente in *Chant* da Jean-Baptiste Barrière – da quest'ultimo utilizzato in *Hybris*, una composizione del 1988 – è stato però alterato:

Seuls les partiels d'une largeur de bande inférieure à 1,5 Hz – c'est-à-dire ceux qui ont les durées de résonances les plus longues – ont été conservés et leurs largeurs de bande ont été multipliées par trois pour augmenter les durées de résonances. Chaque résonance a été dupliquée et transposée au quart de ton supérieur pour enrichir le modèle.¹⁸⁰

Il modello di risonanza è quindi stato modificato in *Patchwork* sfruttando le possibilità offerte dalla libreria *PW-chant*. Purtroppo però in questo caso non è stato possibile accedere alle patch.¹⁸¹ Di conseguenza la comprensione del processo di trasformazione è stata più ardua.

Per analizzare i suoni basati su sintesi FOF e comprenderne il legame con il modello armonico, ci si può avvalere di due strumenti: le annotazioni in partitura e l'analisi dei singoli suoni. In partitura, Romitelli indica i suoni percussivi evidenziandone la frequenza

¹⁸⁰ LAURENT POTTIER, *Le contrôle de la synthèse sonore informatique*, p. 188.

¹⁸¹ Si veda *infra*, par. A.2.3.

fondamentale – in notazione tradizionale – e il coefficiente di distorsione spettrale. È il caso ad esempio riportato in figura 59.

Figura 59 – FAUSTO ROMITELLI, EnTrance, p. 5. Prima sottosezione. In rosso si evidenzia l'indicazione del suono percussivo basato sul modello di risonanza di un contrabbasso pizzicato (cb). È inoltre indicato il coefficiente di distorsione (0,8).

L'analisi FFT dei suoni di sintesi evidenzia una buona approssimazione delle frequenze calcolate per gli spettri distorti S. Rimanendo all'interno della prima sottosezione, in figura 60 si riporta l'analisi del suono evidenziato in figura 59:

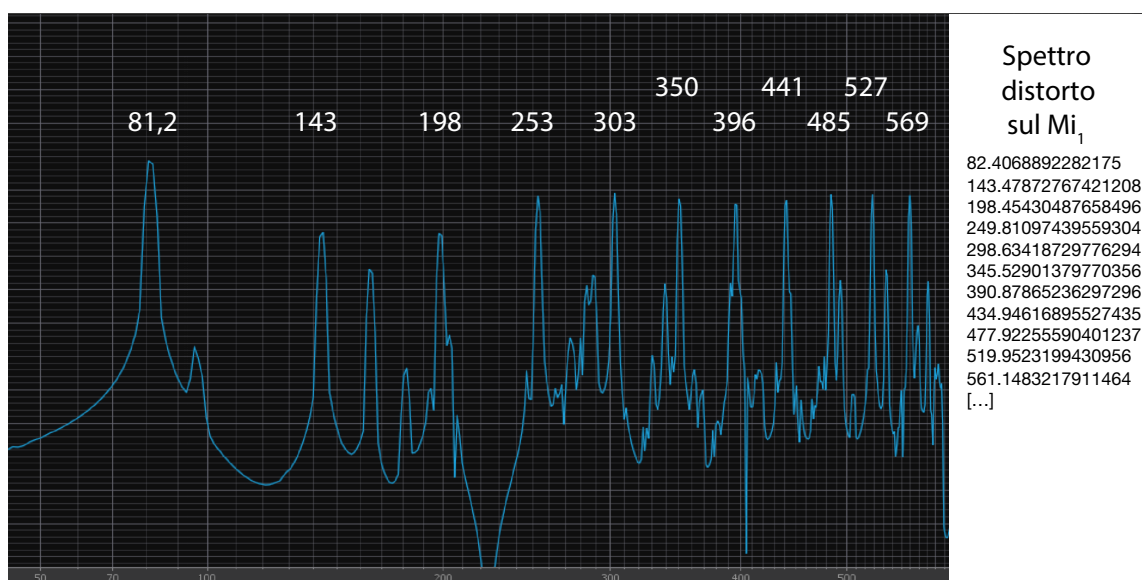


Figura 60 – Analisi FFT con il dettaglio in Hz delle frequenze principali, comparata con le frequenze ottenute dalla distorsione dello spettro S'. Analisi effettuata con iZotope RX 4. Finestra di tipo "cos3" di 32768 punti.

La FFT ci mostra come le frequenze del suono analizzato appartengano allo spettro sul Mi_1 , distorto secondo il coefficiente indicato da Romitelli in partitura (0,8). Si tratta di un'ulteriore conferma che anche i suoni basati su sintesi FOF sono ancorati ai diversi spettri distorti su cui si basa la sezione **IB**.

4.2.4.3 Sintesi per forma d'onda variabile

Questa tipologia di sintesi sfrutta le possibilità offerte dagli oscillatori digitali di generare un suono complesso a partire da un unico oscillatore. L'oscillatore digitale, infatti, legge ripetutamente una tabella di valori – più o meno semplice, ma discreta – che va a descrivere il periodo. Nella sintesi per forma d'onda variabile la tabella dell'oscillatore muta nel tempo, in un processo di trasformazione graduale da una forma iniziale a una finale. La transizione tra le due tabelle avviene in modo graduale, interpolando i valori numerici dei singoli punti. In *EnTrance* la sintesi è affidata a *Csound*, ma per agevolare la manipolazione dei diversi parametri Laurent Pottier si è avvalso dell'ambiente *Patchwork* e delle librerie *SpData* e *Csound/Edit-sco*.

Nella sezione **1B**, la sintesi per forma d'onda variabile è stata impiegata per generare tre diverse tipologie di suoni, chiamati rispettivamente 'celldist', 'V91' e 'V04'.

4.2.4.3.1 Suoni 'celldist'

La sintesi dei suoni 'celldist' prende a modello il suono reale di un violoncello. La forma d'onda proviene infatti dall'analisi delle sue prime dieci parziali. L'obiettivo, come ricorda Pottier, non risiedeva nella semplice risintesi di un suono d'archi, bensì nella generazione di un suono complesso e allo stesso tempo distante dagli indici percettivi di riferimento dello strumento acustico: «chaque son synthétisé était destiné à être employé comme un son partiel faisant partie d'un son plus complexe».¹⁸² Il suono finale è ottenuto prendendo ognuna delle 30 parziali dello spettro distorto **S** come fondamentale per il controllo della lettura della tabella di sintesi. A differenza di una semplice sintesi additiva ottenuta per sovrapposizione di suoni semplici sinusoidali, la ricchezza di armoniche offerta dall'analisi del violoncello crea uno spettro complesso, in un procedimento che richiama molto da vicino la tecnica spettrale della sintesi strumentale.

Per generare i parametri della sintesi, Pottier ha creato un'apposita patch in *Patchwork* (figura 61):

¹⁸² LAURENT POTTIER, *Le contrôle de la synthèse sonore informatique*, p. 186.

software *Diphone*.¹⁸³ Nel terzo riquadro è possibile notare il modulo `edit-sco-obj`, che ha il compito di aggregare tutti i parametri calcolati e di scrivere il file `score` per *Csound*.

Pottier ha inoltre conservato i singoli file score. La loro analisi comprova che anche i suoni di tipo 'celldist', come già quelli generati dalla FOF, sono costruiti a partire dagli spettri distorti *S*. Nell'esempio in figura 62 sono state evidenziate in blu le frequenze di ciascuna parziale in *Csound*, affiancate alle frequenze calcolate dalla funzione di distorsione spettrale. La corrispondenza è precisa.

	<i>f</i>	<i>p1</i>	<i>p2</i>	<i>p3</i>	<i>p4</i>	<i>p5</i>	<i>p6</i>	<i>p7</i>	<i>p8</i>	<i>p9</i>	
20	<i>il</i>	0.079662	9	0.229480	82.406889	65					82.4068892282175
21	<i>il</i>	0.061983	9	0.226531	143.478728	65					143.47872767421208
22	<i>il</i>	0.132895	9	0.214834	198.454305	65					198.45430487658496
23	<i>il</i>	0.030740	9	0.192215	249.810974	65					249.81097439559304
24	<i>il</i>	0.048863	9	0.197297	298.634187	65					298.63418729776294
25	<i>il</i>	0.029146	9	0.058766	345.529014	65					345.52901379770356
26	<i>il</i>	0.047138	9	0.009308	390.878652	65					390.87865236297296
27	<i>il</i>	0.020824	9	0.050324	434.946169	65					434.94616895527435
28	<i>il</i>	0.121356	9	0.104380	477.922556	65					477.92255590401237
29	<i>il</i>	0.143463	9	0.144840	519.952320	65					519.9523199430956
30	<i>il</i>	0.060143	9	0.019439	561.148322	65					561.1483217911464
31	<i>il</i>	0.122432	9	0.028231	601.600955	65					601.6009551934901
32	<i>il</i>	0.143606	9	0.022606	641.384119	65					641.3841186639341
33	<i>il</i>	0.137698	9	0.017053	680.559262	65					680.559261990032
34	<i>il</i>	0.044343	9	0.008280	719.178222	65					719.1782217192105
35	<i>il</i>	0.098248	9	0.001501	757.285265	65					757.2852647750104
36	<i>il</i>	0.119511	9	0.005630	794.918597	65					794.9185970819026
37	<i>il</i>	0.145157	9	0.009810	832.111501	65					832.1115005083225
38	<i>il</i>	0.035393	9	0.039311	868.893205	65					868.8932052694906
39	<i>il</i>	0.110284	9	0.022870	905.289570	65					905.2895700271768
40	<i>il</i>	0.101227	9	0.004826	941.323620	65					941.3236195697617
41	<i>il</i>	0.113615	9	0.017879	977.015975	65					977.0159752559147
42	<i>il</i>	0.143117	9	0.010669	1012.385204						1012.3852035090151
43	<i>il</i>	0.200000	9	0.767317							[...]

Figura 62 – Suono 'celldist', sezione *iB*, prima sottosezione. Confronto tra le frequenze in *Csound* (a sinistra) e le frequenze dello spettro distorto *S* definite dalla variabile `*distchds*` (a destra).

4.2.4.3.2 Suoni 'V91' e 'V04'

Anche i suoni 'V91' e 'V04' sono stati ottenuti tramite sintesi per forma d'onda variabile in ambienti *Patchwork* e *Csound*. In questi casi, però, non è stato impiegato alcun modello

¹⁸³ Comunicazione privata con Laurent Pottier.

strumentale. Lo studio delle patch e dei file di *Csound* fa chiarezza sul loro contenuto armonico.

I suoni di tipo 'V91' sono ancorati alle frequenze definite dagli aggregati T:

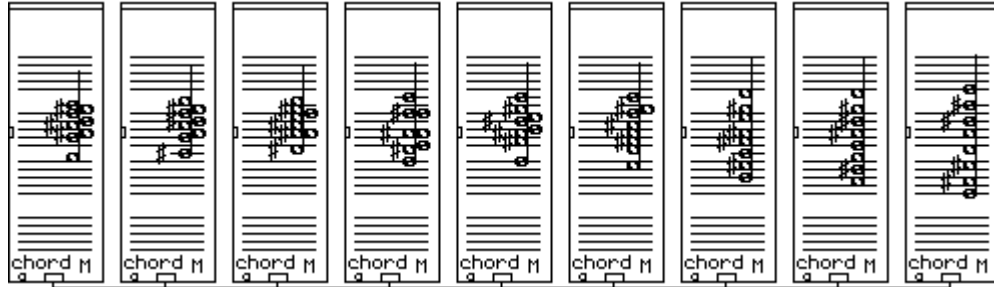


Figura 63 – I nove aggregati T. Patch di Patchwork.

Lo studio della patch di *Patchwork* utilizzata per la loro generazione (figura 64) mostra come le frequenze dell'aggregato T siano sottoposte a moltiplicazione secondo due diversi fattori: 1,0006 e 1,001 (moduli evidenziati in blu). I file score generati comprendono quindi:

- Le frequenze dell'aggregato T;
- le stesse moltiplicate per 0,0006;
- le stesse moltiplicate per 0,001.

Ciò si traduce in complessità e ricchezza del suono.

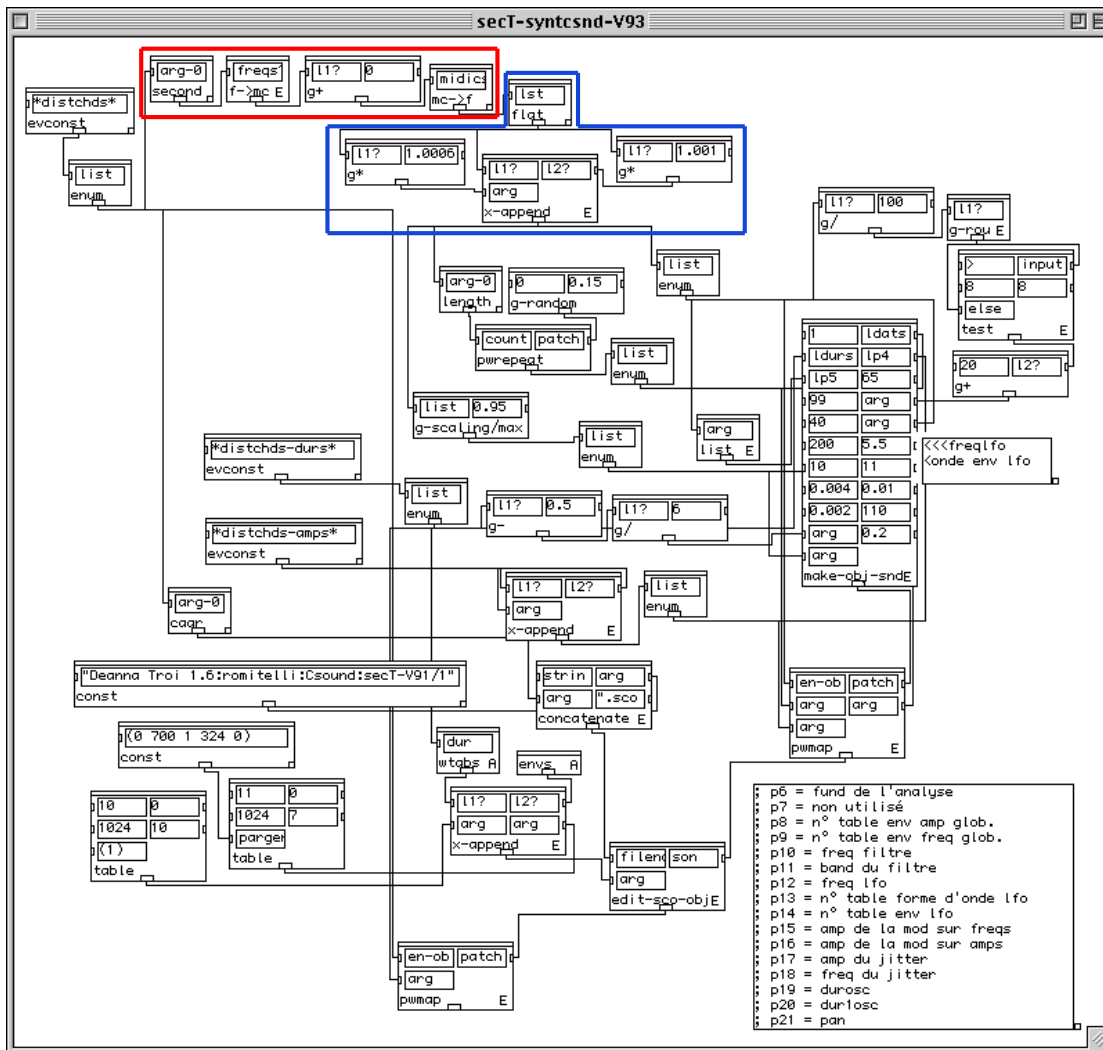


Figura 64 – Patch per la scrittura dei file di Csound per i suoni di tipo 'Vqr'.

Per la generazione di file audio viene utilizzato un codice *Csound*, sviluppato originariamente nel maggio del 1992 da Pottier e modificato per l'occasione nel 1995:

```

; inizializzazione
sr = 44100
kr = 4410
ksmps = 10
nchnls = 2

; definizione dello strumento
instr 1

; definizione di ampiezza e frequenza del suono
iamp = p4 * 22000
ifreq = p5

; oscillatore principale
kfreq oscill 0, p5/p6, p3, p9 ; frequenza
    
```

```

kenv oscill    0, 1, p3, p8 ; inviluppo gestito da oscillatore

; LFO
klfoenv oscill 0, 1, p3, p14
klfo  oscil klfoenv, p12, p13
kmodf = p15/2 * klfo
kmoda = p16 * klfo

; componenti aleatorie
kalea1 randi    p17, p18
kalea2 randi    p17, p18*3
kalea3 randi    p17, p18*10
kalea = kalea1 + kalea2 + kalea3

; frequenza risultante
kfreq = kfreq * (1 + kmodf) * (1 + kalea)
; Inviluppo globale
kenv = kenv * iamp * (1 + kmoda)

; Inviluppo delle singole parziali
kenv1 linseg 1, p20, 1, p19, 0, p3-p19-p20, 0
kenv2 linseg 0, p20, 0, p19, 1, p19, 0, p3-(2*p19)-p20, 0
kenv3 linseg 0, p19+p20, 0, p19, 1, p19, 0, p3-(3*p19)-p20, 0
kenv4 linseg 0, 2*p19+p20, 0, p19, 1, p19, 0, p3-(4*p19)-p20, 0
kenv5 linseg 0, 3*p19+p20, 0, p19, 1, p19, 0, p3-(5*p19)-p20, 0
kenv6 linseg 0, 4*p19+p20, 0, p19, 1, p19, 0, p3-(6*p19)-p20, 0
kenv7 linseg 0, 5*p19+p20, 0, p19, 1, p3-(6*p19)-p20, 1

; definizione del suono risultante come sintesi additiva di
; sette frequenza con i relativi inviluppi
aoscl oscil kenv*kenv1, kfreq, 1
aoscl2 oscil kenv*kenv2, kfreq, 2
aoscl3 oscil kenv*kenv3, kfreq, 3
aoscl4 oscil kenv*kenv4, kfreq, 4
aoscl5 oscil kenv*kenv5, kfreq, 5
aoscl6 oscil kenv*kenv6, kfreq, 6
aoscl7 oscil kenv*kenv7, kfreq, 7
aoscl = aoscl1 + aoscl2 + aoscl3 + aoscl4 + aoscl5 + aoscl6 + aoscl7

; filtraggio finale
afil  reson aoscl, p10, p11, 1
afil  balance afil, aoscl

; definizione delle uscite nei due canali del file stereo
outs1 afil*sqrt(1-p21)
outs2 afil*sqrt(p21)

endin

```

Il suono risultante è definito dalla somma di sette oscillatori distinti. È interessante notare come la frequenza delle singole parziali che compongono il suono sia ‘disturbata’ da una componente aleatoria e da un oscillatore a bassa frequenza (LFO) che in questo caso crea

un effetto sia di vibrato che di tremolo. In questo modo la sintesi è resa complessa da due fattori: da una, seppur limitata, indeterminazione frequenziale e dall'utilizzo di tabelle di generazione elaborate.

Come è possibile dedurre dai file *score*, infatti, le tabelle di generazione variano per grado di complessità. Alcune funzioni sono semplici, come ad esempio:

```
f10  0      1024 10    1
```

dove viene definita la funzione f10, creata all'istante 0, descritta da 1024 valori e che utilizza la routine di generazione 10: una semplice senoide senza componenti armoniche.

Più complessa invece la tabella generata dalla funzione f40:

```
f40  0      1024 -7    64.600000 29    65.060000 30    65.050000
      29    65.060000 88    65.060000 29    65.070000 29
      65.070000 59    65.060000 29    65.060000 88    65.060000
      58    65.070000 59    65.060000 58    65.060000 29
      65.060000 30    65.060000 29    65.060000 29    65.060000
      146   64.990000 30    64.990000 58    64.990000 29
      64.970000 30    65.040000 29    61.810000
```

creata all'istante 0, descritta da 1024 valori e che utilizza la routine di generazione 7,¹⁸⁴ una spezzata definita dalle coppie di valori successivi (ordinata e proiezione del segmento in ascissa).

La funzione utilizzata per l'involuppo del filtro LFO è molto semplice:

```
f11  0      1024 7     0     700  1     324  0
```

dove viene definita la funzione f11, creata all'istante 0, descritta da 1024 valori e che utilizza la routine di generazione 7 (spezzata).

I parametri di generazione del suono sono poi così impostati:

```
i1   0.071861  9     0.229480  246.941651 65    99    22    40
      246.941651 200  5.500000  10    11    0.004000  0.010000
      0.002000  110  1.416667  0.200000  0.266318
```

dove lo strumento i1 inizia a suonare all'istante 0.071861 per 9 secondi, con un'ampiezza iniziale di 0.229480 (valore assoluto su una scala da 0 a 1) e una frequenza fondamentale di 246.941651 (Si₃). I parametri seguenti controllano:

¹⁸⁴ Il valore negativo indica che la tabella non viene scalata a 1.

- 65 = il divisore da utilizzare per definire la frequenza con cui leggere la tabella di generazione;
- 99 = non utilizzato;
- 23 = la tabella da utilizzare per l'involuppo di ampiezza (f_{23} in questo caso);
- 40 = la tabella da utilizzare per la definizione delle frequenze (f_{40} in questo caso);
- 246.941651 = la frequenza centrale del filtro passa banda;
- 200 = la larghezza di banda del filtro;
- 5.500000 = la frequenza per l'effetto LFO;
- 10 = la tabella da utilizzare per il controllo dell'effetto LFO (f_{10} una funzione sinusoidale in questo caso);
- 11 = la tabella da utilizzare per il controllo dell'ampiezza dell'effetto LFO (f_{11} è una funzione coseno in questo caso);
- 0.004000 = l'ampiezza della modulazione in frequenza dell'effetto LFO;
- 0.010000 = l'ampiezza della modulazione in ampiezza dell'effetto LFO;
- 0.002000 = l'ampiezza della componente aleatoria (jitter);
- 110 = la frequenza con cui vengono generati numeri casuali per la componente aleatoria;
- 1.416667 = la durata della prima parziale del suono sintetizzato;
- 0.200000 = la durata della seconda parziale del suono sintetizzato,¹⁸⁵
- 0.266318 = la distribuzione nello spazio stereofonico.

Nella cartella che contiene i file 'V91' sono presenti però 18 file *score*, che generano due suoni per ogni sottosezione, differenziati esclusivamente da una trasposizione di un ottavo di tono nelle frequenze delle componenti parziali. Come avviene questa trasposizione? Osservando la patch di *Patchwork* riportata in figura 64 si può notare la presenza di un blocco dedicato alla trasposizione intervallare in cent (evidenziato in rosso). È sufficiente inserirvi il valore 25 (un ottavo di tono in cent) e rivalutare l'intera patch per generare altri file *score*. Suoni con frequenze derivate dal modello e suoni trasposti saranno poi sovrapposti, come vedremo, nel montaggio finale per creare effetti di battimento.

I suoni 'V04' provengono invece da una selezione di frequenze tratte dagli spettri distorti S (figura 65).

¹⁸⁵ La versione modificata del file *orchestra* prevede sette oscillatori al posto di due. Viene pertanto utilizzato esclusivamente il parametro precedente per controllare la durata di tutti gli oscillatori. Tuttavia esso rimane nei file *score* anche se la sua funzione è nulla.

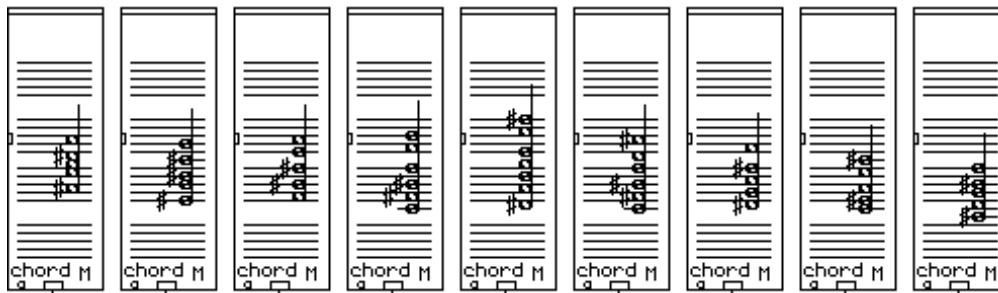


Figura 65 – Selezione delle componenti dai nove spettri S , approssimate al semitono.
Patch di Patchwork.

Alla base della generazione dei suoni 'Vo₄' vi è lo stesso codice utilizzato per la generazione dei suoni 'V₉₁', ulteriormente modificato. In particolare, la sintesi qui si limita alla somma di tre oscillatori basati sempre su tabelle di generazione complesse:

```

; inizializzazione
sr = 44100
kr = 4410
ksmps = 10
nchnls = 2

; definizione dello strumento
instr 1

; definizione di ampiezza e frequenza del suono
iamp = p4 * 32767
ifreq = p5

; oscillatore principale
kfreq oscil1 0, p5/p6, p3, p9 ; frequenza
kenv oscil1 0, 1, p3, p8 ; inviluppo gestito da oscillatore

; LFO
klfoenv oscil1 0, 1, p3, p14 ; lfo
klfo oscil klfoenv, p12, p13 ; lfo
kmodf = p15/2 * klfo
kmoda = p16 * klfo

; componenti aleatorie
kalea1 randi p17, p18
kalea2 randi p17, p18*3
kalea3 randi p17, p18*10
kalea = kalea1 + kalea2 + kalea3

; frequenza risultante
kfreq = kfreq * (1 + kmodf) * (1 + kalea)

; inviluppo globale
kenv = kenv * iamp * (1 + kmoda)

```

```

; inviluppo delle singole parziali
kenv1 linseg    1, p19, 0, p20, 0
kenv2 linseg    0, p19, 1, p20, 0
kenv3 linseg    0, p19, 0, p20, 1

; definizione del suono risultante come sintesi additiva di tre
; frequenze con i relativi inviluppi
aoscl oscili    kenv*kenv1, kfreq, 1
aoscl2 oscili   kenv*kenv2, kfreq, 2
aoscl3 oscili   kenv*kenv3, kfreq, 3
aoscl = aoscl + aoscl2 + aoscl3

; filtraggio finale
afil  reson aoscl, p10, p11, 1

; definizione delle uscite nei due canali del file stereo
outs1 afil*sqrt(1-p21)
outs2 afil*sqrt(p21)

endin

```

Osservando le tabelle di generazione, la funzione utilizzata per il controllo delle frequenze è anche in questo caso la *f40*, con alcune modifiche rispetto a quella vista in precedenza:

```

f40  0      1024  -7    64.600000  29    65.060000  29    65.050000
      30    65.060000  87    65.060000  29    65.070000  29
      65.070000  59    65.060000  29    65.060000  87    65.060000
      59    65.070000  58    65.060000  58    65.060000  30
      65.060000  29    65.060000  29    65.060000  29    65.060000
      146   65.030000  26    65.030000  59    65.050000  29
      65.030000  32    65.040000  32    65.020000

```

I parametri di generazione del suono sono strutturati analogamente a quelli utilizzati per 'V9r' e vengono definiti nei file *score*.

Anche per i suoni 'V04' è previsto un effetto di battimento. In questo caso però esso non è generato nel montaggio, bensì creato direttamente all'interno della patch di *Patchwork* utilizzata per la generazione dei file *score*. Come si può notare in figura 66 (parte evidenziata), analogamente a quanto avviene per i suoni 'V9r', le frequenze selezionate dallo spettro *S* vengono sottoposte a moltiplicazione, per generare altre componenti che andranno ad arricchire il suono. I fattori in questo caso sono tre: 1.01, 1.001 e 1.0003. L'ultimo in particolare produce frequenze con uno scarto tale da ricadere, nella sovrapposizione con il suono originale, all'interno della banda dei battimenti.

sfasamento invece ha una duplice funzione: aumentare la durata del suono, specialmente nel caso di 'celldist', e trasformarlo grazie alla simulazione di un filtro *comb*.¹⁸⁶

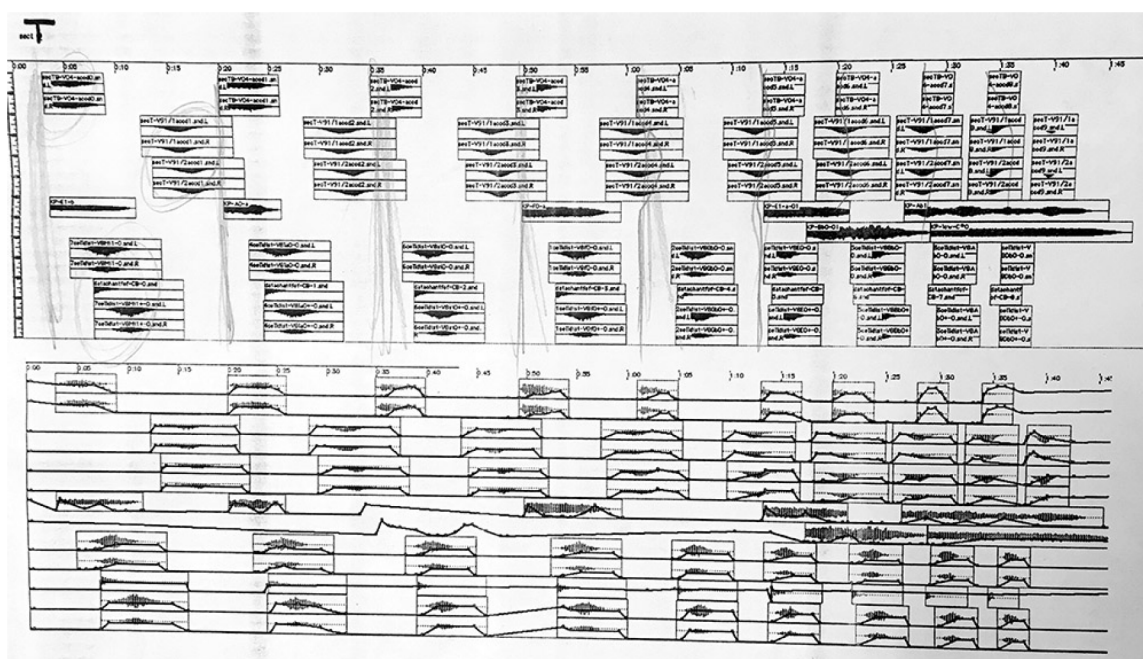


Figura 67 – Screenshot del progetto in Pro Tools per il montaggio della parte elettronica della sezione *IB*. Si notino nella parte superiore i file con le rispettive forme d'onda, in quella inferiore le automazioni in volume. Fotocopia conservata da Laurent Pottier.

Una volta ottenuta una singola traccia, questa è stata infine spazializzata nei quattro canali previsti per la riproduzione in concerto.

4.2.6 Simulazione del processo di montaggio

Il confronto tra la parte elettronica licenziata dal compositore e i file audio contenuti nel CD, agevolato dalle informazioni sulle sessioni in *Pro Tools*, ha permesso di simulare in un

¹⁸⁶ La tecnica della duplicazione e dello sfasamento è comunemente impiegata nella post-produzione di musica commerciale. Mira a creare sonorità nuove grazie alla simulazione 'manuale' di un filtro *comb*. I filtri *comb*, infatti, sono caratterizzati da una risposta in frequenza essenzialmente piatta su tutto lo spettro fatta eccezione per un intorno, più o meno ampio, di una frequenza ω_0 . In questo punto, la risposta in frequenza presenta una forte variazione che dà luogo a un picco o a un minimo in funzione del quale si parla rispettivamente di *notching comb filter* o *peaking comb filter*. In generale, questo comportamento può essere esibito in corrispondenza di n frequenze ω .

ambiente tecnologico differente il processo di montaggio e mixaggio della parte elettronica della sezione **iB**.

Gli esempi di seguito riportati mostrano il dettaglio dei primi 34 secondi della sezione che corrispondono alle prime due sottosezioni, quella sullo spettro del Mi_r (S^1) e quella sullo spettro del La_o (S^2). La figura 68 mostra la sezione ricostruita in ambiente *Adobe Audition*.

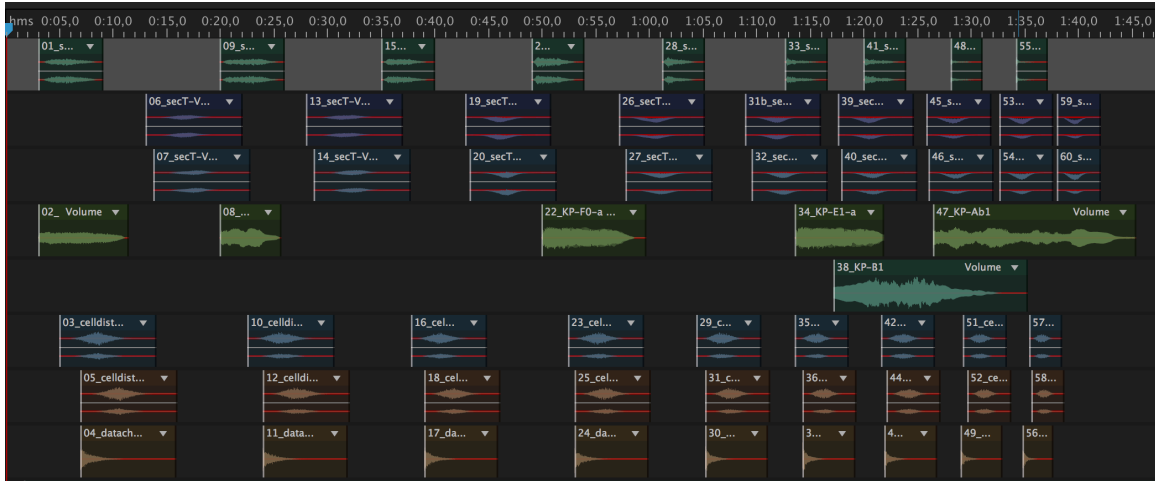


Figura 68 – Simulazione del montaggio della parte elettronica della sezione **iB**.

La possibilità di studiare i singoli suoni che compongono la parte elettronica offre all'analista un'opportunità eccezionale. La natura irreversibile dei processi di mixaggio, infatti, non consente di discernere con precisione gli elementi costitutivi, rendendo di fatto impossibile una loro precisa individuazione e analisi. Il nuovo montaggio consente invece di indagare non solo la natura dei singoli suoni, ma anche le relazioni tra la componente elettronica la partitura. In particolare è possibile far luce sulle annotazioni che Romitelli fornisce nel sistema dedicato alla trascrizione della parte elettronica. Si constata infatti una perfetta corrispondenza tra le note ricavate dalle patch in *Patchwork* e quelle indicate in partitura (figura 57).

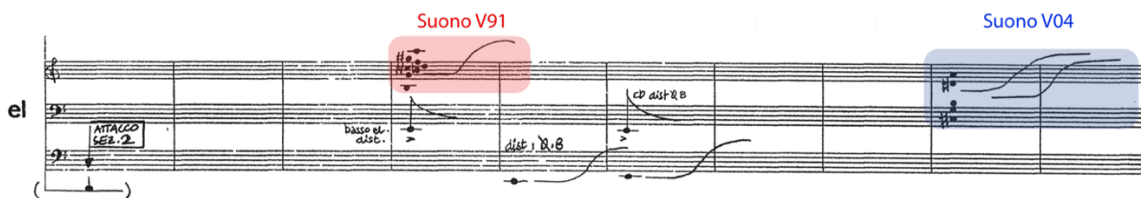


Figura 69 – FAUSTO ROMITELLI, *EnTrance*, batt. 41-50. **iB** – Prima sottosezione. Dettaglio dei suoni 'V91' e 'V04'. In rosso l'aggregato T^1 (suono di sintesi 'V91', in blu la selezione dallo spettro distorto S^1 (suoni di sintesi 'V04').

Se però si va ad analizzare il montaggio – coerente con le schermate di *Pro Tools* conservate da Pottier e quindi analogo al montaggio dell'epoca – ci si accorge immediatamente di alcune incongruenze che affliggono l'intera sezione **IB**. Sembra infatti che tutti i suoni 'V91' e 'V04' siano stati tra loro invertiti di posizione (figura 70).

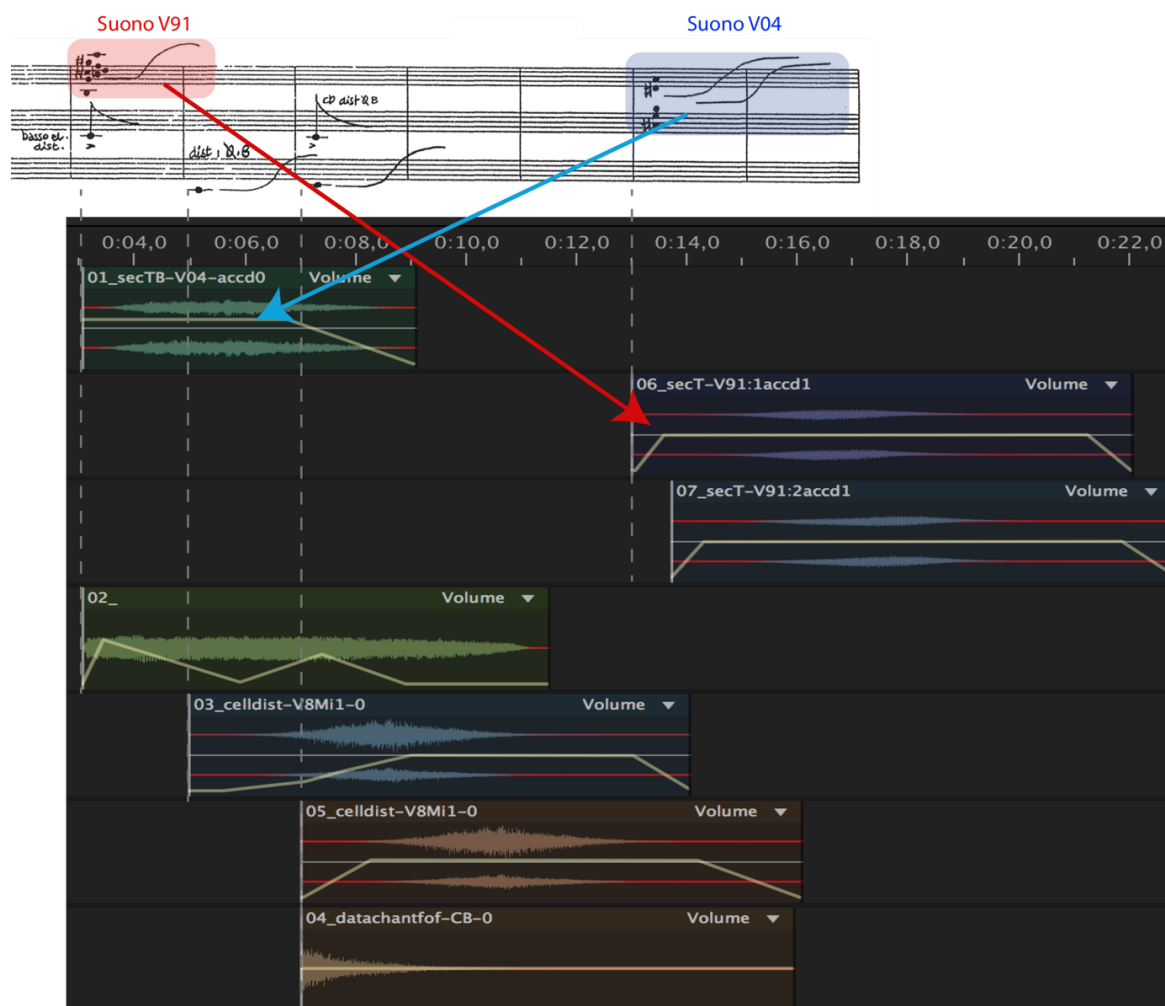


Figura 70 – inversione della posizione dei suoni 'V91' e 'V04' nel montaggio.

Com'è possibile giustificare tale deviazione dal modello compositivo? Si pongono solo due casi: ripensamento ed errore di montaggio.

Nel primo caso saremmo di fronte a uno scenario in cui Romitelli, dopo aver riportato i riferimenti all'elettronica in partitura, ha deciso di invertire i suoni, lasciandosi guidare dall'ascolto e (forse) da una migliore resa acustica del montaggio. Il fatto non stupirebbe. Lo studio di registrazione ha mutato drasticamente il rapporto con la scrittura musicale: il compositore è stato proiettato all'interno di un sistema complesso, è diventato interprete e ascoltatore di sé stesso, osservatore attivo che determina vincoli e che aggiunge proprietà

emergenti capaci di retroagire sulle parti. Si tratta di proprietà osservabili solo empiricamente, non direttamente deducibili dallo studio riduzionistico delle singole parti e delle loro relazioni, né sempre prevedibili da materiali preparatori, schizzi, abbozzi e partitura. Il suono nasce spesso in un modo non previsto, perché i sistemi di diffusione audio creano un continuo feedback d'ascolto che permette di 'tarare' le apparecchiature, modificarne parametri e configurazioni: il sistema si auto-ri-organizza a ogni istante, in una pulsione 'viva' che può far persino mutare l'idea compositiva originaria. La linearità del modello comunicativo viene stravolta e, se la semiologia musicale di Jean-Jacques Nattiez aveva già riposizionato le frecce del modello della teoria della comunicazione da emittente → messaggio → ricevitore a poetica → livello neutro ← estetica, l'osservatore attivo costringe ora a riorganizzare la disposizione delle frecce a feedback, a creare una retroazione continua tra i diversi piani, che coinvolge spesso anche il fruitore dell'opera.¹⁸⁷

Non è possibile però ignorare alcuni elementi che rendono poco plausibile questa tesi. Innanzi tutto lo studio delle fonti notate – e in particolare delle stratificazioni nelle partiture d'uso conservate presso l'IRCAM – non dà adito a dubbi nell'attribuire l'inserimento dei riferimenti alla parte elettronica a uno stadio di redazione molto avanzato, prossimo alla versione licenziata. In particolare, le stesure utilizzate per la prima esecuzione sono completamente prive di riferimenti.¹⁸⁸ Inoltre, nei suoi scritti e nelle presentazioni di *EnTrance* l'autore ha sempre sottolineato la sua ricerca della perfetta coerenza armonica tra voce, ensemble e parte elettronica. In particolare, nel testo di presentazione dell'opera si legge:

En ce qui concerne la relation entre la voix, les instruments et l'électronique, nous avons, d'une part, une parfaite cohérence du point de vue du matériel harmonique, les fréquences étant les mêmes dans les sons de synthèse et dans l'ensemble.

È innegabile che l'inversione dei suoni nel montaggio mini alla base questa coerenza armonica nell'intera sezione **IB**. Dobbiamo quindi orientarci verso l'attribuzione di questa incongruenza a un errore materiale verificatosi durante montaggio?

La questione è spinosa. Anche in considerazione del fatto che Romitelli ha licenziato la parte elettronica con l'inversione. Tuttavia è possibile proporre una ricostruzione, ove i suoni compaiono nella posizione indicata in partitura, che potrà essere vagliata anche in sede esecutiva nel rapporto con la parte strumentale.

¹⁸⁷ Si veda LUCA COSSETTINI, *Opere chiuse in sistemi aperti. Autopoiesi nella musica elettronica*, Lucca, LIM, 2013.

¹⁸⁸ Si veda la recensione delle fonti, *infra*, par. A.2.2.1.

4.3 L'esperimento di ricostruzione della parte 1B

Per affrontare la ricostruzione della parte elettronica è necessario innanzi tutto superare alcuni problemi:

- File mancanti. Alcuni file audio non sono stati preservati. Si tratta di nove suoni di sintesi, uno per ogni sottosezione, più un suono concreto di basso elettrico distorto che assume il ruolo di coda della sezione. Nel caso del suono concreto, la sua assenza costituisce a tutti gli effetti una lacuna; in tutti gli altri casi, Pottier ha conservato i file *orchestra* e *score* utilizzati per la sintesi in *Csound*. È stato quindi possibile rigenerare i suoni mancanti con *Csound* 1.0 su workstation *PowerMac G4*.¹⁸⁹
- Automazioni in volume. Le automazioni in volume rappresentate graficamente nella schermata di *Pro Tools* sono indicative, e non preservano quindi gli esatti valori in *dB*. La ricostruzione degli involuppi si è pertanto fondata anche su basi percettive;
- Spazializzazione. Se le automazioni in volume sono tramandate nello schema di *Pro Tools*, seppur sommariamente, ciò non si può dire per il panning, ossia la spazializzazione quadrifonica del mixaggio. Essa potrà quindi essere ricostruita esclusivamente per via percettiva e attraverso il confronto con la forma d'onda delle quattro tracce originali.

L'assenza tra i materiali preparatori del suono di basso elettrico rende purtroppo impercorribile la ricostruzione della parte elettronica per semplice mixaggio. È possibile però sviluppare altre strategie. L'ipotesi operativa è stata quella di individuare strumenti in grado, se non di deconvolvere dal segnale mixato il suono mancante, quantomeno di isolarlo nelle bande frequenziali di sua pertinenza. L'esperimento di estrazione del suono si è basato sul secondo canale della parte quadrifonica mixata. Qui, infatti, il basso elettrico si presenta con intensità maggiore (figura 71).

¹⁸⁹ La sintesi in *Csound* può essere effettuata anche con le più recenti versioni del programma, a patto di accorpate i file *orchestra* e *score* in un unico file con estensione 'csd'.

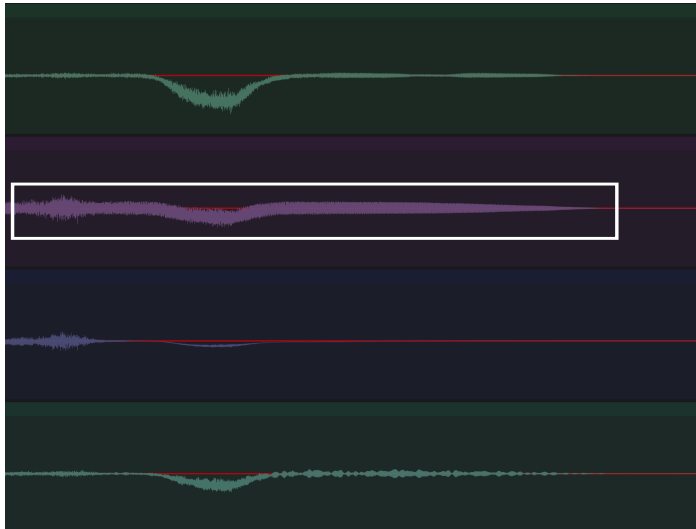


Figura 71 – Rappresentazione in forma d'onda della parte finale della sezione. Si evidenzia in traccia 2 il suono di basso elettrico distorto. In traccia 3 il suono è del tutto assente.

Si tratta ora di discernere i suoni. Tuttavia, data la natura tra loro simile dei suoni di sintesi, l'osservazione del sonogramma anche se coadiuvata dall'ascolto non è esplicativa. Impercorribile pertanto una categorizzazione su basi spettromorfologiche. È solo la conoscenza precisa dei parametri di generazione a rendere possibile il riconoscimento.

Ad esempio, sappiamo dallo studio delle patch che il suono 'datachantfof-cb' è caratterizzato da un transitorio d'attacco impulsivo, il suo inizio è quindi chiaramente individuabile al secondo 2 del sonogramma in figura 72 (riquadro 1); le sue componenti parziali si mescolano poi con quelle del suono di basso elettrico. Il suo carattere impulsivo, che si traduce nel dominio frequenziale in energia a banda larga, lo rende difficilmente eliminabile senza compromettere lo spettro del suono di basso elettrico.

Anche il suono 'celldist' presenta caratteristiche che lo rendono inadatto alla rimozione: ha un andamento dinamico a doppia forcilla; inoltre, la sua densità armonica lo rende complesso e difficilmente scomponibile nelle sue parziali, anche utilizzando una finestrazione con un elevato numero di punti (figura 72, riquadro 2). Siamo qui di fronte a un classico problema dell'analisi sonografica, riconducibile al principio di indeterminazione tempo/frequenza: come individuare con pari precisione attacco ed estinzione del suono, qui gradualmente, e il suo contenuto armonico?

Per la rimozione di questi due suoni ci scontra quindi con una difficoltà teorica e operativa insormontabile allo stato attuale dello sviluppo degli algoritmi di elaborazione del segnale audio: una loro rimozione per filtraggio e/o risintesi andrebbe a modificare sensibilmente anche il suono di basso elettrico originale.

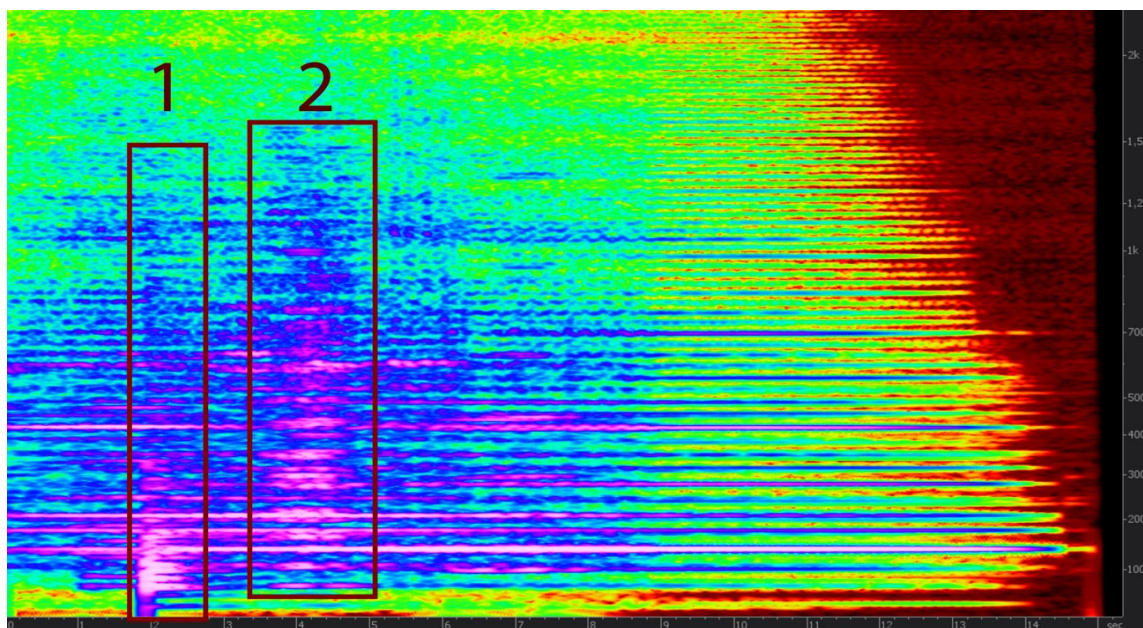


Figura 72 – Spettrogramma di traccia 2 con evidenziati: 1) il suono ‘datachantfof-cb’; 2) il suono ‘celldist’. Analisi effettuata con iZotope RX 4. Multiresolution, finestra base di 4096 punti.

Il suono ‘Vo₄’ non è confluito in questo caso in traccia 2 e pertanto non va rimosso. Paradossalmente su un piano spettromorfologico, il suono ‘V_{9r}’, di primo acchito quasi ‘invisibile’ nel sonogramma, non pone particolari problemi: sebbene il suo involuppo sia simile a quello dei suoni ‘celldist’, il suo contenuto armonico è meno complesso e quindi, note le frequenze delle parziali che lo compongono e la loro ampiezza teorica, sono facilmente individuabili anche nel complesso di un’analisi FFT (figura 73).

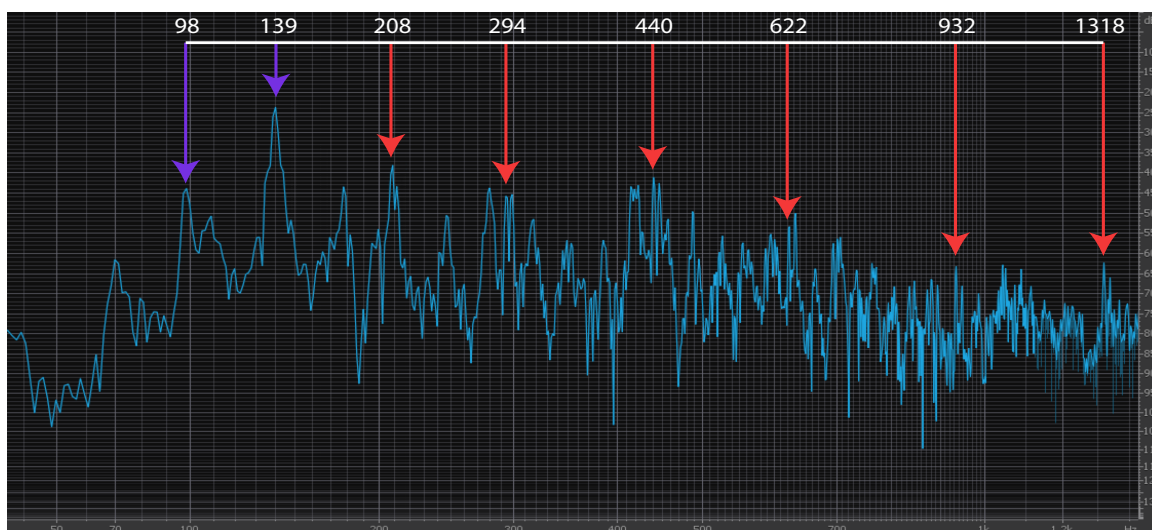


Figura 73 – Analisi FFT con il dettaglio delle componenti parziali di ‘V_{9r}’. In viola, le frequenze che vanno a sovrapporsi con quelle definite dallo spettro S⁹ o suoi multipli.

Con gli algoritmi oggi implementati in software come ad esempio *iZotope RX* è possibile pertanto rimuoverli senza intaccare percettivamente il suono del basso elettrico; nei casi più complessi, ove le parziali dei due suoni si sovrappongano o siano in una banda frequenziale molto prossima, è possibile attenuarli fino al di sotto della soglia di mascheramento (figura 74).

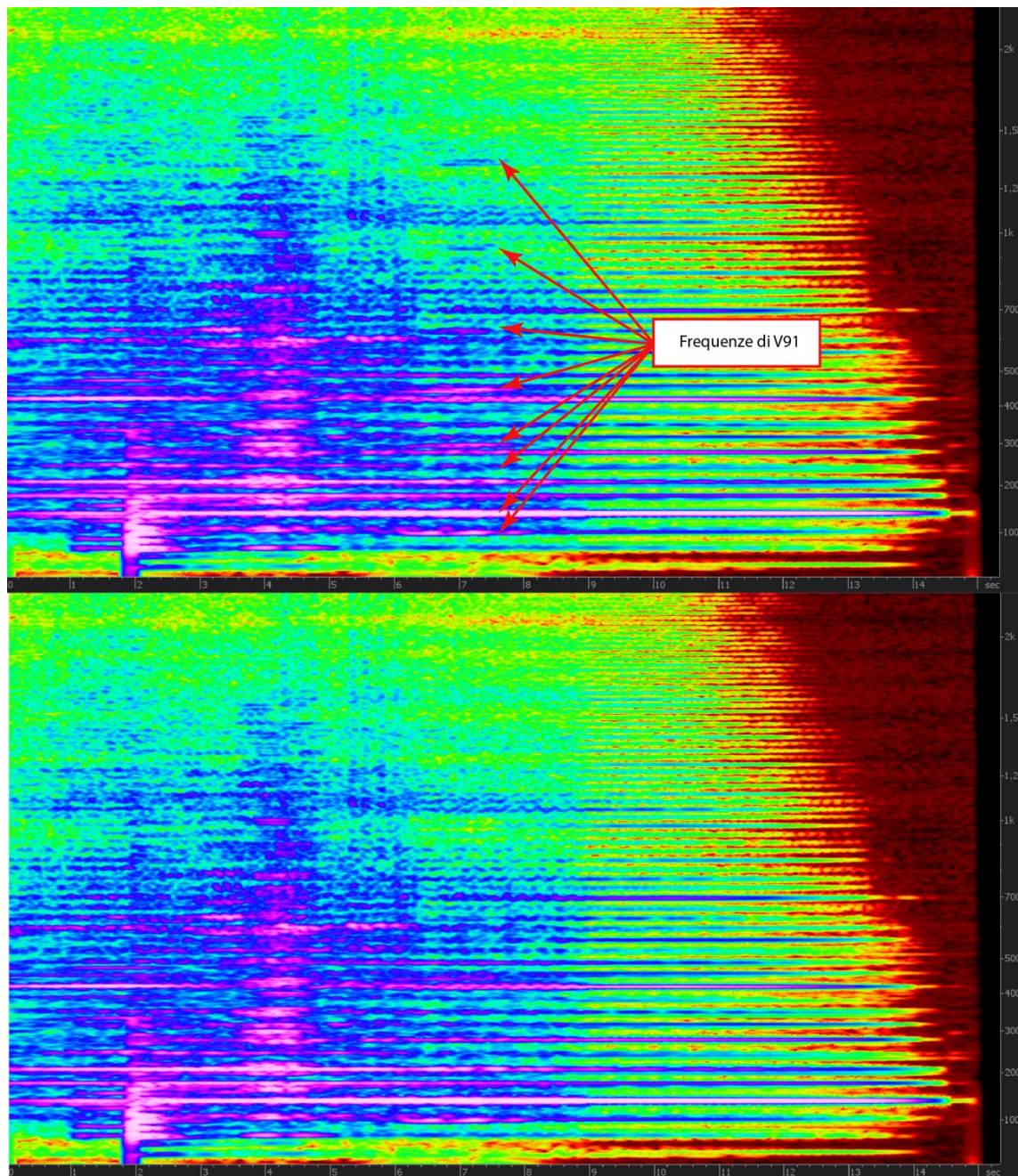


Figura 74 – Dettaglio del suono 'V91' prima e dopo la sua rimozione con *iZotope RX 4*.

Con queste procedure di editing, analoghe alle pratiche del restauro audio, si è ottenuto così un ‘nuovo’ suono di basso elettrico (da qui in poi indicato con ‘NB’) da poter utilizzare nel montaggio (figura 75).

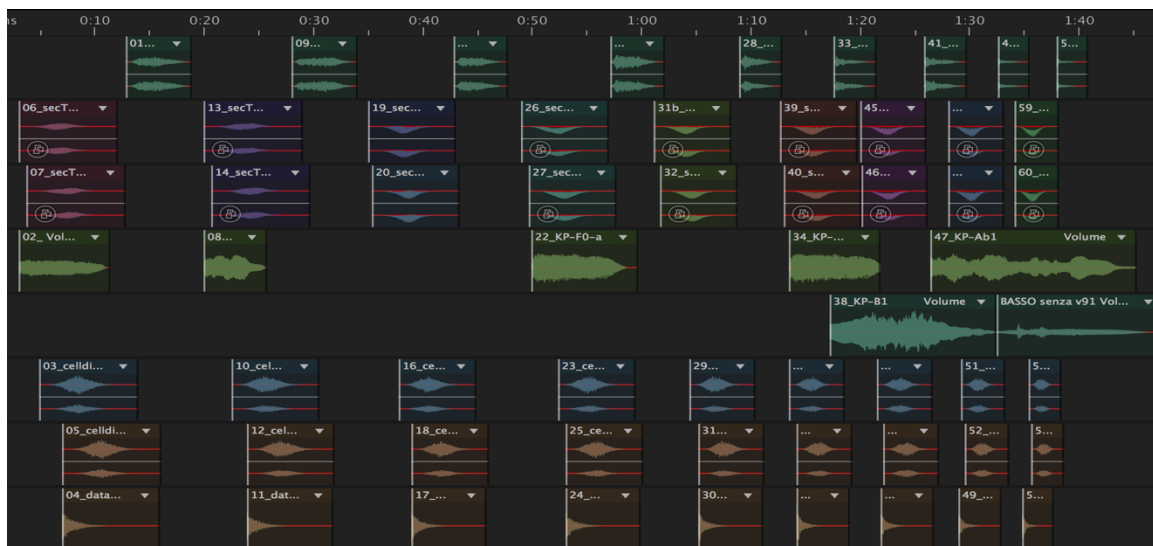


Figura 75 – Nuovo montaggio della sezione 1B. Si noti la presenza, in coda alla sezione, del suono di basso elettrico distorto, assente nella prima simulazione a figura 68. Si noti inoltre l’inversione dell’ordine di comparsa dei suoni ‘Vo4’ e ‘V91’.

La presenza in ‘NB’ dei suoni ‘celldist’ e ‘datachantfof-cb’, seppur a bassa intensità, impone però degli accorgimenti ulteriori nelle fasi di ricostruzione delle automazioni in volume e della spazializzazione. Se per le automazioni in volume possiamo avvalerci degli appunti, seppur sommari, di Pottier, per la spazializzazione non esistono schemi operativi e l’indagine va condotta su basi percettive, coadiuvate da analisi in tempo/ampiezza. Come emerge in figura 76, la distribuzione del suono nello spazio nella sezione in esame è generalmente stereofonica su 4 tracce, salvo elaborazioni più complesse come nel caso del finale di sezione, *locus criticus* della nostra proposta di ricostruzione.

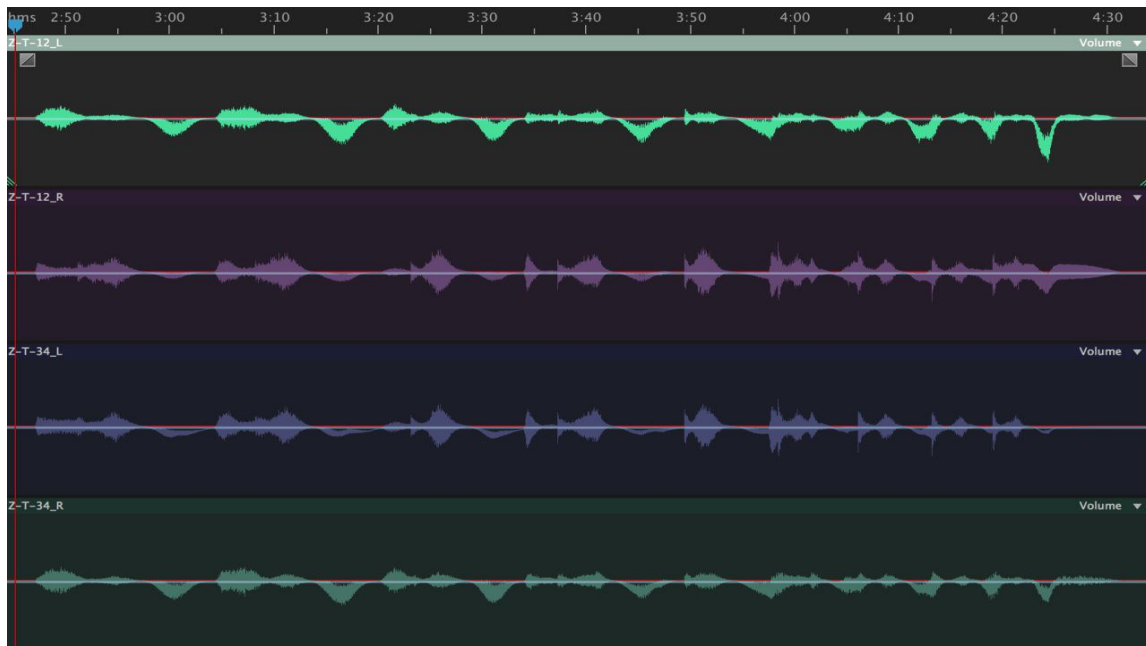


Figura 76 – Sezione 1B. Distribuzione del materiale sonoro nelle quattro tracce.

Per stimare le variazioni di volume nelle singole tracce, utile è stato lo sviluppo di uno script *MATLAB* per il tracciamento dell'involucro del suono condotto su finestrata del segnale liberamente variabile che permette quindi di confrontare analisi condotte con diversi gradi di risoluzione:

```
function [ out1 ] = involuppo( signal, N )
%involuppo: Suddivide il segnale (signal) in un numero (N) di
%segmenti e ne calcola l'RMS.

% Inizializzazione
L=length(signal) % Ricava la lunghezza in punti del segnale
SP=1 % Permette la sovrapposizione delle finestre (<1)

%Imposta la finestra per la segmentazione
W=fix(L/N)
SP=W.*SP
Window=hamming(W);
Window=Window(:);

%Segmenta il segnale e calcola l'RMS

Index=( repmat(1:W,N,1)+repmat((0:(N-1))*SP,1,W))';
hw=repmat(Window,1,N);
Seg=signal(Index).*hw;
rmsseg=rms(Seg);

% Calcola il rapporto Segnale/rumore
% (non utilizzato in questa sede)
```

```

SNR=db(max(rmsseg)) - db(min(rmsseg))

% restituisce la serie di valori RMS calcolati

out1=rmsseg;

end

```

In traccia 2 le automazioni dei suoni ‘celldist’ e ‘datachantfof-cb’ sono già integrate nel suono ‘NB’; è stato quindi sufficiente ripristinare le automazioni per i suoni ‘Vo4’ e ‘Vgr’. La ricostruzione degli equilibri dinamici nelle altre tracce è stata più complessa. È necessario, infatti, tenere in considerazione che il suono ‘NB’ deve essere fatto confluire anche in traccia 1 e traccia 4, causando un seppur minimo squilibrio nelle ampiezze nel momento in cui le sue componenti ‘celldist’ e ‘datachantfof-cb’ si sommano in fase con i suoni ‘celldist’ e ‘datachantfof-cb’ originali ivi direttamente mixati.

Per ovviare al problema si è quindi deciso di pesare l’incidenza del suono residuo di ‘celldist’ e ‘datachantfof-cb’ nel nuovo suono ‘NB’. A tal fine è venuto in aiuto un algoritmo di restauro in *MATLAB* sviluppato dal Laboratorio MIRAGE dell’Università degli Studi di Udine per la ricostruzione di sezioni prive di segnale intenzionale (rumore di fondo) dei nastri magnetici analogici.¹⁹⁰ L’algoritmo prevede le seguenti fasi:

- Modellare parametricamente le differenti tipologie di rumore presenti nella sezione corrotta, in particolare il rumore da induzione elettromagnetica, il rumore di fondo, la diafonia, il rumore additivo deterministico;
- stimare, per ogni sezione da ricostruire, i parametri che meglio approssimano le caratteristiche dei vari rumori in essa presenti;
- ricostruire mediante modello le differenti tipologie di rumore, eccetto gli impulsi a bassa frequenza, che compongono ogni singola sezione;
- sostituire le sezioni del nastro prive di segnale intenzionale corrotte da impulsi a bassa frequenza con quelle ricostruite.¹⁹¹

In particolare, per l’esperimento qui presentato ci si è avvalsi del modulo per la stima della diafonia, modellata dall’equazione:

¹⁹⁰ L’algoritmo è stato utilizzato dal Laboratorio MIRAGE per il restauro della parte elettronica della *Fabbrica illuminata* di Luigi Nono. Per i dettagli si rimanda a Luca Cossettini (a cura di), *Luigi Nono. Studi, edizione, testimonianze*, Lucca, LIM, 2010.

¹⁹¹ Si veda ANTONIO RODÀ, “Ri-mediazione e ricostruzione di un tessuto sonoro: il restauro del nastro da concerto della *Fabbrica illuminata* di Luigi Nono”, in Luca Cossettini (a cura di), *Luigi Nono. Studi, edizione, testimonianze*, pp. 189-212.

$$d(t) = w_L \cdot s_L(t) + w_R \cdot s_R(t)$$

dove $s_L(t)$ e $s_R(t)$ sono i segnali che generano la diafonia – nel caso di una singola sorgente di diafonia $s_R(t)=0$ –, w_L e w_R sono termini costanti incogniti. La loro stima avviene mediante regressione lineare multipla sull'equazione

$$F[s(t)] = w_L \cdot F[s_L(t)] + w_R \cdot F[s_R(t)] + F[n(t)]$$

dove $n(t)$ è il rumore a larga banda caratteristico della traccia e precedentemente stimato, $F[s(t)]$ indica l'involuppo di ampiezza del segnale.

L'algoritmo è stato utilizzato due volte: dapprima ponendo una risintesi del suono 'celldist', con le debite automazioni in volume ricavate dallo schema di montaggio e dall'analisi del mixaggio, come sorgente di diafonia ($s_L(t)$); poi utilizzando una risintesi del suono 'datachantfof-cb', sempre modellata con le sue automazioni in volume. Ottenuto così il peso in dB delle loro componenti nel suono NB, se ne è potuto tenere conto nella regolazione dei parametri d'ampiezza in fase di spazializzazione. Si è ottenuta così infine una parte elettronica 1B completa e con i suoni 'Vo4' e 'V91' posizionati ove previsto dalla partitura.

CONCLUSIONI

EnTrance rappresenta il punto di approdo dell'esperienza di Fausto Romitelli all'IRCAM, in cui le diverse tecniche compositive, impiegate in opere precedenti come *Natura morta con fiamme* e *Mediterraneo I. Les idoles du soleil* con un approccio che potremmo definire ancora sperimentale, trovano non solo piena maturazione, ma soprattutto la possibilità di convivere in un pensiero musicale organico e originale.

All'IRCAM Romitelli cercava di mitigare un'opposizione sempre esistita in ogni scrittura musicale, ma che le tecniche di sintesi (siano esse elettroniche o strumentali) e la musica mista avevano reso evidente: da un lato la concorrenza di entità simultanee che determinano la percezione del timbro a partire da un processo di categorizzazione e proiezione nello spazio acustico, dall'altro la concatenazione di entità successive marcate da caratteristiche di discretizzazione e di organizzazione gerarchica del discorso.¹⁹² All'inizio degli anni Novanta i due mondi sembravano ancora inconciliabili, a maggior ragione perché, come fa notare Hugues Dufourt, anche nel secondo novecento, dopo l'illusione di una possibile democratizzazione lasciata intravedere dal processo di ricostruzione del pensiero musicale a seguito del secondo conflitto mondiale, si sono ristabiliti di nuovo poteri di denegazione simbolica reciproca per cui, «lunghi dal muoversi all'interno di un puro idealismo, il divenire della musica sviluppa una dinamica propria, i suoi postulati e le sue forme radicati in un'esperienza sociale della quale esso simbolizza istituzioni e conflitti».¹⁹³ Nel 1985, infatti, con *Répons*, monumentale esito del suo lavoro all'IRCAM, Boulez, non solo forniva risposte teorico-musicali per la strutturazione del dialogo tra solisti ed ensemble, tra solisti e parte elettronica, tra suono acustico e suono elaborato in tempo reale, ma intendeva dare anche – e forse soprattutto – una risposta *tranchant* ad altre teorie

¹⁹² Si veda Rapport IRCAM 1993, p. 45-46.

¹⁹³ HUGUES DUFOURT, *Musica, potere, scrittura*, Milano/Lucca, Ricordi/LIM, 1997, p. 10.

musicali in quegli anni concorrenti. Ancora nel 1989 in un'intervista a Peter McCallum Boulez affermerà a proposito di *Répons*:

Oh yes, there is a metre, slightly irregular on one level but very regular on another. There are so many irregular things in this piece that at one point you need to have a regular metre as you say – a bass and a regular pulse anyway – but also a series of harmonies which are all symmetrical. The harmony always gives this impression of something followed by its inverse; there is always a centre – an axis of symmetry. This symmetry of harmony corresponds in harmonic terms to a regular metre. This is very important.¹⁹⁴

Le basi concettuali erano poste: il legame tra tempo e frequenza, metro e spettro, così com'era concepito dalla visione processuale e intrinsecamente continua degli spettralisti (si veda in particolare *Tempus ex Machina* di Grisey) non era contemplabile. Riportata l'armonia sul piano discreto della combinatoria, anche il timbro, di conseguenza, doveva essere strutturato su basi gerarchiche, per essere inserito in modo coerente in un discorso musicale.¹⁹⁵

L'importanza di *EnTrance* risiede allora nella dimostrazione che Romitelli dà in chiave musicale di come il suo pensiero estetico rifiutasse alla base questa dicotomia, forse solo aprioristica. La via per far convivere i due universi concettuali gli viene fornita proprio dai sistemi CAO. LISP e *Patchwork* divengono così strumenti polifunzionali. Anche attraverso lo sviluppo di codice originale, Romitelli si crea un proprio 'scrittoio' informatico in grado unire analisi, manipolazione dei parametri e sintesi in un unico ambiente modulare, e di convogliare direttamente le soluzioni compositive sia nella notazione musicale, sia verso la redazione di script *Csound* per la generazione dei suoni.

Grazie alla possibilità aperte dalla moderna CAO di superare una concezione algoritmica della composizione, Romitelli realizza un sistema assieme concettuale e operativo che gli permette di costruire campi armonici radicati su un duplice controllo del materiale: il timbro è determinato da un lato da procedimenti spettrali, mutuati in particolare dall'armonia funzionale di Murail (basti pensare alle tecniche di distorsione spettrale), dall'altro da relazioni intervallari soggette a regole combinatorie, gli aggregati. A partire da parametri fissi di uno o dell'altro tipo, il sistema costruisce un reticolo di vincoli in grado di determinare il grado di compatibilità degli oggetti timbrici, la loro 'prossimità' misurata anche nei termini estetici di continuità o urto (rilettura contemporanea dei

¹⁹⁴ PETER MCCALLUM, "An Interview with Pierre Boulez", in «The Musical Times» v. 130 n. 1751, 1989, pp. 9–10.

¹⁹⁵ Si veda *infra*, par. 3.3.1.

concetti classici di tensione e risoluzione), e di illustrare così i percorsi possibili per la loro organizzazione. In *EnTrance* il momento spettrale precede quello di strutturazione gerarchica, e fornisce il materiale di base alla definizione degli aggregati. In questo modo, saldando i due momenti, le tecniche di sintesi strumentale entrano a pieno titolo nel discorso musicale; coerentemente integrate nel sistema, costituiscono solo un'ulteriore possibilità.

Questa concezione è esemplificata dall'analisi condotta sulla sezione **IB**: il materiale armonico deriva da nove spettri distorti, tra loro diversificati per frequenza fondamentale e grado di distorsione. Da questi spettri **S**, orchestrati in partitura per sintesi strumentale, Romitelli calcola con procedimenti combinatori gli aggregati **T** che costituiranno una loro anticipazione. La sezione si articola quindi in nove sottosezioni, caratterizzate dall'alternanza tra **T** e **S**. All'interno di ciascuna sottosezione le differenze tra i due elementi sono mitigate dalla presenza di frequenze in comune, affidate ai fiati, che garantiscono una transizione sfumata e graduale tra due oggetti timbrici concettualmente lontani.

Le tecniche compositive impiegate in *EnTrance* si ritroveranno poi, ulteriormente approfondite, nelle opere della piena maturità. Si prenda ad esempio il trittico *Professor Bad Trip* (1998-2000). Qui il discorso musicale continua ad essere organizzato rapportando la definizione del materiale armonico, legato ad un approccio sia spettrale sia intervallare, con la sua articolazione 'orizzontale' definita da un sistema gerarchico di relazioni. Elementi armonici, melodici e ritmici sono tra loro intrecciati, regolati da un insieme di aggregati predefiniti e preordinati. Inoltre, come in *EnTrance*, anche nella *Lesson I* Romitelli insiste sulla profonda coerenza armonica tra parte elettronica ed ensemble.¹⁹⁶

Un ulteriore elemento che non solo accomuna le due opere ma che può essere riscontrato in buona parte della produzione romitelliana è la costante volontà di trascendere i limiti della naturale percezione, oltre le soglie del quotidiano, alla ricerca di uno stato alterato della coscienza – basti pensare a titoli quali *Have your trip* (1988-1989), *Acid Dreams & Spanish Queens* (1994), *Trash TV Trance* (2002), solo per citare alcuni. Ritualità, ciclicità, ripetitività, violenza sonora, sono tratti distintivi della sua musica. Emblematico quanto scrive a proposito di *Professor Bad Trip*: «Le même matériau est travaillé trois fois dans une sorte de rituel énigmatique et violent».¹⁹⁷

¹⁹⁶ Si veda PIERRE MICHEL, "Professor Bad Trip (Lessons I, II, III)", in Alessandro Arbo (a cura di), *Le corps électrique. Voyage dans le son de Fausto Romitelli*, Parigi, L'Harmattan, 2005, pp. 51-77.

¹⁹⁷ FAUSTO ROMITELLI, "Professor Bad Trip : presentation", in Alessandro Arbo (a cura di), *Le corps électrique*, pp. 135-137.

In conclusione possiamo affermare che Romitelli opera all'IRCAM una sintesi mirabile tra spettralismo, calcolo e gioco combinatorio, per sviluppare un sistema compositivo del tutto personale, specchio della sua estetica, in cui vincoli e possibilità riaffermano a gran voce la libertà della scelta artistica. Perché «l'idea di vincolo trasforma le idee di ordine e necessità. I giochi dell'ordine e del disordine consentono di intendere il sorgere delle organizzazioni biologiche e umane nel senso di una storia di coproduzioni reciproche di vincoli e possibilità. [...] L'autonomia non si emancipa dai vincoli. Si costruisce al contrario all'interno della dipendenza ecologica e, per noi esseri umani, della dipendenza sociale e culturale».¹⁹⁸

¹⁹⁸ EDGAR MORIN, dalla presentazione del volume di MARIO CERUTTI, *Il vincolo e la possibilità*, Raffaello Cortina, Milano, 2009.

APPENDICE A: NUOVE FONTI MUSICALI

A partire dai primi anni del secondo Novecento nuovi mezzi di generazione del suono hanno permesso al musicista di passare con continuità dalle forme simboliche di notazione alla sintesi del suono, della pratica orchestrale all'interazione *live* con gli apparati di controllo in tempo reale del suono; la sua creatività si è espressa nell'intreccio di mezzi elettronici, acustici e informatici. Ai sistemi analogici di registrazione, sintesi ed elaborazione del suono si sono via via affiancati i nuovi media digitali: la versatilità del dominio numerico, accompagnata da una sempre maggiore potenza di calcolo, ha consentito lo sviluppo di una moltitudine di sistemi informatici che si è tradotto in una proliferazione di fonti eterogenee: un «bouillonnement organologique, où instruments acoustiques, technologies analogiques et technologies numériques issues de l'informatique forment un système de plus en plus intégré».¹⁹⁹ Il frenetico susseguirsi di nuove versioni dei software di sintesi, elaborazione e controllo in tempo reale dell'audio, raramente retro compatibili, è solo uno degli esempi della rapidità con cui i formati divengono obsoleti. Dal punto di vista della sopravvivenza di un'opera, questa condizione impedisce di fatto la sua ri-eseguibilità in un ambiente tecnologico differente da quello in cui è stata creata.

¹⁹⁹ BERNARD STIEGLER, “*Bouillonnements organologiques et enseignement musical*”, in «Les dossiers de l'ingénierie éducative» n. 43, Parigi, CNDP, 2003, pp. 11-15.

A.1 Stato dell'arte

Gli studi sulla ri-mediazione dell'audio analogico dichiarano la necessità di trasformare la visione dia-logica della filologia testuale in dia-sistemica,²⁰⁰ dove, nel quadro della teoria della comunicazione i sistemi tecnologici di produzione e registrazione dell'audio costituiscono di fatto l'elemento di codifica e decodifica dell'informazione. In uno scenario che potremmo definire shannoniano, in cui il sistema di decodifica è parte integrante ed essenziale del processo comunicativo, anche gli strumenti elettronici devono quindi essere preservati, perché determinano le condizioni di trasmissione e di esistenza dell'opera musicale.

È dagli anni Ottanta che la comunità scientifica è attiva nello sviluppo di strategie di preservazione del patrimonio audiovisivo degli ultimi 60 anni. Nel 1991 l'UNESCO promuoveva l'edizione di una guida alla preservazione dei documenti audiovisivi²⁰¹ con l'intento di fissare un codice di standard professionali – analogo alla Carta di Venezia – che costituisse un quadro di riferimento internazionale per disciplinare le modalità di interventi di preservazione, restauro e riedizione delle registrazioni analogiche audio e video. Nasceva così l'etica della preservazione del patrimonio documentale audiovisivo.

Oggi, come sostengono Cossettini e Orcalli:

trascorsi venticinque anni dalla stesura della *Guida* possiamo affermare che l'emergenza, sentita allora dai redattori del testo, è in parte superata. Oggi le procedure di trasferimento dell'audio dal dominio analogico a quello digitale possono ormai dirsi stabilmente definite. Degli obiettivi della migrazione dall'analogico al digitale, l'esito forse più rilevante è stato l'accesso facilitato ai documenti audio; conseguentemente è cresciuta anche la sensibilità degli storici e della critica musicale per le registrazioni. Inoltre, nel trentennio 1950-80, ricca e significativa è stata la produzione di opere audiovisive e musicali miste (scritte per strumenti acustici e parte elettronica). Tutto ciò ha portato ad attribuire ai documenti audiovisivi il ruolo di fonte primaria.²⁰²

²⁰⁰ Si veda ANGELO ORCALLI, "Traces sonores du XXe siècle. Pour une critique des sources audiovisuelles", in Evelyne Gayou (a cura di), *Musique et Technologique. Préserver - Archiver - Re-produire*, pp. 33-74.

²⁰¹ GEORGE BOSTON, *Guide to the basic technical equipment required by audio, film and television archives*. Coordinating Committee for the Technical Commissions of the International Organisations for Audio, Film and Television Archives., & International Federation of Film Archives, London 1991.

²⁰² LUCA COSSETTINI, ANGELO ORCALLI, *L'invenzione della fonologia musicale. Saggi sulla musica sperimentale di Luciano Berio e Bruno Maderna*, Lucca LIM, 2015, p. 99.

Il principio si estende al dominio dell'audio digitale e dell'informatica musicale, abbracciando la nuova organologia 'allargata' e l'intero sistema di produzione dell'opera.

Nel nuovo ambiente mediatico digitale, si ripropongono quindi in forme nuove e complesse antiche problematiche: a) l'autenticità dei documenti, b) il concetto di rumore, c) le forme di accesso all'audio.

In questo particolare ambito, le ricerche della comunità archivistica sono solo ai prodromi, e per ora concentrate sullo sviluppo di soluzioni in grado di rispondere ai problemi urgenti di conservazione attiva dei documenti digitali.²⁰³ Maggiori passi avanti sono stati fatti in ambito musicologico, sul fronte della ri-mediazione delle opere fissate su supporti informatici. La letteratura specializzata si distingue già diverse strategie:²⁰⁴

Migrazione

Con la migrazione si vuole interpretare il codice ormai obsoleto per poi riscriverlo in un linguaggio compatibile con il moderno ambiente hardware-software. Operazioni di questo tipo in campo musicale sono esemplificate dal saggio di Laurent Pottier sulla restituzione di *Turenas* di John Chowning:²⁰⁵ il codice creato originariamente per il programma di sintesi *Musicro* è stato riscritto in tre diversi ambienti: *MaxMSP*, *Csound* e *Faust*.

Virtualizzazione

La migrazione trova la sua giustificazione nell'urgenza di adattare le fonti ai nuovi sistemi, ma difficilmente garantisce una sopravvivenza dei documenti nel lungo periodo. Virtualizzare un documento, invece, mira a un livello di astrazione maggiore, al fine di renderlo indipendente da qualsiasi piattaforma. Il lavoro su *EnEcho* di Philippe Manoury,²⁰⁶ nel quadro del progetto ASTREE, si muove in questa direzione: le patch utilizzate per il controllo in tempo reale dell'audio sono state tradotte in linguaggio *Faust*, sorta di 'lingua franca' sviluppata dal GRAME di Lione, e in seguito in linguaggio matematico, più

²⁰³ Si veda Kevin Bradley (a cura di), *Guidelines on the Production and Preservation of Digital Audio Objects – IASA-TC 04*, Auckland Park, IASA, 2009.

²⁰⁴ Si veda BRUNO BACHIMONT, "Préservation culturelle numérique", in Evelyne Gayou (a cura di), *Musique et Technologique. Préserver – Archiver – Re-produire*, Parigi, INA, 2013, pp. 11-32.

²⁰⁵ LAURENT POTTIER, "La «régénération» des sons de *Turenas* de John Chowning", in Evelyne Gayou (a cura di), *Musique et Technologique. Préserver – Archiver – Re-produire*, pp. 145-196.

²⁰⁶ ALAIN BONARDI, "Pérenniser pour transmettre, transmettre pour pérenniser. Destins de l'œuvre mixte interactive – Autour de *En Echo*, pièce de Philippe Manoury", in Evelyne Gayou (a cura di), *Musique et Technologique. Préserver – Archiver – Re-produire*, pp. 105-125.

universale. L'approccio non è esente da problemi di ordine propriamente musicale, infatti è noto che lo stesso modello, applicato a sistemi informatici diversi, non restituirà risultati acustici tra loro equivalenti. Inoltre, la virtualizzazione altera inevitabilmente la specificità sintattica e semantica del linguaggio originale.²⁰⁷

Emulazione

L'emulazione muove dalla volontà di mantenere inalterato il codice del programma. L'obiettivo è di simulare un determinato ambiente hardware all'interno di un nuovo sistema, mantenendo così immutato il codice della fonte informatica. Mentre questa tecnica è particolarmente diffusa tra chi si occupa di conservazione di videogame,²⁰⁸ essa non è ancora stata ben indagata sul fronte musicale, dove solitamente si preferisce adottare la strategia della migrazione. I rari studi che aprono a possibilità di emulazione dell'hardware²⁰⁹ sono più che altro dettati dalla difficoltà di astrarre e quindi riscrivere il codice sorgente originale, spesso proprietario.

Campionamento

Il campionamento rappresenta un'altra possibile via per preservare il suono prodotto da un apparato di sintesi sonora minacciato da obsolescenza. Un esempio in questo senso si può trovare nella restituzione di *Diadèmes* di Marc-André Dalbavie,²¹⁰ opera che comprende nell'organico un sintetizzatore *Yamaha TX816*: una volta caricate le patch negli strumenti, i diversi suoni previsti in partitura sono stati digitalizzati e inseriti in un campionatore. L'approccio, a differenza dell'emulazione, non si incentra sulla preservazione dello

²⁰⁷ Si veda SERGE LEMOUTON, ALAIN BONARDI, RAFFAELE CIAVARELLA, "Peut-on envisager une organologie des instruments virtuels de l'informatique musicale ?", in *Proceeding of the Fifth Conference on Interdisciplinary Musicology (CIM'09)*, 2009. <<http://architexte.ircam.fr/textes/Lemoutono9b/index.pdf>>, ultima visita 5 marzo 2017.

²⁰⁸ Si veda NICOLAS ESPOSITO, "L'exemple de la préservation des jeux vidéo", in Evelyne Gayou (a cura di), *Musique et Technologique. Préserver - Archiver - Re-produire*, pp. 197-217..

²⁰⁹ Si veda YANN GESLIN, "Saturne, d'Hugues Dufourt ; ou la préservation d'une œuvre avec électronique", in Evelyne Gayou (a cura di), *Musique et Technologique. Préserver - Archiver - Re-produire*, pp. 127-144, e NICOLA BERNARDINI, ALVISE VIDOLIN, "Sustainable Live Electro-Acoustic Music", in «Proceedings of Sound and Music Computing 2005», Salerno, 2005, consultabile online all'indirizzo <http://econtact.ca/8_3/bernardini_vidolin.html>, ultima visita 5 marzo 2017.

²¹⁰ Si veda SERGE LEMOUTON, ALAIN BONARDI, RAFFAELE CIAVARELLA, "Peut-on envisager une organologie des instruments virtuels de l'informatique musicale ?".

strumento in sé quanto piuttosto sull'acquisizione di un suo determinato output, legato unicamente all'opera in esame.

Tuttavia la fattibilità dell'operazione deve essere valutata caso per caso: spesso infatti i compositori prescrivono agli strumentisti determinate azioni sui parametri dei sintetizzatori durante il concerto – oppure, al contrario, lasciano loro un certo grado di libertà. In questi casi, il campionamento andrebbe a cristallizzare la componente interpretativa dell'esecuzione, limitandone le possibilità espressive.²¹¹

Gli studi sulla preservazione dell'audio digitale e sulla migrazione/emulazione dei sistemi informatici rispondono in modo più o meno efficace ai problemi di obsolescenza tecnologica, ma questa fase del lavoro, seppur necessaria, non è sufficiente quando si ha a che fare con opere musicali: anche per i documenti digitali si pone il problema delle varianti. La stessa evoluzione tecnologica impone, in primo luogo al compositore, un continuo lavoro di riscrittura delle proprie opere per assecondare i mutamenti della nuova organologia. Gérard Grisey, in un'intervista a David Bündler, espone molto chiaramente questa difficoltà:

toutes les pièces que j'ai composées qui ont nécessité de l'électronique ont dû être révisées constamment du fait de l'évolution de la technologie. [...] Si vous écrivez une pièce pour électronique, vous êtes constamment obligé de renouveler le système afin qu'elle soit toujours disponible pour la salle de concert. [...] La technologie m'oblige à revenir en arrière et à travailler dessus a nouveau. Un nouveau type de bande. Passer d'une bande à l'ordinateur. Et puis d'un ordinateur à un nouveau type d'ordinateur. Ou d'un synthétiseur à un nouveau modèle de synthétiseur. Ça n'a pas fin.²¹²

Questo lavoro si traduce in una proliferazione di versioni eterogenee, di volta in volta adattate per assecondare le esigenze della tecnologia in uso al momento. L'importanza della documentazione del sistema di produzione di ogni singola versione diviene allora evidente, per scopi archivistici, riesecutivi ma anche analitici/musicologici. Se infine, come sottolineano Guillaume Boutard e Fabrice Marandola, «documentation, dissemination and preservation are intimately interconnected»,²¹³ anche la diffusione della conoscenza è parte

²¹¹ Si veda YANN GESLIN, “*Saturne*, d'Hugues Dufourt ; ou la préservation d'une œuvre avec électronique”.

²¹² GERARD GRISEY, “Entretien avec David Bündler”, in *Écrits, ou l'invention de la musique spectrale*, pp. 269-270. L'intervista è datata 1996.

²¹³ GUILLAUME BOUTARD, FABRICE MARANDOLA, “Mixed music creative process documentation methodology: outcomes of the DiP-CoRE project”, in *Proceeding of the 9th Conference on Interdisciplinary Musicology (CIM'14)*, 2014. Disponibile online all'indirizzo <https://www.academia.edu/20970403/Mixed_

integrante del processo di salvaguardia del patrimonio musicale. Già Bernardini e Vidolin, ormai un decennio fa, auspicavano una collaborazione tra i diversi soggetti in campo, una comunità attiva nello scambio di documenti ed esperienze.²¹⁴ Questo auspicio, ad oggi, ha trovato riscontro nel mondo archivistico (PrestoCentre.org) ma ancora non emerge una community internazionale per la condivisione delle pratiche artistiche musicali.

A partire dal 2002, il progetto MUSTICA ha avuto come obiettivo la restituzione delle opere di musica elettroacustica: alle pratiche di preservazione attiva delle fonti ha affiancato la documentazione delle ‘versioni’ tramite la raccolta delle informazioni – spesso orali – derivate dalle prassi del lavoro in studio. La centralità del concetto di ‘versione’ è dichiarata nella ridefinizione del modello FRBR: i quattro stadi *Work, expression, manifestation e item* sono coniugati in *Work, version, execution, item*.²¹⁵

Con l’affermazione del modello OAIS,²¹⁶ del progetto europeo CASPAR e dei progetti incentrati sull’autenticità dei documenti numerici,²¹⁷ si è cercata una convergenza tra i sistemi di documentazione e archiviazione in uso e i nuovi standard. È il caso ad esempio della riscrittura di MUSTICA: il sistema, originariamente fondato sul modello FRBR, è stato poi ridefinito in MustiCaspar, una piattaforma sviluppata per garantire la compatibilità con diversi standard: OAIS, XFDU, CIDOC-CRM e RDF.²¹⁸

Nicolas Esposito e Yann Geslin hanno sviluppato una strategia di preservazione e documentazione della musica acusmatica, basata sull’ontologia CIDOC-CRM e inserita nel quadro del progetto CASPAR.²¹⁹ Altri progetti, in particolare GAMELAN²²⁰ e DiP-

music_creative_process_documentation_methodology_outcomes_of_the_DiP-CoRE_project>, ultima visita 5 marzo 2017.

²¹⁴ NICOLA BERNARDINI, ALVISE VIDOLIN, “Sustainable Live Electro-Acoustic Music”.

²¹⁵ Xavier Sirven, *Authenticité et accessibilité des archives électroniques. MUSTICA. Le cas de la création musicale numérique*, IRCAM internship report, 2004. <<http://polaris.gseis.ucla.edu/blanchette/papers/RapportSirven.pdf>>, ultima visita 5 marzo 2017.

²¹⁶ CCSDS: The Consultative Committee for Space Data Systems, *Reference model for an open archival information system (OAIS). Recommended practice*, CCSDS 650.0-M-2, Magenta book, 2012.

²¹⁷ Si veda MARIA GUERCIO, “La notion d’autenticité en conservation numérique”, in Evelyne Gayou (a cura di), *Musique et Technologique. Préserver - Archiver - Re-produire*, pp. 75-92.

²¹⁸ Si veda SERGE LEMOUTON, ALAIN BONARDI, RAFFAELE CIAVARELLA, “Peut-on envisager une organologie des instruments virtuels de l’informatique musicale ?”.

²¹⁹ NICOLAS ESPOSITO, YANN GESLIN, “Long-term preservation of acousmatic works: Toward a generic model of description”, *Electrotechnical Conference*, 2008, MELECON 2008. The 14th IEEE Mediterranean, pp. 270-274.

²²⁰ Si veda ANTOIN VINCENT, *Préservation d’œuvres musicales. Étude du processus de production*, Tesi presso l’Université de Technologie de Compiègne, sotto la direzione di Bruno Bachimont e Alain Bonardi, 2010.

CoRE²²¹ focalizzano l'attenzione sulla registrazione e documentazione in presa diretta del sistema di produzione e delle operazioni svolte durante la creazione di un'opera.

Nella stessa direzione muove anche il recente progetto Sidney-IRCAM.²²² I suoi obiettivi sono chiaramente dichiarati:

- Storicità: tener traccia della storia delle esecuzioni di una determinata opera (in particolare le diverse produzioni IRCAM);
- Preservazione: garantire la ri-eseguibilità delle opere, pianificando almeno una preservazione a medio termine delle opere;
- Documentazione: per ciascuna versione, fornire una documentazione esaustiva sull'allestimento del sistema;
- Disseminazione: validare le singole versioni in modo da renderle comprensibili – e quindi eseguibili – anche fuori dall'IRCAM.

Sydney costituisce un notevole passo in avanti rispetto a MUSTICA, perché riducendo di fatto il numero di campi da compilare semplifica il sistema dal punto di vista della gestione informatica e archivistica. L'idea è di concedere una maggiore libertà descrittiva, per rispondere all'eterogeneità delle fonti e delle informazioni da inserire. Tuttavia, è indispensabile che il documentalista chiamato a immettere i dati conosca a fondo le pratiche e le tecniche di produzione e diffusione della musica elettroacustica, nonché le diverse versioni delle opere. In altri termini, il soggetto chiamato in causa generalmente risponde alla figura del *réalisateur en informatique musicale* che ne ha curato la versione, più che a quella di un archivista.

²²¹ Si veda GUILLAUME BOUTARD, FABRICE MARANDOLA, "Mixed music creative process documentation methodology: outcomes of the DiP-CoRE project".

²²² <<http://brahms.ircam.fr/sidney/>>, ultima visita 5 marzo 2017.

A.2 Le fonti di *EnTrance*

A.2.1 Recensione

A.2.1.1 Le partiture

P1 Parigi, Ricordi R.2744, *EnTrance*, 1996, partitura edita.

La partitura rappresenta lo stadio finale della lavorazione di *EnTrance*.

P2 Parigi, IRCAM [senza collocazione], *EnTrance*, bozza di partitura.

Testimonia una fase della stesura della partitura in cui è assente l'annotazione della parte elettronica e le parti di tastiera non sono complete. Annotazioni a matita sul frontespizio riportano: «18/01/96 AJ». AJ con tutta probabilità sono le iniziali di Alain Jacquinot, all'epoca direttore della produzione IRCAM.

P3 Parigi, IRCAM [senza collocazione], *EnTrance*, bozza di partitura.

La partitura testimonia uno stadio intermedio tra **P2** e **P1**. Le parti di tastiera sono redatte, ma la parte elettronica è ancora assente. Sono presenti indicazioni che suggeriscono l'impiego di questa partitura per la prima esecuzione di *EnTrance*: indicazioni sull'innesco delle diverse sezioni della parte elettronica, così come indicazioni sulla taratura del modulo per il riverbero artificiale *Lexicon LXP1* impiegato in concerto.

P4 Saint-Étienne, archivio privato di Laurent Pottier [senza collocazione], *EnTrance*, bozza di partitura.

Così come per **P3**, le parti di tastiera sono complete ma la notazione dell'elettronica è assente. Sul sistema destinato alla scrittura della parte elettronica sono presenti numerose indicazioni che fanno presumere un suo impiego per la creazione del progetto multitraccia utilizzato per diffondere l'elettronica in concerto: appunti precisi sull'inizio e la fine delle sezioni elettroniche, così come l'esplicitazione della scansione metrica per la generazione della *click-track*.

A.2.1.2 Supporti magnetici

ADAT1 Parigi, IRCAM [senza collocazione], [1996?]. nastro magnetico su Cassetta S-VHS *AMPEX 489* registrato in formato audio digitale ADAT (48 kHz, 16 bit).

Il nastro contiene la parte elettronica di *EnTrance*. Delle otto tracce disponibili, le prime quattro tramandano la parte elettronica quadrifonica, la quinta contiene la click-track per il direttore. Le ultime tre non sono state utilizzate.

F1 Parigi, IRCAM [senza collocazione], [1996?]. Floppy disk 3,5" *MEMOREX 2s/HD* Microdisk Double Sided 135 TPI, formattato a densità normale (720 Kb). File contenuto: 'EnTrance.T01'.

Il disco contiene la patch per il sintetizzatore *Yamaha SY99*.

A.2.1.3 Supporti ottici

CD1 Parigi, IRCAM [senza collocazione], 1996. Cd *Philips* CD-R 74 (6x), file system: Mac OS standard [HFS], creato il 15 febbraio 1996, 16:45; modificato il 1 marzo 1996, 14:52; Dimensione: 644.466.688 bytes. Nome disco: 'Romitelli'.

Il CD-ROM contiene il materiale da concerto impiegato nella prima esecuzione di *EnTrance*: parte elettronica, elementi per la configurazione e il controllo del campionatore *SampleCell* e per il sintetizzatore *Yamaha SY99*. A margine, sono preservati materiali impiegati nella composizione della parte elettronica.

CD2 Parigi, IRCAM [senza collocazione], 2005. Cd *Verbatim DataLifePlus Super Azo Crystal* 700MB 52X DL+, file system: Mac OS extended [HFS+], creato il 16 maggio 2005, 16:31, modificato il 17 maggio 2005, 10:0, dimensione: 493.125.632 bytes. Nome disco: 'EnTrance_2005'.

Il CD-ROM tramanda una versione postuma del dispositivo elettronico di *EnTrance*, realizzata da Denis Lorrain nel 2005, in cui il campionatore *SampleCell* è stato sostituito con il modello *Akai S5000*. Contiene: istruzioni e suoni per il campionatore, patch per il sintetizzatore *Yamaha SY99*, parte elettronica quadrifonica controllata da una patch in *Max/MSP* creata ad hoc.

CD3 Parigi, IRCAM [senza collocazione], 2005. Cd *Verbatim DataLifePlus Super Azo Crystal* 700MB 52X DL+, file system: Mac OS extended [HFS+], creato il 22 marzo 2005, 13:41, dimensione: 10.006.528 bytes. Nome disco: 'EnTrance_Akai_S5000'.

Il CD-ROM tramanda la versione postuma curata nel 2005 da Denis Lorrain. Contiene le istruzioni e i file da caricare nel campionatore *AKAI S5000*.

CD4 Parigi, IRCAM [senza collocazione], 2005. Cd *Verbatim DataLifePlus Super Azo Crystal 700MB 52X DL+*, file system: Mac OS extended [HFS+], creato il 1 marzo 2005, 13:41, dimensione: 451.940.352 bytes. Nome disco: 'Romitelli_EnTrance'.

Il disco contiene il riversamento delle tracce audio di **ADAT1**. Le quattro tracce più la *click track* sono incapsulate in un unico file 'EnTrance.aif'. È inoltre conservata la patch di *Max/MSP* con cui si è effettuato il riversamento.

A.2.2 Accesso alle fonti originali

A.2.2.1 Il nastro ADAT (ADAT1)

La cassetta S-VHS contiene la parte elettronica quadrifonica pensata per l'esecuzione in concerto, qualora si incontrassero difficoltà nell'impiego dei file audio originali.²²³ Il flusso audio digitale è codificato *ADAT (Alesis Digital Audio Tape)*, un formato proprietario e ormai obsoleto della *Alesis*. Riprodurre il nastro oggi non è agevole, in quanto è sempre più raro individuare apparati perfettamente funzionanti e allineati. Un mancato allineamento della testina può produrre infatti errori di lettura. I sistemi *ADAT* integrano un algoritmo di controllo e correzione dell'errore fondato principalmente sul codice di Hamming. Pertanto, quando la corruzione supera una certa soglia il sistema non riesce a ricostruire il segnale digitale originale e sostituisce la porzione corrotta con dati stimati per interpolazione.

La presenza presso gli studi IRCAM di un lettore *Alesis XT* perfettamente allineato ha permesso di accedere all'informazione contenuta nel nastro. Il segnale originale è codificato alla frequenza di campionamento di 48 kHz e alla risoluzione di 16 bit. Sono state utilizzate le prime cinque tracce (tracce da 1 a 4, parte elettronica di *EnTrance*; traccia 5, *click track* per il direttore).

²²³ «Sono possibili due soluzioni: 1) A-Dat: contiene una versione continua di tutte le sequenze audio ed il metronomo per il direttore su tutta la durata del brano. L'A-Dat deve scorrere, come un nastro, dall'inizio alla fine del brano senza interruzioni; 2) Direct-to-Disk: le sequenze audio sono contenute su hard-disk e lanciate di volta in volta via MIDI da una tastiera aggiuntiva suonata dal secondo tastierista. Questa soluzione è preferibile». FAUSTO ROMITELLI, *EnTrance*, Parigi, Ricordi, 1996, p. iv.

Per la rimediazione dell'audio registrato nel nastro si è configurato il seguente sistema:²²⁴ *Alesis ADAT XT* → interfaccia ottica *ADAT* → scheda *RME Fireface UC* → interfaccia USB → *Mac Pro* con *Pro Tools*.

Il sistema ha permesso di acquisire il flusso audio senza passare per il dominio analogico, operando di fatto una conversione DD, e infine di memorizzarlo in formato wave. Per garantire l'assenza di errori prodotti dall'interfaccia di acquisizione, il nastro è stato acquisito due volte. La somma in controfase tra i due riversamenti ha garantito l'assenza di possibili corrottele inserite in fase di acquisizione dei dati.

A.2.2.2 Il floppy disk (F1)

Il floppy disk tramanda la patch per i suoni del sintetizzatore *Yamaha SY99*. Le patch conservate su questo supporto difficilmente possono essere caricate direttamente nel dispositivo; ciò è dovuto a un diffuso e ben noto problema legato alla scarsa qualità dei lettori floppy integrati nel sintetizzatore. Ciononostante, è stato possibile accedere all'informazione contenuta nel documento: allestita una workstation *PowerMac G4*, il file contenuto nel floppy – formattato a normale densità, 720 KB – è stato salvato su disco rigido. Se l'operazione consente di 'salvare' l'informazione dal degrado fisico-chimico a cui il floppy va incontro, d'altro canto non risolve i problemi legati alla comunicazione con il sintetizzatore. Il codice contenuto nel file infatti non è direttamente interpretabile.

A.2.2.3 Il CD-ROM (CD1)

Creato nel febbraio del 1996, il CD tramanda il materiale da concerto utilizzato per la prima esecuzione di *EnTrance*, insieme al materiale preparatorio impiegato nella composizione della parte elettronica. Come si è visto nel quarto capitolo, per la sua lettura si è allestita una postazione *PowerMac G4* con *Mac Os 9.2* che ha garantito una buona retro-compatibilità nell'accesso ai dati.

A.2.3 *Interpretazione dei dati: ambienti software*

La presenza di materiale informatico eterogeneo ha imposto di individuare i software necessari all'interpretazione delle informazioni contenute nei file. In questo senso, la ricerca

²²⁴ L'operazione di riversamento si è svolta a Settembre 2015 in collaborazione con Jeremie Bourgogne.

presso gli archivi del dipartimento di produzione dell'IRCAM ha costituito un canale privilegiato in quanto, fortunatamente, vi sono ancora conservati buona parte dei software storici. Si è quindi provveduto ad installare e configurare i seguenti software:

- *Motu Digital Performer 2.7*, con cui si è potuto accedere al progetto multitraccia impiegato per la diffusione della parte elettronica;
- *Soft SampleCell Editor*, che decodifica le istruzioni per il campionatore;
- *Max 3.5.8* ha consentito di aprire la patch originariamente adibita alla selezione via MIDI del suono da utilizzare per il campionatore. Il file può essere agevolmente aperto anche con le moderne versioni di *Max/MSP*;
- *Opcodes Galaxy 2.5.4*, che ha garantito l'accesso alle patch per il sintetizzatore Yamaha SY99. È probabile che i suoni siano stati programmati direttamente con versioni precedenti del software;
- *Csound 1.0*, in grado di rigenerare gran parte dei suoni di sintesi poi confluiti nella parte elettronica. In ogni caso, le più recenti versioni del software sono retro compatibili e quindi possono essere impiegate per la rigenerazione dei suoni;
- *Patchwork 2.7*, con le librerie *Csound/Edit-sco* e *SpData*, ha consentito di far 'rivivere' l'ambiente grafico dedicato alla composizione assistita all'elaboratore con cui si sono confrontati Romitelli e Pottier. In questo modo si è potuto consultare buona parte dei file di *Patchwork*. Fanno eccezione alcune patch, che richiedono una libreria esterna non compatibile con il sistema allestito. La difficoltà deriva da un problema di incompatibilità tra diverse architetture informatiche: la libreria *PW-chant* è stata creata per le versioni di *Patchwork* sviluppate per i sistemi *Apple* basati su processori *Motorola* della serie *68k*. Con la diffusione dei processori *PowerPc* (1994), incompatibili con la tecnologia *68k*, *Patchwork* fu interamente riscritto, e con lui le diverse librerie. Ma non *PW-chant*, che rimase ancorata all'architettura precedente. Le ricerche condotte presso l'IRCAM hanno consentito di risalire a una versione di *Patchwork* scritta per processori *68k*, così come alla libreria *PW-chant*. Ciononostante, non è stato possibile individuare una workstation compatibile e sufficientemente performante, in grado di eseguire correttamente il software;
- il programma *WORKLISP*, scritto da Romitelli per la composizione di *Natura morta con fiamme*, non ha posto ostacoli: è infatti scritto in puro linguaggio Common Lisp, di cui esistono numerose implementazioni per ogni sistema operativo;
- anche l'accesso ai singoli file audio non ha comportato particolari problemi. Alcuni di essi sono registrati nel moderno formato 'aiff', mentre la maggior parte sono in formato

‘snd’, all’epoca predefinito in *Csound*, o ancora in ‘sd2’(sound designer II). Nonostante tali formati siano ormai in disuso da anni, i file sono tuttora facilmente leggibili da software quali *iZotope RX 4* o *Audacity*.

Già a distanza di soli due decenni dalla creazione dell’opera sono emerse dunque difficoltà nell’accesso e nell’interpretazione dei dati digitali: se le moderne versioni di software quali *Csound* o *Max-MSP* sono retro compatibili, in altri casi si è imposto l’utilizzo di una tecnologia prossima all’ambiente tecnologico dell’epoca; un approccio che potremmo definire archeologia informatica, in cui l’impossibilità di reperire tecnologie compatibili impedisce di fatto lo studio delle fonti (come nel caso delle patch originali del sintetizzatore *Yamaha SY99* o della libreria *PW-Chant*).

A.3 Nuova organologia

A.3.1 Il sintetizzatore Yamaha SY99

L’*SY99* fu prodotto da *Yamaha* a partire dal 1991. Si tratta di un sintetizzatore digitale che combina due tecniche di sintesi del suono: AFM (advanced frequency modulation) e AWM (advanced wave memory). La sintesi AFM è il risultato di un progressivo affinamento e potenziamento della sintesi per modulazione di frequenza scoperta da Chowning (si veda *infra*, p. 32): nel sintetizzatore, gli operatori coinvolti nella sintesi sono sei, ognuno dei quali può assumere una tra 16 forme d’onda differenti (sinusoidale, triangolare, a dente di sega ecc.). Inoltre, i parametri in ingresso degli operatori possono essere controllati a feedback da altri operatori. La complessità dell’algoritmo sviluppato da *Yamaha* era tale da garantire un potenziale espressivo all’epoca ineguagliato. La sintesi per AFM è invece basata su suoni precedentemente campionati a 48 kHz e 16 bit. L’elemento base è costituito da quattro semplici blocchi principali che vanno a controllare il suono: il blocco *Waveform*, che legge la forma d’onda campionata, il blocco *Filter* che lo modifica attraverso il filtraggio, il blocco *Amplifier* che gestisce l’ampiezza e il blocco *Pan* che va a definire il panorama stereofonico. A questi blocchi, si affiancano elementi di controllo quali ad esempio un generatore di involuppo (EG), la *key velocity* (parametro legato alla velocità di abbassamento dei tasti), così come un oscillatore a bassa frequenza LFO che può essere applicato ai diversi blocchi per ottenere effetti diversi: sul blocco *Waveform* produce un vibrato, sull’*Amplifier* un tremolo, sul *Filter* un effetto di *wah-wah*.

Il sintetizzatore è in grado di gestire contemporaneamente fino a quattro diversi elementi, combinando AFM e AWM secondo necessità (figura 77):

Mode	Element	E1	E2	E3	E4
01	1AFM mono	AFM	—	—	—
02	2AFM mono	AFM	AFM	—	—
03	4AFM mono	AFM	AFM	AFM	AFM
04	1AFM poly	AFM	—	—	—
05	2AFM poly	AFM	AFM	—	—
06	1AWM poly	AWM	—	—	—
07	2AWM poly	AWM	AWM	—	—
08	4AWM poly	AWM	AWM	AWM	AWM
09	1AFM & 1AWM poly	AFM	AWM	—	—
10	2AFM & 2AWM poly	AFM	AFM	AWM	AWM
11	Drum Set	76 AWM waveforms			

Figura 77 – Possibili modi di impiego dei diversi elementi di sintesi. Immagine tratta dal manuale d'uso dell'SY99.

A.3.1.1 Obsolescenza tecnologica e strategie di preservazione

Così come ogni apparato elettronico, l'obsolescenza che minaccia il sintetizzatore è legata principalmente a due fattori: da un lato, l'industria immette freneticamente nel mercato strumenti sempre più performanti che soppiantano rapidamente i loro predecessori; dall'altro, i componenti elettronici tendono col tempo a deperire. Inoltre, l'eventuale processo di riparazione è reso sempre più complesso dalla progressiva miniaturizzazione e irreperibilità dei componenti stessi. È pertanto indispensabile pianificare strategie di preservazione che possano garantire la sopravvivenza a lungo termine di strumenti musicali che negli ultimi decenni sono entrati stabilmente nelle pratiche orchestrali.

Pur comprendendo il funzionamento del sintetizzatore, vi è una profonda limitazione che esclude ogni possibile strategia di migrazione o di emulazione finalizzata alla sua replicazione fedele in un sistema tecnologico differente. L'inaccessibilità del codice proprietario impedisce di comprendere perfettamente le relazioni che intercorrono tra i parametri impostati in fase di progettazione del suono e il loro effettivo ruolo nella sintesi. Nell'evidente impossibilità di preservare a lungo termine lo strumento nel suo insieme, forse l'unica strada percorribile al fine di garantire la sopravvivenza del suono è quella del campionamento. Il caso di *EnTrance* è compatibile con questo approccio: la partitura non prevede interventi sui parametri di sintesi del suono durante la performance, ma solo una variazione del *pitch* e il controllo in volume.

A.3.1.2 I suoni di *EnTrance*

Nel caso in cui non sia possibile reperibile un sintetizzatore SY99 con il lettore di floppy disk funzionante, i file *Galaxy* conservati in CDi costituiscono l'unica via per caricare le patch nello strumento. Il procedimento è semplice, a patto di allestire una workstation in grado di eseguire il software *Galaxy*. Nel nostro caso il sintetizzatore è stato collegato a un computer *PowerMac G4* con sistema operativo *Mac OS 9.2* tramite un'interfaccia MIDI-USB,²²⁵ e si è quindi effettuato il *dump* delle patch.

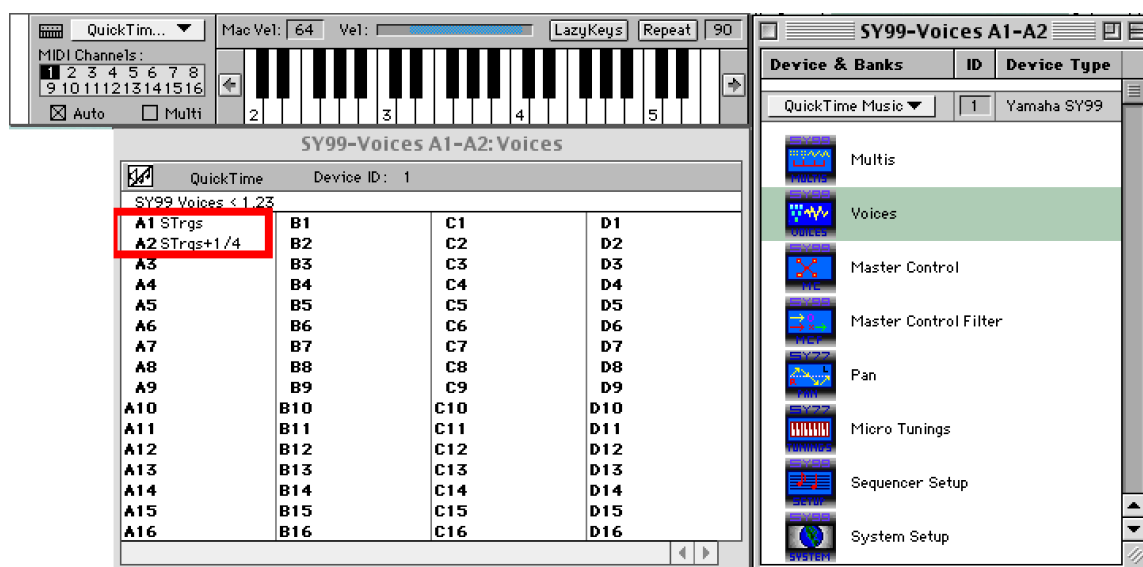


Figura 78 – Ambiente *Galaxy*. In rosso sono evidenziate le due voci che vanno caricate nel sintetizzatore SY99.

Come si evince dalla figura 78, i suoni previsti in *EnTrance* sono due, tra loro identici fatta salva una trasposizione all'acuto di $\frac{1}{4}$ di tono per il secondo suono. Romitelli li denomina 'archi filtrati': si tratta di suoni dal contenuto spettrale rarefatto, con tempi di attacco e di rilascio molto lunghi. Dall'interfaccia integrata del sintetizzatore si possono ricavare tutti i parametri impiegati per la sintesi. Ad esempio, emerge che i suoni sono ottenuti per sintesi AWM a due elementi (*mode 7* in figura 77), e che le forme d'onda a cui fanno riferimento sono due *preset*: '48-AnlgBass' e '46-Pad r'.

Di seguito, i parametri rilevati:

²²⁵ Sebbene questa soluzione possa apparire agevole, va considerato che l'interfaccia MIDI-USB deve essere compatibile con il sistema operativo *Mac Os 9*, ultima versione che supporta il software *Galaxy*. Da ormai oltre un decennio, però, le case produttrici non sviluppano più *driver* per il vecchio sistema, di fatto imponendo l'utilizzo di apparati in disuso già da anni.

- MODE — 07: 2AWM POLY
- COM
 - ELEMLVL
 - TOTAL VOICE VOLUME = 123;
 - E1 = 127
 - E2 = 127
 - ELEMDTN — DETUNE:
 - E1 = +4
 - E2 = -3
 - NTSHIFT — NOTE SHIFT:
 - E1 = +12
 - E2 = +12
 - NTLIMIT — ELEMENT NOTE LIMIT:
 - E1 = LOW C-2, HIGH G-8
 - E2 = LOW C-2, HIGH G-8
 - VLLIMIT — ELEMENT VELOCITY LIMIT:
 - E1 = LOW1, HIGH 127
 - E2 = LOW 1, HIGH 127
 - ELEMPAN — ELEMENT DYNAMIC PAN:
 - E1 = P-9 = LEFT 5
 - E2 = P-3 = RIGHT 5
 - OUTSEL — OUTPUT GROUP SELECT:
 - E1 = BOTH;
 - E2 = BOTH
 - RANDOM — RANDOM PITCH DEPTH: 0
 - PORTA — PORTAMENTO:
 - MODE = FOLLOW
 - TIME = 0
 - EFFECT — 01 EFFECT MODE SELECT = 0
 - MCRTUNE — MICRO TUNING SET
 - STRGS = P-1 = EQUAL, E1 = OFF, E2 = OFF;
 - STRGS + $\frac{1}{4}$ = P-53 = $\frac{1}{4}$ SHIFT, E1 = ON, E2 = ON
 - CNTRLR
 - PITCH BEND RANGE = 2
 - AFTERTOUC MODE = ALL, SPLIT POINT = C3, PITCH BEND = 0
 - MODULATION DEPTH
 - PITCH DEPTH 64, MIDI 01, CONTROLLER: MODULATION WHEEL;
 - AMPLITUDE DEPTH: 64, MIDI: 012, CONTROLLER: NON ASSIGNED,
 - FILTER: 0
 - PAN CONTROL
 - PAN LFO DEPTH = 0, MIDI = 0, CONTROLLER OFF;
 - PAN BIAS DEPTH: 127, MIDI: 013, CONTROLLER NON ASSIGNED;
 - OTHER
 - VOL LOW LIMIT: VALUE: 0, MIDI: 014, CONTROLLER NON ASSIGNED;
 - EG BIAS DEPTH: VALUE: 0, MIDI: 0;
 - CUTOFF DEPTH: VALUE: 76, MIDI: 013, CONTROLLER: NOT ASSIGNED;
- ELEMENTS

	ELEM.1	ELEM.2
○ WAVESET		
▪ WAVEFORM	48 ANLGBASS	46 PAD 1
▪ FREQ MODE	NORMAL	NORMAL
▪ FREQ FINE	+0	+0

○ EG			
▪ MODE	ATTACK	ATTACK	
▪ R1	12	14	
▪ R2	9	16	
▪ R3	0	0	
▪ R4	31	0	
▪ RR	19	27	
▪ L2	63	63	
▪ L3	62	63	
▪ RS	+2	+0	
○ OUTPUT			
▪ BP1			
• NOTE	C-2	C1	
• OFFSET	+0	+0	
▪ BP2			
• NOTE	G0	G2	
• OFFSET	+0	+0	
▪ BP3			
• NOTE	C6	C6	
• OFFSET	+0	+0	
▪ BP4			
• NOTE	G7	G7	
• OFFSET	-34	-34	
○ SENSITIVITY			
▪ VELOCITY SENS	+2	+2	
▪ RATE VEL SWITCH	OFF	OFF	
▪ AMP MODE SENS	+0	+0	
▪ PITCH MODE SENS	0	3	
○ LFO			
▪ WAVE	SINE	SINE	
▪ SPEED	0	0	
▪ DELAY	0	0	
▪ A MOD DEPTH	0	0	
▪ M MOD DEPTH	0	0	
▪ F MOD DEPTH	0	0	
▪ INIT PHASE	0	0	
○ PITCH EG			
▪ DATA			
• RATE SCALING	+0	+0	
• VELOCITY SW	OFF	OFF	
• PEG RANGE	2 OCT	2 OCT	
▪ EG:			
• R1	62	63	
• R2	63	63	
• R3	63	63	
• RR	63	63	
• L0	0	0	
• L1	0	0	
• L2	0	0	
• L3	0	0	
• RL	0	0	
○ FILTER			
▪ CUTOFF FREQ			
• F1			
○ TYPE	LPF	LPF	
○ CUTOFF F.	251,2 HZ	1052 HZ	

	○ CTRL	EG	LFO
• F2			
	○ TYPE	LPF	LPF
	○ CUTOFF F.	774,3 HZ	774,3 HZ
	○ CTRL	LFO	LFO
• RESONANCE		82	60
• VELOCITY SENS		+0	+0
• LFO CUTOFF SENS		+0	+5
▪ CUTOFF SCALING FLT1			
• LPF COF		251,2 HZ	1052 HZ
• BP1			
	○ NOTE	A#2	C1
	○ OFFSET	-24	+0
• BP2			
	○ NOTE	C5	G2
	○ OFFSET	+54	+0
• BP3			
	○ NOTE	F#8	E4
	○ OFFSET	+0	+0
• BP4			
	○ NOTE	G8	C6
	○ OFFSET	+0	+0
▪ CUTOFF SCALING FLT2			
• LPF COF		774,3 HZ	774,3 HZ
• BP1			
	○ NOTE	E0	C1
	○ OFFSET	+80	+0
• BP2			
	○ NOTE	B2	G2
	○ OFFSET	+0	+0
• BP3			
	○ NOTE	F#8	E4
	○ OFFSET	+0	+0
• BP4			
	○ NOTE	G8	C6
	○ OFFSET	+0	+0
▪ CUTOFF EG			
• FLT 1			
	○ RATE		
	▪ R1	63	0
	▪ R2	0	0
	▪ R3	0	0
	▪ R4	0	0
	▪ RR1	0	0
	▪ RR2	0	0
	▪ RS	+1	0
	○ LVL		
	▪ L0	-64	0
	▪ L1	0	0
	▪ L2	0	0
	▪ L3	0	0
	▪ L4	0	0
	▪ RL1	0	0
	▪ RL2	0	0
• FLT 2			
	○ RATE		
	▪ R1	63	0

	▪ R2	0	0
	▪ R3	0	0
	▪ R4	0	0
	▪ RR1	0	0
	▪ RR2	0	0
	▪ RS	0	0
○ LVL			
	▪ L0	0	0
	▪ L1	0	0
	▪ L2	0	0
	▪ L3	0	0
	▪ L4	0	0
	▪ RL1	0	0
	▪ RL2	0	0

L'estrazione dei parametri in forma testuale, pur non rispondendo all'urgenza di una preservazione a lungo termine dell'apparato, consente di programmare i suoni direttamente dalla tastiera e così elude la necessità di configurare un sistema di comunicazione MIDI per il *dump* della patch.

A.3.2 Il campionatore SampleCell

In *EnTrance* Romitelli utilizza il campionatore *SampleCell* nelle sue funzioni di base: acquisire uno o più suoni reali e riprodurli trasposti a diverse altezze, grazie al controllo di una master keyboard. Al campionatore sono affidati cinque diversi suoni:

- Suono di archi filtrati;
- lo stesso suono, trasposto di $\frac{1}{4}$ di tono all'acuto;
- chitarra elettrica distorta, nota singola (Metal);
- chitarra elettrica distorta, triade maggiore (Metal Triad);
- pianoforte.

In CD1 sono conservati i campioni audio, in formato aiff, che dovranno esser gestiti dal campionatore. La lettura delle istruzioni in *SampleCell Editor* fa chiarezza sui parametri di controllo. All'apertura del *bank* generale, il software visualizza cinque programmi, uno per ogni suono (figura 79):

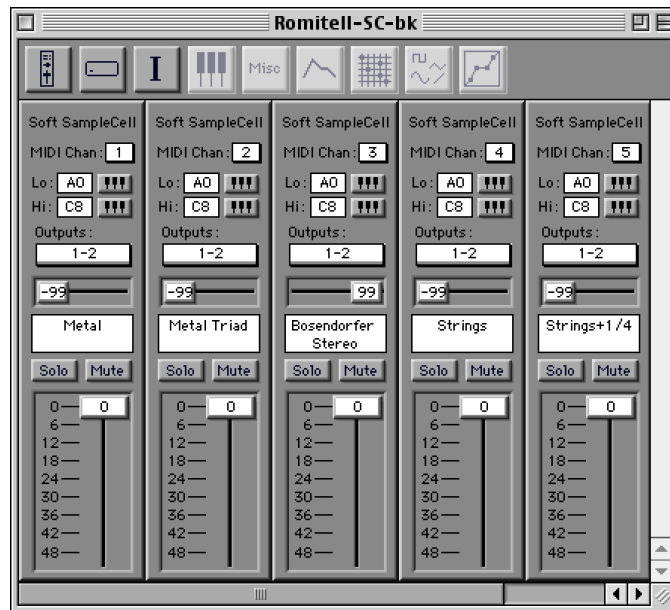


Figura 79 – Bank generale per il campionatore in EnTrance

Prendendo in esame il set ‘Metal Triad’, si può notare come la gamma delle altezze sia definita da tre diversi suoni campioni: ‘Metal Triad Power E1 LP’, ‘Metal Triad Power A1 LP’ e ‘Metal Triad Power C2 LP’ (figura 80). Il confronto con le altezze prescritte in partitura (racchiuse all’interno di un intervallo di quinta diminuita $L_{a2}-Mi_{b3}$), mostra che in realtà il primo suono, ‘Metal Triad Power E1 LP’, non è mai impiegato.

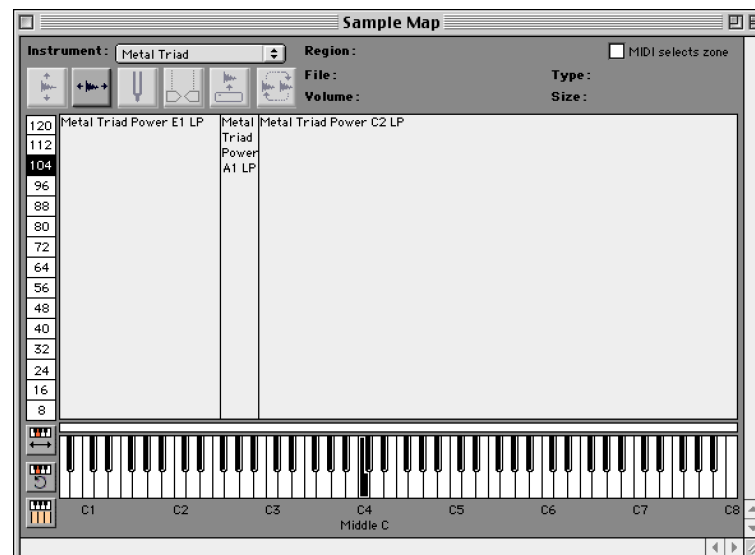


Figura 80 – ‘Metal Triad’. Distribuzione dei campioni nel dominio delle altezze.

I suoni vengono inoltre controllati da una funzione ADSR (*attack, decay, sustain e release*) che ne descrive l’inviluppo globale:

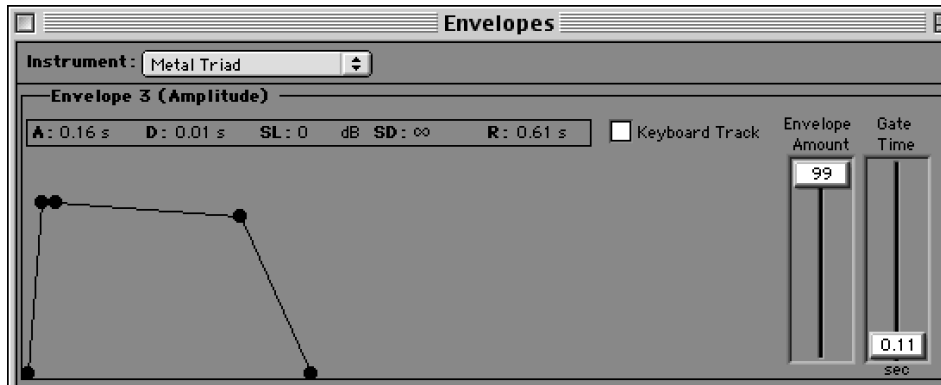


Figura 81 – Grafico ASDR per i suoni ‘Metal Triad’. Si noti come la fase di sustain non sia vincolata ad una durata precisa bensì dipenda dall’azione dello strumentista sulla tenuta del tasto.

I suoni campionati possono essere riprodotti in *loop*, gestendo la transizione tra l’estinzione e l’attacco con *crossfade*. Ad esempio, per il campione denominato ‘Metal Triad Power C2 LP’ della durata di circa 4 secondi (176,463 campioni a 44,1 kHz), sono impostati i parametri in figura 82:

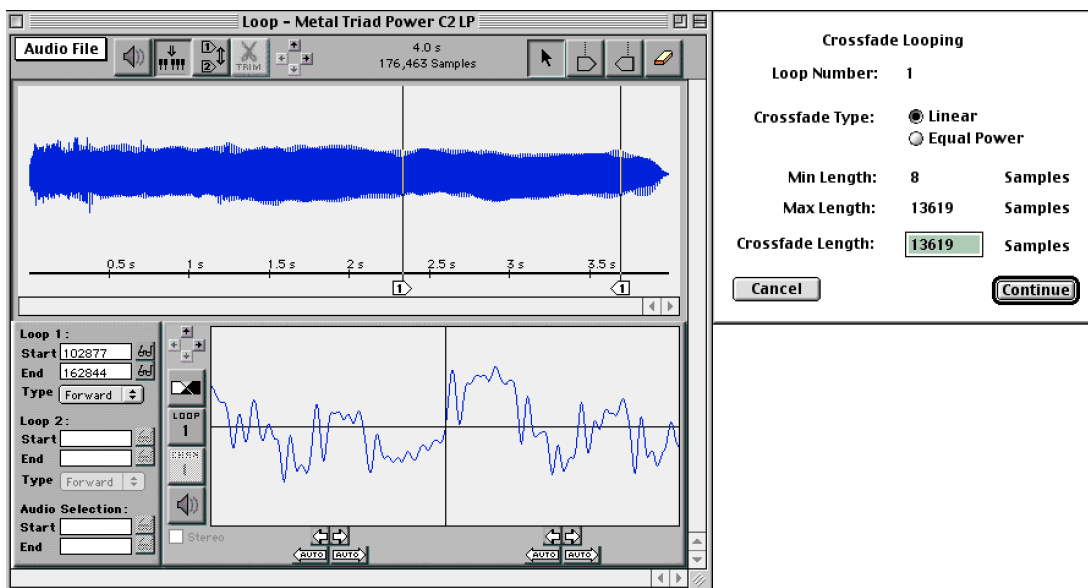


Figura 82 – Inizio del loop a 102877 campioni; fine del loop a 162844 campioni; tipo di crossfade: lineare; lunghezza del crossfade: 13619 campioni.

I set ‘Strings’ e ‘Strings +1/4’ condividono invece un unico, singolo campione sonoro. La differenza tra i suoni è marcata da un preciso parametro, il *Detune*, che come si può notare in figura 83 è innalzato di 50 cents, un quarto di tono:

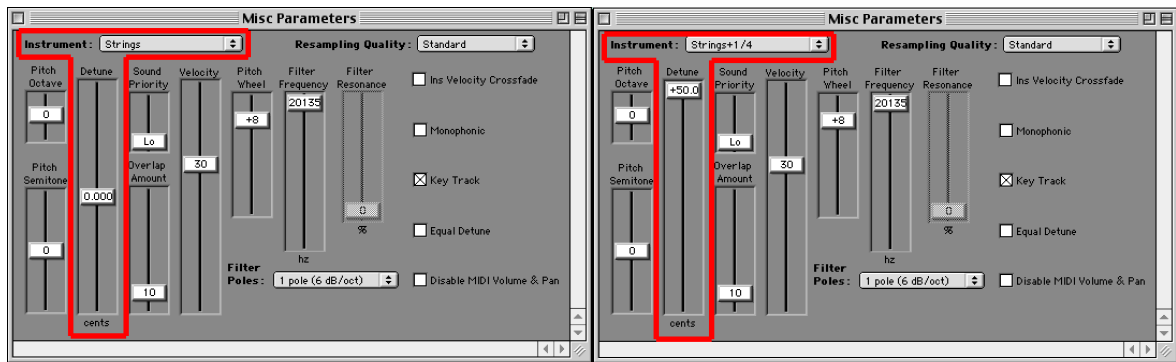


Figura 83 – Confronto tra i set ‘Strings’ e ‘Strings +1/4’, con il dettaglio del parametro Detune.

Per ciò che concerne i suoni di pianoforte e di chitarra elettrica, la scelta del campionario sembra essere dettata da vincoli imposti dalla situazione performativa più che da un reale interesse verso l’utilizzo dello strumento elettronico. Nelle note alla partitura, infatti, dopo aver descritto i suoni del campionario, Romitelli sottolinea: «Tuttavia altre soluzioni sono possibili; ad esempio è possibile immaginare l’utilizzazione di una chitarra elettrica ‘live’, o di un pianoforte acustico anche per la tast. II [il tastierista al campionario]». ²²⁶

Per gli ‘archi filtrati’, suono sintetico probabilmente prelevato da un sintetizzatore, non sono invece previste, né si possono immaginare, soluzioni esecutive diverse dall’impiego di un campionario.

A.4 Le performance

Come si è visto, anche per *EnTrance* problemi di accesso e interpretazione dei dati digitali minacciano la sua rieseguitività. Il problema si era già posto nel 2005 quando l’opera è stata registrata e poi pubblicata per Stradivarius nel CD *Romitelli: Audiodrome*. ²²⁷ La versione, curata da Denis Lorrain, è stata inoltre utilizzata quando il brano è stato inserito in cartellone nel festival Manca 2008. ²²⁸

²²⁶ FAUSTO ROMITELLI, *EnTrance*, p. iv.

²²⁷ FAUSTO ROMITELLI, *Audiodrome*, Stradivarius, 2007. *EnTrance*, seconda traccia del CD, è stata registrata nel 2005: Orchestra sinfonica nazionale della RAI, Peter Rundell direttore, Donatienne Michel-Dansac soprano.

²²⁸ Concerto del 22 novembre 2008 presso l’Opéra de Monaco. Orchestre Philharmonique de Monte-Carlo, Jean Deroyer direttore, Françoise Kubler soprano.

È sufficiente analizzare gli schemi tecnici della regia del suono (figure 84 e 85), per comprendere come il panorama mediatico fosse già allora mutato e quali accorgimenti informatici si resero necessari.

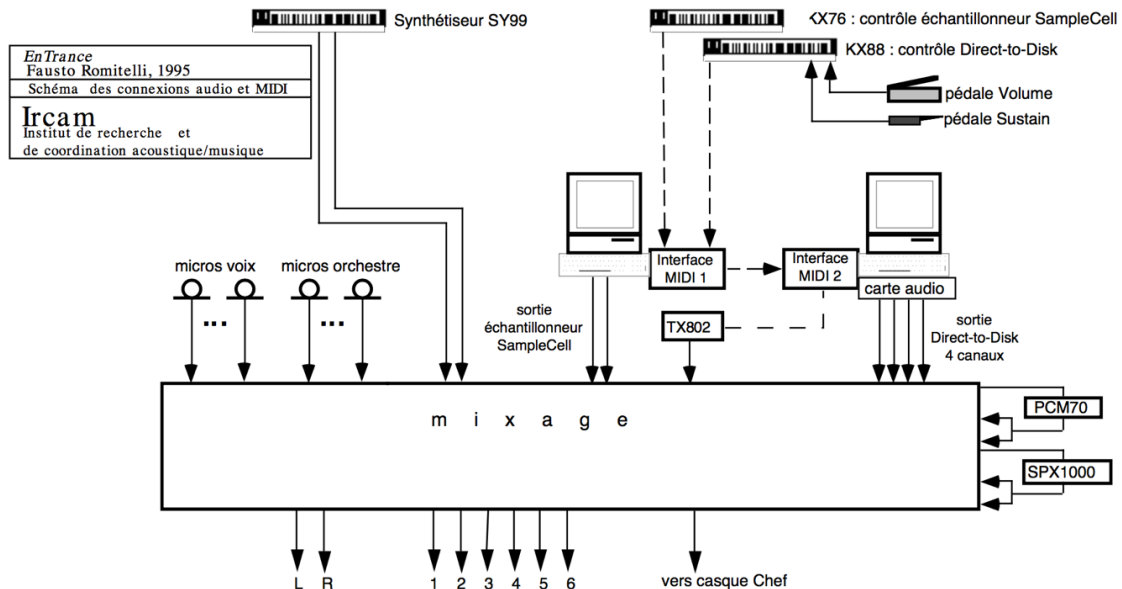


Figura 84 – Il setup originale per la regia del suono di EnTrance.

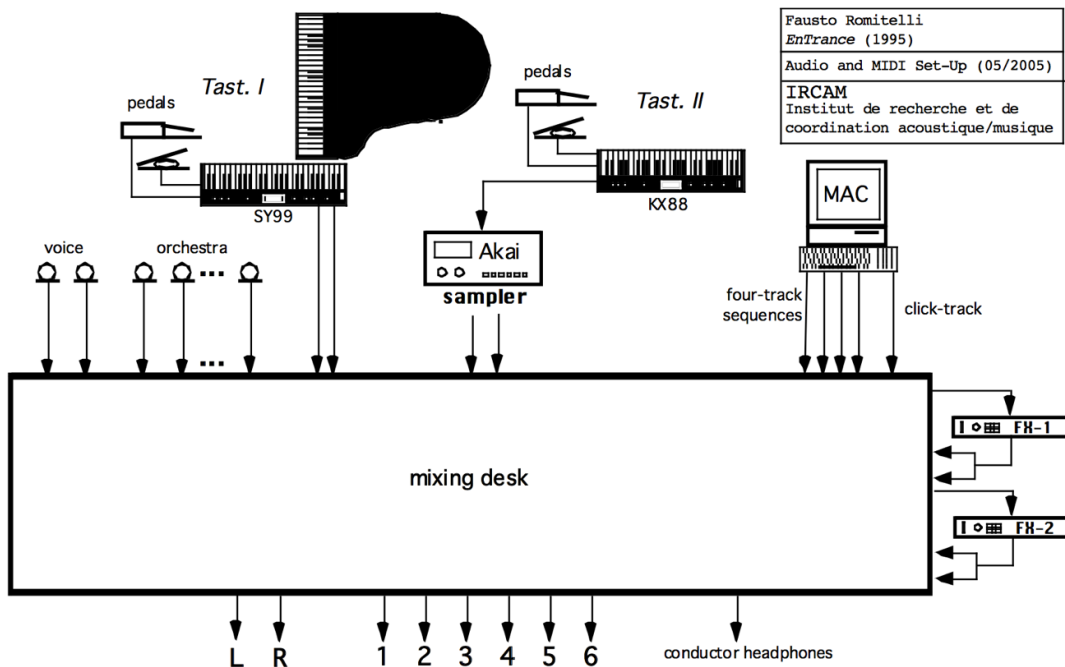


Figura 85 – Il setup di Denis Lorrain.

Come si legge nelle note di esecuzione redatte da Lorrain:

The electroacoustic set-up of the present 2005 version of *EnTrance* is simplified compared to the original described pp. iv-vi of the score.

1. The electronic parts (four-track audio sequences, figured on the two lowest staves of the score) are **triggered by hand** from a Max/MSP patcher running on a Macintosh computer. **These include the conductor click-track**, on a fifth audio track.

The following equipment is thus **no longer needed**, being replaced by manual triggering and the Max/MSP patcher:

- second "KX76" MIDI keyboard for the second keyboardist (*Tast. II*)
- "TX802" synthesizer expander
- "studio2" MIDI interface [...]

2. The "KX88" MIDI keyboard for the second keyboardist (*Tast. II*) is connected directly to an Akai S5000 **sampler**. The following equipment is thus **no longer needed**, being replaced by the Akai S5000 sampler:

- "studio5" MIDI interface
- "Mac I" computer running SampleCell.²²⁹

A una patch *Max/MSP*, controllata dal regista del suono, veniva affidato quindi il compito di gestire la riproduzione delle parti elettroniche e la generazione della *click track* (nella versione originale affidata al *TX802*). Il problema del sintetizzatore non si poneva, dal momento che all'epoca non era difficile rinvenire strumenti ancora perfettamente funzionanti. Di contro, il campionatore *SampleCell*, ormai obsoleto, doveva essere sostituito con un apparato hardware che svolgesse le stesse funzioni: *l'Akai S5000*.

Dal 2005 a oggi la situazione è ulteriormente cambiata: oltre al persistere della necessità di sostituire il campionatore, inizia ad affacciarsi il problema della reperibilità di un sintetizzatore *SY99* funzionante. Le patch di *Max/MSP*, invece, sono ancora leggibili, grazie alla retrocompatibilità offerta dall'ambiente software. Andranno però adattate al nuovo setup che il regista del suono vorrà allestire. Le scelte non saranno solo estetiche, ma anche dettate dalla strumentazione reperibile.

Il lavoro condotto per questa tesi sulle fonti digitali di *EnTrance* costituisce un suo ulteriore 'aggiornamento' informatico. L'opera è oggi di nuovo eseguibile. Tuttavia rimane aperta una questione fondamentale: se non si formerà una *community* di compositori,

²²⁹ Denis Lorrain, *Fausto Romitelli EnTrance_2005 Performance Handbook*, IRCAM, Parigi, 2005, p. 1.

musicisti e informatici con il compito di prendersi carico di quel costante lavoro di revisione – già intravisto da Grisey – necessario a fronteggiare la minaccia dell'obsolescenza tecnologica, ben presto di *EnTrance* e di tutto l'immenso patrimonio musicale 'nativo digitale' resterà ben poca memoria. Ma la situazione non è nuova nella storia della musica. A ben vedere, è ciò che accade anche con la memoria della prassi esecutiva, da sempre tramandata oralmente nel rapporto maestro-allievo. L'urgenza ormai è manifesta, e come dimostrano i recenti studi in ambito archivistico e musicologico, la comunità scientifica, assorbito il violento *choc* mediatico subito a cavallo tra gli anni Ottanta e Novanta, sta già iniziando a fare i conti con la nuova memoria digitale.

APPENDICE B: FAUSTO ROMITELLI. *PERTINENCE DU TIMBRE*

Viene riportato nelle prossime pagine uno scritto inedito di Fausto Romitelli, conservato presso l'archivio privato di Laurent Pottier. Lo scritto, redatto tra il 1992 e il 1995, consente di far luce sulla maturazione del pensiero musicale del compositore. Ancorandosi ai fondamenti della linguistica, Romitelli sviluppa una riflessione sul timbro che si rifletterà in modo evidente sulla composizione di *EnTrance*. Non a caso, il testo fu consegnato a Pottier come base concettuale nel momento in cui i due iniziarono a lavorare all'opera.

Pertinence du timbre si articola in due parti: il quadro teorico delineato nella prima parte trova applicazione nella seconda, con esempi tratti da *Natura morta con fiamme* (1990-1991) e *Les idoles du soleil* (1992).²³⁰ Gli esempi richiamati all'interno del testo sono riportati, trascritti, in calce.

²³⁰ Nel testo originale, le due parti sono separate dallo scritto *Résonance*, pubblicato già nel 1993 in Christine Buci Glucksmann e Michael Levinas (a cura di), *L'Idée musicale*, Vincennes-Saint-Denis, Presses Universitaires de Vincennes, 1993, pp. 43-45.

Fausto Romitelli

Pertinence du timbre

Je voudrais démontrer, en définissant une comparaison avec le modèle linguistique, que la **coordination** entre la **concurrence d'entités simultanées**

- c'est à dire l'ensemble des caractéristiques de la morphologie spectrale qui déterminent la perception du timbre à partir d'un processus de **catégorisation** et de **projection dans l'espace acoustique** -

et la **concaténation d'entités successives**

- c'est à dire l'articulation d'objets et leur combinaison à différentes niveaux hiérarchiques à partir d'un processus de **discrétisation** et de **projection dans le temps** -

n'est pas seulement la base de la communication linguistique, mais aussi un outil puissant pour contrôler la formalisation de structures musicales.

Contrôler un processus musical signifie maîtriser la complexité d'un phénomène déterminé par l'interaction de différentes catégories **fonctionnelles**:

les opérations **dans l'espace** (du timbre)

- caractérisées par la logique d'adhérence (Bonnet), la transformation d'ensembles, la texture -

doivent être **fonctionnelles** à les opérations **dans le temps**

- caractérisées par la logique combinatoire (algèbre), l'articulation d'objets, le geste.

Donc:

un processus musical doit gérer l'évolution d'un objet dans l'espace acoustique

- en le intégrant dans un système phonologique-

et dans le temps

- en le définissant dans un système syntaxique.

Discrétisation et temps

Le discours musical exige une **dialectique temporelle**.

Le **temps est pluriel**: l'oeuvre est une texture tissée par diverses lignes du temps, chacune parmi elles convoie un ordre différent d'objets. Le principe d'identification est la base de la dialectique temporelle: la perception de différentes trajectoires est liée à la possibilité de reconnaître, mémoriser, suivre dans leur successifs, apparitions et transformations les objets projetés dans *les temps* de l'oeuvre.

La notion d'objet est donc fondamentale: l'articulation des différents temps est possible seulement en partant de la définition d'un repertoire d'objets discrets (chacun projeté dans des orbites temporelles distinctes mais intersecantes), et eux mêmes à leur tour constitués à partir d'une série limitée d'éléments plus petits.

Tout l'ensemble se fonde sur un principe de "délimitation" d'ordres concentriques: le matériel amorphe (le continuum sonore) doit être segmenté en différentes séries d'éléments finis.

La discrétisation est alors la *conditio sine qua non* de la formalisation du discours musical: si un langage formel est tout ensemble de séquences finies de symbols tirés d'un alphabet fini, c'est clair que sa existence est liée à la possibilité de définir un repertoire de variables discrètes et de règles de combinaison.

Donc:

1) Il n'y a point de communication sans articulation: le langage est cela qui divise le réel.

2) Il ne peut pas y avoir composition sans logique combinatoire: le travail compositionnel est lié à la manipulation d'unités élémentaires,

leur organisation dans des systèmes complexes de relations et leur propulsion dans les lignes du temps de l'oeuvre.

Cependant, il y a une considération importante à faire:

la logique combinatoire risque de créer un système absolument *relatif* dans le quel le jeu entre différence et répétition, variant et invariant, n'assume pas une dimension vraiment fonctionnelle et hiérarchique, mais il s'intègre dans un espace où tension et détente sont des qualités abstraites, conventionnelles, relatives et elles ne sont pas perçues comme *nécessaires* ; à partir d'une logique combinatoire, on peut différencier l'agrégat x de l'agrégat y , mais on ne peut pas "établir" par convention que à x correspond la perception d'une "tension" et à y d'une "détente"; pour réaliser cette opposition il faut définir des aires de "consonance" et de "dissonance" dans l'espace acoustique à partir de la valuation à niveau perceptuel des interrélations entre les qualités morphologiques intrinsèques des sons et des ensembles des sons: par exemple, le degré d'harmonicité du contenu spectral.

C'est donc un processus perceptif de catégorisation que définit la "courbe consonantique" de l'espace sonore en "hiérarchisant et fonctionnalisant" les objets et les agrégats individualisés par un processus de discrétisation.

Es que nous revenons alors à soutenir la nécessité d'un fondement naturel? La nécessité est très simplement celle de connaître et exploiter les conséquences logiques des lois de la perception acoustique.

Catégorisation et espace

On a souligné que les systèmes complexes ne sont pas "donnés" mais "ordonnés" à partir de listes finies d'éléments ultime discrets: tout composant du matériel de base doit être neutre et abstrait pour devenir objet d'une manipulation, pour être *agi*. "Cependant le son est un système complexe par définition et on ne peut pas le considérer uniquement un élément ultime, symbol abstrait d'une pensée algébrique, muni d'une valeur purement différentielle; le son est processus par autonymie, organisme en changement permanent, absolument instable et projeté dans le temps et dans l'espace. C'est justement cette dimension dynamique intrinsèque du

son, liée à la perception catégorielle du timbre, qui est capable d'exercer une fonction primordiale dans la perception de l'organisation syntaxique des objets musicaux: on a vu que c'est bien la configuration de la morphologie spectrale des sons et des agrégats de sons à donner une base hiérarchique *naturelle* au discours musical.

Notres lois conventionnelles d'organisation doivent *reagir* avec les lois naturelles du son pour "entrer en collision" ou, au contraire, se fortifier réciproquement, en établissant une dialectique capable de rendre vraiment perceptibles les différents degrés d'attraction- repulsion entre les composants d'une structure musicale.

L'oeuvre ne doit pas se limiter au jeu de lignes, de surfaces, mais elle doit les insérer dans une espace, les projeter dans une perspective, elle doit assumer une connotation *en profondeur*, nécessaire à la "fonctionnalisation" des parcours: il faut définir une coordination entre la propulsion des objets dans les trajectoires temporelles de l'oeuvre et leur *mise en perspective* dans les champs de tension et détente de l'espace acoustique.

Dans un'oeuvre instrumentale la profondeur est liée à une *écriture de synthèse* capable d'activer la perception d'une image sonore ambiguë, dont les composantes sont dans un état de fusion instable; la perception oscille entre la détection d'une ou plusieurs sources: la même image semble provenir de localisations et distances différentes, c'est à dire elle semble s'étendre dans l'espace sur différents points. On a l'illusion de percevoir diverses surfaces du même objet: c'est l'illusion d'un **volume**.

(Auguste Rodin raconte que son maître lui conseillait de considérer la surface comme étant rien d'autre que l'extrémité d'un volume.)

Une *écriture du timbre* est donc nécessaire, non pas pour rendre moins banal et attendu l'*image sonore globale* de l'oeuvre, mais pour rendre l'organisation syntaxique *pertinente* à l'écoute.

On peut trouver la racine historique de cette conception dans les techniques d'orchestration visantes à souligner les trajectoires formelles, plutôt que faire "couleur". Un exemple extraordinaire est constitué par "Nuages" de Debussy; l'analyse de cette oeuvre par Robert Cogan démontre que

"The sonic design results, then, from precise coordination of the movement of fundamental pitches with transformations of instrumental spectra and dynamics. Other possible choices instrumental sound and dynamics would destroy this spectral design."

Une référence au modèle linguistique sera nécessaire pour mieux comprendre le problème de cette interrelation entre concurrence et concaténation.

Le double caractère du langage

Les deux axes du langage sont ces de la combinaison et de la sélection.

Combinaison: chaque unité linguistique est, au même temps, un agrégat de unités plus simples et une composant d'un agrégat plus complexe. C'est le plan du syntagme: les termes sont unis *in praesentia*, l'articulation a comme support l'extension linéaire; les composants sont en un rapport de contiguïté: l'expression condensée de cet plan de l'articulation est la métonymie.

Selection: chaque unité linguistique peut être remplacée par un autre a la fois équivalent et différent. C'est le plan du paradigme ou système: les termes sont unis *in absentia* et forment une "série mnémonique virtuelle"; les composantes sont unies par similarité et l'expression condensée est la métaphore.

Dans la langue le morphème est la composante ultime qui possède sa propre signification .

Le morphème est constitué d'une séquence de phonèmes (éléments non signifiants) combinés sur la base d'une série de règles de concaténation; chaque phonème est constitué d'un faisceau de traits distinctifs simultanés, c'est à dire des caractéristiques d'émission dont la présence-absence détermine l'identité du phonème et le différencie des autres. En commutant un phonème on obtient un changement de sens. Donc chaque communication se produit à partir d'oppositions organisées en système.

"On peut donc dire que la concurrence d'entités simultanées et la concaténation d'entités successives sont les deux modes selon

lesquels nous, sujets parlants, combinons les constituants linguistiques."(Jakobson)

Chaque discours est la concaténation réglée à plusieurs niveaux (**système syntaxique**) d'éléments ultimes (phonèmes) déterminés par la présence simultanée de traits distinctifs (**système phonologique**).

Nous reconnaissons donc une unité de discours, non sur la base de chaque détail acoustique d'émission, mais sur la base de **certaines oppositions fonctionnelles constantes**.

Notre intérêt n'est pas alors dans la phonétique (étude des propriétés acoustiques et articulatoires du langage) mais dans la **phonologie** (étude des sons pertinents au système du langage, c'est à dire des caractéristiques d'articulation qui " s'insèrent dans un système d'oppositions fonctionnantes dans le paradigme d'une langue, dans le but d'une combinaison syntagmatique d'unités significatives". (Eco).

A ce point, notre objectif apparait clairement:

établir, à partir du modèle linguistique, la possibilité d'agir sur les caractéristiques pertinentes de la morphologie spectrale afin de créer un système d'oppositions capables de "différencier, cimenter, compartimenter, ou de mettre en relief" les diverses unités de combinaison syntagmatique.

Les opérations sur le timbre doivent donc être nécessaires et pertinentes à l'articulation syntaxique: les caractéristiques spectrales doivent définir les niveaux hiérarchiques, les fonctionnalités des composantes syntaxiques à différents niveaux de combinaison, des contextes locaux aux contextes généraux.

Les règles phonologiques interagissent avec les règles syntaxiques, les unes existent en fonction des autres.

Le compositeur trouve dans **l'informatique l'outil capable de gérer cette interrelation**.

Dans "L'aspect phonologique et l'aspect grammatical du langage dans leur interrelations", Roman Jakobson démontre

"Que des traits phonologiques différentes soient choisis et utilisés de différentes manières selon les catégories grammaticales, est un fait essentiel pour la compréhension des traits phonologiques de la langue en question et des relations hiérarchiques qu'ils ont entre eux.

... l'étude d'un système grammatical conduit inévitablement au problème des moyens phonologiques mis en oeuvre pour exprimer les différentes catégories grammaticales de la langue en question."

Je refuse donc l'idée d'une pensée combinatoire qui articulerait des objets en se désintéressant du contexte qui est inévitablement créé au niveau de la morphologie spectrale; je refuse aussi l'idée de tracer deux trajectoires indépendantes et parallèles: une articulation syntaxique d'objets au-dessus de laquelle émerge un dessin spectral, un parcours dans le timbre;

Je refuse enfin une conception dialectique qui oppose les opérations sur l'harmonie et l'articulation d'objets discrets, à les opérations sur le timbre et à l'articulation d'ensembles.

Aujourd'hui, la pensée que je trouve riche de potentialités, est celle qu'en partant d'une interprétation du modèle linguistique, définie une morphologie spectrale en fonction de l'articulation syntaxique; *comme, dans le langage, la coexistence de caractéristiques spectrales simultanées, qui déterminent l'identification des phonèmes, a une valence différentielle et non-signifiante et c'est l'enchaînement des phonèmes à déterminer la constitution des unités significatives; ainsi, dans le discours musical, les opérations sur le timbre doivent accomplir une fonction distinctive, différentielle: elles doivent définir l'objet dans un contexte syntaxique, non seulement en le séparant des autres objets et ainsi permettant sa reconnaissabilité et mémorisation, mais aussi en déterminant la logique de sa présence dans le contexte.*

On peut dire exactement le même de l'espace(; sa fonction est complémentaire à celle du timbre: il doit différencier les objets ainsi que les groupes d'objets pour faciliter leur organisation syntaxique. Aussi bien dans ce cas, il faudra individuer les traits pertinents de la représentation spatiale, c'est à dire ces caractéristiques relatives à la localisation, à la profondeur et à l'image de la source, qui sont capables d'exercer une fonction distinctive en se intégrant dans un système de oppositions et en déterminant l'identification des objets dans la chaîne syntaxique.

Par quels moyens peut-on agir sur la morphologie spectrale pour la rendre pertinente à l'élaboration syntaxique?

Je crois à l'interaction de trois systèmes de production:

1) l'instrument acoustique, considéré comme producteur d'une image spectrale dont les caractéristiques peuvent agir au niveau fonctionnel sur l'image globale.

"... the essential sonic features of any musical instrument are not finally to be discovered either in the practice room or in the anechoic chamber ... they are to be found in the sum total of its structural sonic contributions to musical context." (Cogan)

Dans ce cas la capacité de modification est nulle; il y a uniquement la possibilité de choix de l'instrument.

2) l'agrégat d'instruments, considéré comme image fusionnée individuelle où les composantes perdent partiellement leur identité pour contribuer à un timbre global. Dans ce cas on a une capacité à peu près illimitée pour construire des images de "synthèse instrumentale", celles-ci étant capables de différencier et hiérarchiser les unités d'articulation syntaxique; ce pouvoir est tant de même compensé par le bas degré de fusion des composantes, qui tendent à faire émerger leurs propres identités spectrales.

3) l'ordinateur: l'utilisation de sons de synthèse (ou de sons élaborés par ordinateur) est le seul cas où notre capacité de gérer la morphologie spectrale est absolue.

Maintenant je voudrais expliquer la nécessité de cette interrelation entre la concurrence d'entités simultanées (timbre, catégorisation) et concaténation d'entités successives (articulation, discrétisation) en partant d'une série de considérations complètement différentes, relatives aux problèmes de la **signification musicale**.

Pertinence du timbre - deuxième partie

Dans cette deuxième partie on va présenter 2 exemples de réalisation musicale, afin de clarifier les idées exposées dans la première partie et définir la fonction de l'ordinateur dans ma perspective compositionnelle.

Première exemple

Avec le premier exemple, qui vient du travail préparatoire (dans le domaine harmonique) à ma pièce "Les idoles du soleil", je voudrais démontrer dans quelle façon un système syntaxique constitué par la concatenation et la combinaison réglée de variables discrètes (intervalles), peut être hiérarchisé par sa intégration dans un système phonologique, c'est à dire dans un système d'oppositions.

1.

1.1

Notre point de départ est constitué par 3 agrégats d'intervalles:

Fig.1

Les agrégats diffèrent dans le nombre et conformation des intervalles, la densité, le registre, l'amplitude.

1.2

Chaque agrégat est un répertoire de variables discrètes: en établissant un répertoire de règles de combinaison nous pouvons engendrer les ensembles A, B, C.

Fig.2

1.3

A partir des propriétés constitutives de A, B, C on veut définir le modèle d'une "mise en relation" des objets sonores. Nous établissons un repertoire de règles pour contrôler la transition entre les agrégats .

Règles:

1) La transition entre 2 agrégats est possible lorsque ils ont 2 intervalles consécutifs communs.

2) La transition entre 2 agrégats est possible lorsque ils ont 2 notes communes.

3) Chaque objet d'un ensemble A, B, C ne peut pas se mettre en relation avec plus de 2 autres objets d'un autre ensemble; si une transition est possible avec plus de 2 objets, ils seront choisis sur la base de leur ordre numérique.

4) Les ensembles doivent être explorés de gauche à droite (de A1 à A12, pour exemple) sans possibilité de revenir en arrière.

On construit le graphe d'intersection entre les 3 ensembles:

Fig.3

(A partir de cet graphe on pourrait imaginer la constitution d'un automate pour l'analyse exhaustive des parcours possibles)

1.4

C'est évident que les trois parcours ne peuvent pas être explorés dans leur totalité à partir de ces règles de transition: il sera nécessaire réaliser des liaisons "pour contraste" entre objets "non similaires"; par ailleurs, les transitions "réglées" n'ont pas toutes la même évidence perceptive: la transition par règle 1 est moins pertinente à l'écoute de la transition par règle 2, qui implique une communauté de registre entre les deux agrégats. Le compositeur peut imaginer de gérer les connections par transition réglée, contraste, ou bien par les étapes intermédiaires entre l'un et l'autre, pour réaliser une "courbe de tension" dans le domaine harmonique.

2.

2.1

Nous avons engendré 3 ensembles d'objets sonores et établi un modèle de "mise en relation" à partir de la conformation de leurs intervalles. Maintenant on veut les projeter dans un espace de timbres: nous allons les considérer pas plus comme des produits d'une logique combinatoire, des objets discrets constitués à partir de l'articulation réglée d'éléments abstraits et relationnels, mais comme des ensembles de fréquences qui tendent à la fusion perceptive dans des images synthétiques: des timbres placés à l'intérieur d'un espace multidimensionnel comme produits d'une transition entre modèles opposés.

"Un espace de timbres est déterminé à partir d'un choix de modèles. Ceux-ci sont distribués en fonction d'une catégorisation psychologique arbitraire projetée dans un espace à x dimensions."
(Barrière)

A partir de l'expérience de la phonologie, on définit les objets à l'intérieur d'un système d'oppositions: tout objet se place dans un espace de transition entre les termes d'une opposition:

harmonique/inharmonique, grave/aigu, compressé/expansé, étroit/large, central/extrême, non-espacé/espacé, non-attaque/attaque, etc.

2.2

On peut considérer chaque agrégat comme:

1) Fondamentale commune.

Un ensemble de partiels du même spectre avec une fondamentale virtuelle.

Fig.4-a

2) Superposition de fondamentales.

Un ensemble de n fréquences dans lequel chaque fréquence est le partial n d'une fondamentale différente: n fréquences est la superposition de n spectres. C'est la dimension la plus riche de possibilités de formalisation: chaque composant d'agrégat peut être "harmonisée" par des autres fréquences afin de synthétiser un spectre "autour" la composant et la placer dans un espace timbrale. Un agrégat d'intervalles, perçu comme un objet discret, peut devenir, à la suite d'un processus d' "harmonisation", un élément "collected into a composite image, wherein the object loses its identity but contributes to the quality of the more embracing image"(McAdams).

Fig.4-b

2.3 Fondamentale commune

En utilisant la fonction virt-fund dans PatchWork (avec une approximation de 25 cents, 1/8 de ton), on peut calculer les fondamentales virtuelles de chaque agrégat:

Fig.5

2.4 On veut définir un modèle de mise en relation des agrégats à partir de leur fondamentales virtuelles; on établit la règle suivante:

la transition entre 2 agrégats est possible si leur fondamentales appartiennent à un même spectre.

On construit le graphe d'intersection entre les 3 ensembles:

Fig.6

2.5 Superposition de fondamentales

Chaque parcours A,B,C, constitué par l'articulation de variables discrètes, peut être intégré dans un processus de transformation entre modèles timbrales opposés; j'ai agi dans la manière suivante:

a) J'ai harmonisé les agrégats de B avec des spectres harmoniques (j'ai considéré toute fréquence de chacun de 5 agrégats de l'ensemble B comme partiel n d'un spectre harmonique).

Fig.7

b) J'ai harmonisé les agrégats de A et C avec des spectres progressivement distordus (j'ai considéré toute fréquence de toute agrégat A et C comme le partiel n d'un spectre progressivement distordu à partir d'un modèle constitué par le spectre harmonique).

La fonction Lisp de distorsion est la suivante:

Fig.8

A partir d'une autre fonction Lisp on calcule l'interpolation exponentielle en 12 étapes entre $\text{dist}=1$ et $\text{dist}=2$.

Fig.9

c) A partir de cette fonction Lisp j'ai calculé aussi l'interpolation exponentielle entre les valeurs relatifs à la durée pour l'attaque et le decay du chaque chord-spectre de les ensembles A, B, C.

Fig.10

On a tracé donc une trajectoire entre les termes d'une série d'oppositions en définissant un espace à plusieurs dimensions. Pour exemple, les oppositions pur A sont:

harmonique / inharmonique

aigu / grave

non-attaque / attaque

compressé / expansé

ppp / fff

Deuxième exemple

1.1

Avec le deuxième exemple, qui vient de ma pièce "natura morta con fiamme" pour quatuor à cordes et sons de synthèse, je voudrais démontrer dans quelle façon un système syntaxique constitué par la concaténation et la combinaison réglée de variables discrètes (intervalles), peut être hiérarchisé par sa intégration dans un système phonologique, c'est à dire dans un système d'oppositions.

1.2

La pièce requiert un système d'amplification du quatuor par 8 haut-parleurs. Chaque instrument présente une configuration spatiale qui change pour chaque section mais reste fixe pendant la section elle même; le son bouge jamais de un haut-parleur à l'autre.

1.3

La section F est constituée par la succession de 13 patterns.

Chaque pattern est déterminée par un texture de 4 différentes figures: trois entr'elles sont présentes aussi dans le pattern successif, une est remplacée par une nouvelle figure; dans l'ensemble de la section, 10 figures apparaissent à rotation.

Dans chaque pattern, chaque figure est jouée successivement par chaque instrument, c'est à dire elle est répétée 4 fois. Toute fois une figure passe d'un instrument à l'autre, elle subit des

transformations dans la fréquence, l'amplitude, la durée, le timbre, l'articulation interne, etc.

Toute fois les figures bougent d'un instrument à l'autre avec une vitesse croissante, décroissante ou constante, selon un schéma d'accélération/décélération des trajectoires, elles bougent d'une configuration d'haut-parleurs à l'autre en donnant l'illusion d'un même son qui rebondit dans la salle et, en traversant l'espace, il modifie ses propriétés constitutives.

Fig.11 et 12

Le point de départ de chaque pattern est un interval de troisième mineure si-re qui subit une progressive distorsion vers le grave et vers l'aigu: lorsque un figure passe d'un instrument à l'autre (en bougeant dans l'espace), les fréquences de l'intervall sont baissées et élevées de $15 \text{ Hz} * n$, n étant un nombre entier progressivement croissant; l'image initiale, claire et consonante, devient opaque et se fragmente dans un texture très dense de figures qui bougent dans l'espace, et en bougant elles se transforment.

A la fin, la distorsion extrême des 2 fréquences coïncide avec les partiels d'un fondamentale à 32.7 Hz.

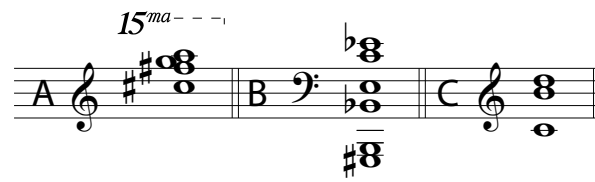


Fig. 1

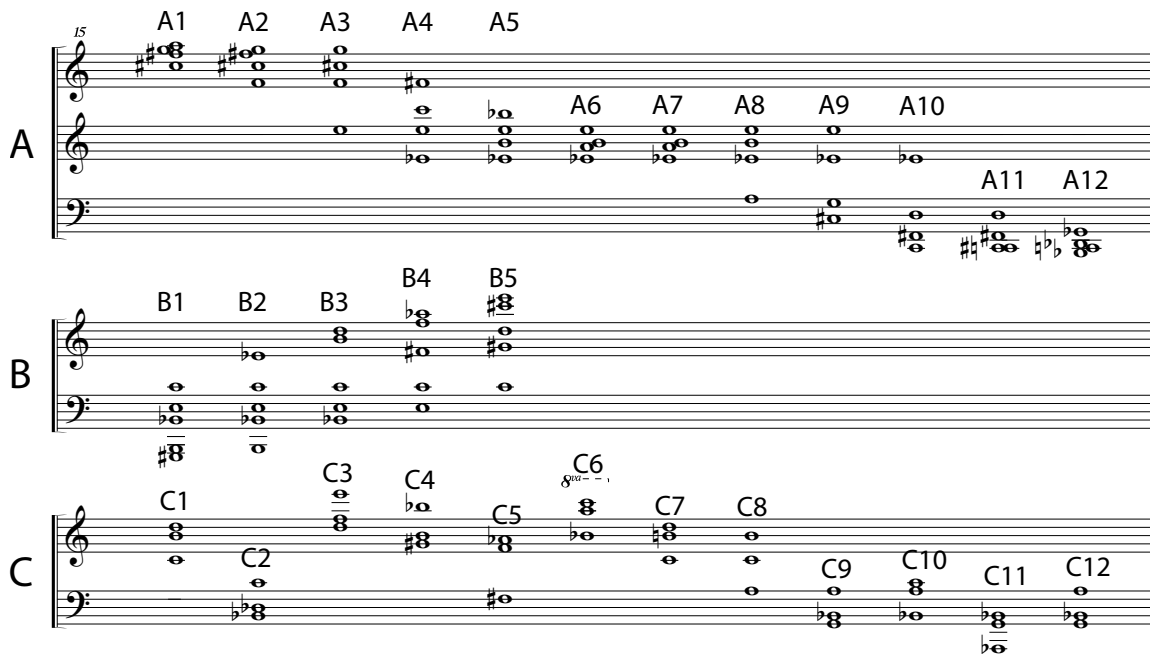


Fig. 2

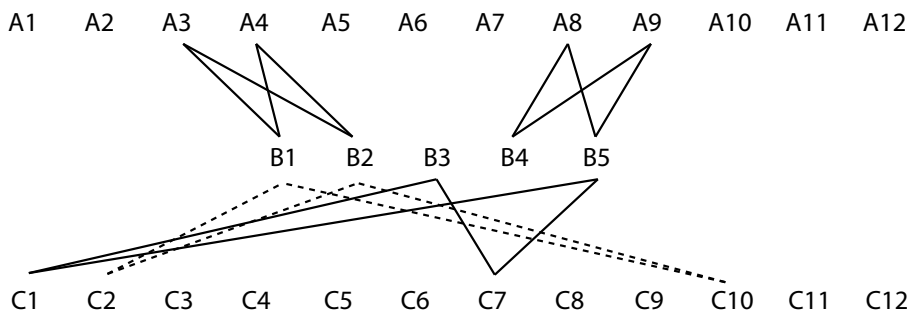


Fig. 3

4a

5 7 8 11

1

4b

5 4 3 2

1

Fig. 4a e 4b

A

1 2 3 4 5 6 7 8 9 10 11 12

15

C

1 2 3 4 5 6 7 8 9 10 11 12

15

B

1 2 3 4 5

15

Fig. 5

Transizioni

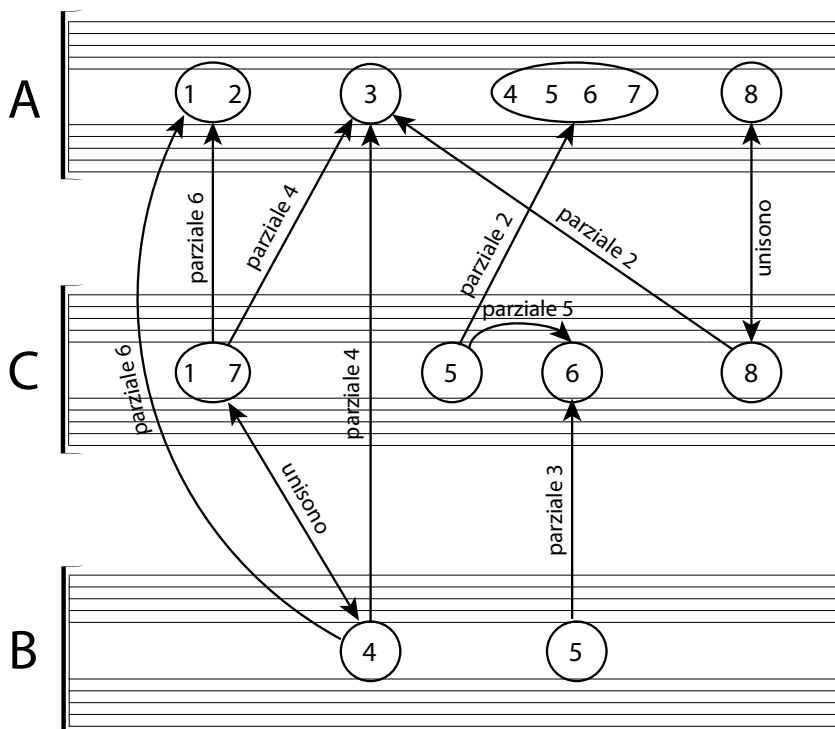


Fig. 6

Musical score for staff B, consisting of five measures. Above the staff are circled numbers 1, 2, 3, 4, and 5. The score includes treble and bass clefs. Below the staff, the first measure is labeled 'parziali 1,5,6' and the second 'idem'. The number 15 is written at the beginning of the staff.

Fig. 7

```
(defun dist (p freq dist)
  (let ((res))
    (dotimes (n p res)
      (setf res (cons (* freq (expt (+ n 1) dist)) res)))
    (reverse res)))
```

Fig. 8

1 dist = 1 2 dist = 1,027 3 dist = 1,067 4 dist = 1,12 5 dist = 1,185 6 dist = 1,263

15

parziali 1,3,5 parziali 1,3,5 parziali 1,3,5 parziali 1,3,5 parziali 1,3,5 parziali 1,3,5

15

7 dist = 1,354 8 dist = 1,458 9 dist = 1,574 10 dist = 1,703 11 dist = 1,845 12 dist = 2

15

parziali 1,3,5 parziali 1,2,3 parziali 1,2,3 parziali 1,2,3,4 parziali 1,2,3,4,5 parziali 1,2,3,4,5,6

15

1 dist = 1 2 dist = 1,027 3 dist = 1,067 4 dist = 1,12 5 dist = 1,185 6 dist = 1,263

15

parziali 1,2,3,4,5,6,7 parziali 1,3,4,5,6,7 parziali 1,3,5,6,7 parziali 1,3,5,6,7 parziali 1,3,5,7 parziali 1,3,5,7

15

7 dist = 1,354 8 dist = 1,458 9 dist = 1,574 10 dist = 1,703 11 dist = 1,845 12 dist = 2

15

parziali 1,3,5,7 parziali 1,3,5,7 parziali 1,2,3,5,7 parziali 1,2,3,5 parziali 1,2,3 parziali 1,2

A

C

Fig. 9

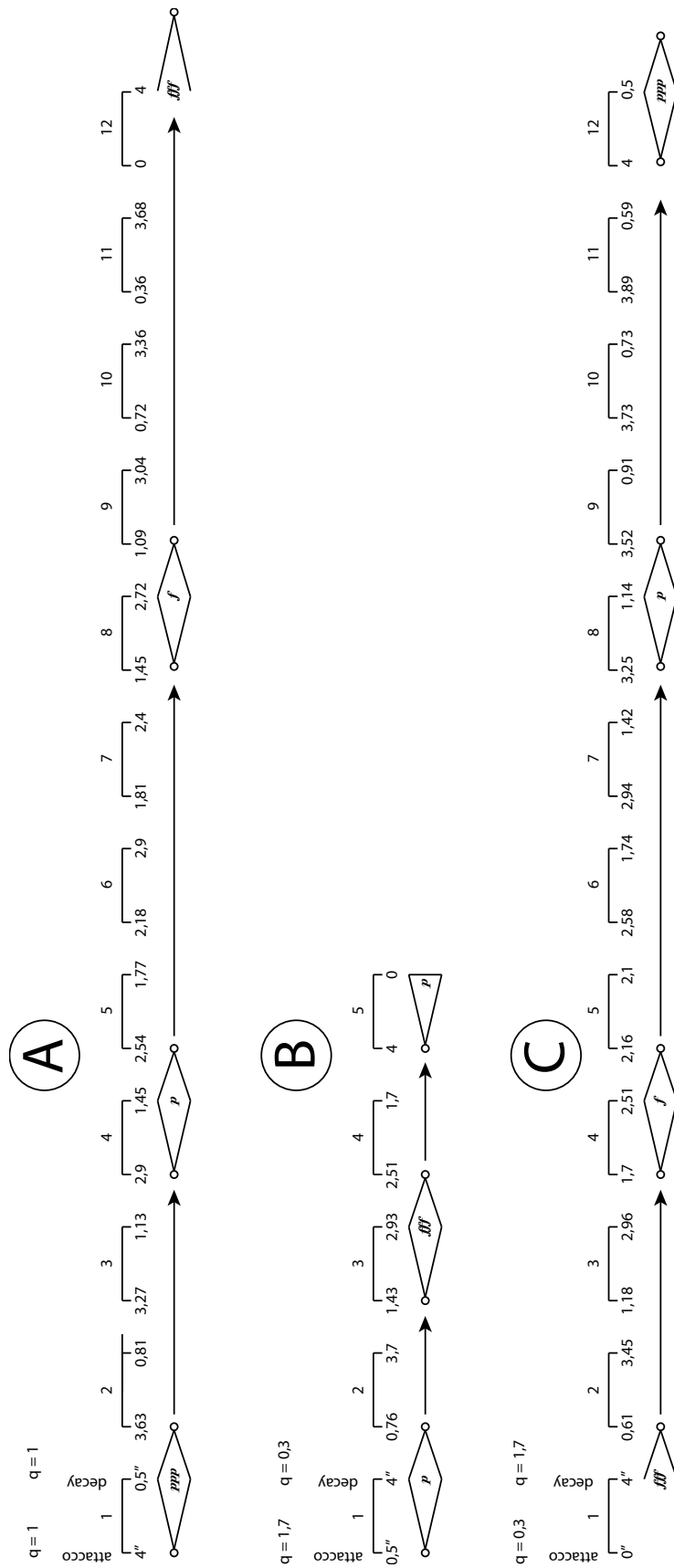


Fig. 10

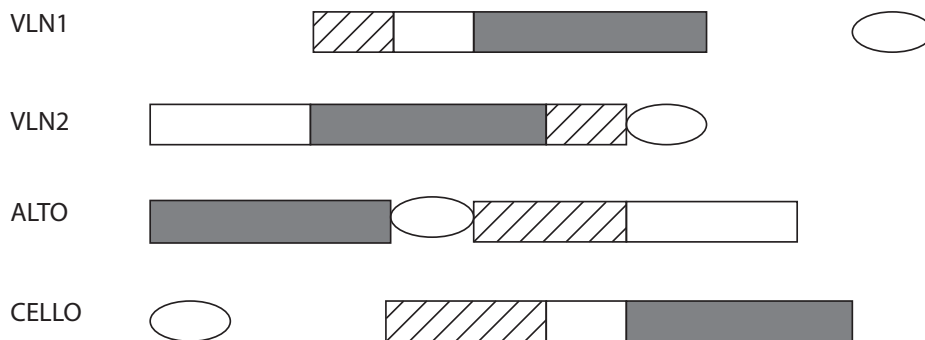


Fig. 11

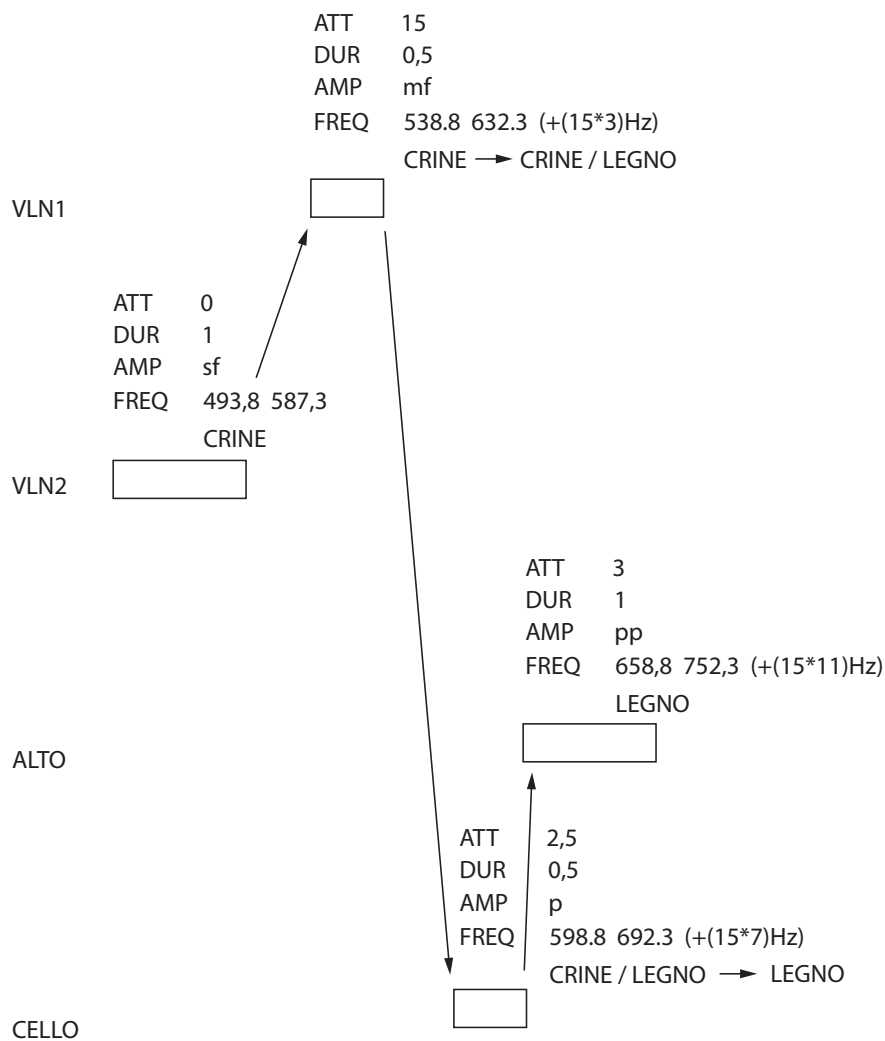


Fig. 12

APPENDICE C: CODICE LISP

C.1 Definizioni delle funzioni LISP in *lispdef*²³¹

```
(defun spectrum (f0 r nmax)
  (let ((l nil))
    (do ((i nmax (1- i)))
      ((<= i 0)l)
      (setq l(cons(* f0 r i)l))))))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(defun interp (a r q b)
  (list a b
        (let((add 0))
          (dolist (x (spectrum 1 1 (+ r 1)) 'begin/end)
            (cond
              ((> q 1)
               (setq add (+*(- q (*(/(- q (- 2 q))r)(- x 1))) (/(- b a)(+
                 r 1))) add)))
              (< q 1)
               (setq add (+(*(+ q (*(/(- (- 2 q) q)r)( - x 1))) (/(- b a)(+
                 r 1))) add)))
              (( = q 1)
               (setq add (*(/(- b a)(+ r 1))x)))
               (format t " ~S"
                 (+ a add)
                 ))))))))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(defun jinterp (a r q b)
  (let ((toto nil))
    (list a b
          (let((add 0))
            (dolist (x (spectrum 1 1 (+ r 1)) 'begin/end)
              (cond
                ((> q 1)
                 (setq add (+*(- q (*(/(- q (- 2 q))r)(- x 1))) (/(- b a)(+
                   r 1))) add)))
                (< q 1)
                 (setq add (+(*(+ q (*(/(- (- 2 q) q)r)( - x 1))) (/(- b a)(+
                   r 1))) add)))
                (( = q 1)
                 ))))))))
```

²³¹ Si veda *infra*, par. 3.2.3.

```

                (setq add (*(/(- b a)(+ r 1))x)))
                (push(+ a add)toto))
    ))(reverse toto))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(defun interpol (a r q b)
  (format t " ~S" a)
  (interp a r q b))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(defun jinterpol (a r q b)
  (cons a
    (jinterp a r q b)))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(defun Scopy (List1 List2 )
  (If (Null List1) Nil
    (If (Null List2) Nil
      (Append(List(Car List1)(Car List2))(Scopy (Cdr List1)(Cdr List2))))))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(defun SScopy (list1 list2 nom-fichier)
  (if (null list1) NIL
    (if (null list2) NIL
      (append(list(format nom-fichier "~8F,"(car list1))
                (format nom-fichier "~8F,"(car list2))
                (SScopy (cdr list1)(cdr list2) nom-fichier))))))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(defun zcopy (lista listb)
  (if (null lista) NIL
    (if (null listb) NIL
      (cons(scopy(car lista)(car listb))
        (zcopy (cdr lista)(cdr listb))))))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(defun ccopy (List nom-fichier)
  (If (Null List) Nil
    (Append(List(format nom-fichier "~4F,"(Car List))
      (ccopy (Cdr List)nom-fichier))))))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(defun chhcopy ( Lista nom-fichier)
  (let ((lista-1 (reverse(cdr(reverse lista)))))
    (If (Null Lista) Nil
      (Append(List(format nom-fichier"kenv linseg ~4F," (Car Lista-1))
        (ccopy (Cdr Lista-1)nom-fichier)
        (format nom-fichier"~4F-&"(car(reverse lista))))))))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(defun cycopy ( List nom-fichier)
  (If (Null List) Nil
    (cons(chhcopy (Car List)nom-fichier)
      (cycopy (Cdr List)nom-fichier))))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(defun lcopy (List nom-fichier)
  (If (Null List) Nil
    (Append(List(format nom-fichier "~8F,"(Car List))
      (lcopy (Cdr List) nom-fichier))))

```

```

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(Defun lhhcopy ( q Lista nom-fichier)
  (let ((lista-1 (reverse(cdr(reverse lista))))))
    (If (Null Lista) Nil
      (append (list
                (format nom-fichier " ~S ~8F," q (Car Lista-1))
                (lcopy (Cdr Lista-1)nom-fichier)
                (format nom-fichier"~8F-&"(car(reverse lista)))
                )))))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(defun pycopy ( list nom-fichier)
  (if(null list)nil
    (append(list(format nom-fichier"kenv~S "(car list))))))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(Defun lllcopy ( q Lista listb nom-fichier)
  (If (Null Lista) Nil
    (If (Null Listb) Nil
      (append(list
                (pycopy lista nom-fichier)
                (cons (lhhcopy q (Car Listb)nom-fichier)
                      (lllcopy q (cdr lista)(Cdr Listb) nom-fichier))))))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(defun addiz (n list)
  (if(null list)nil
    (cons (+ (- n 1)(car list))
          (addiz n(cdr list)))))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(defun xinterp (a r q b)
  (list a b
        (let((add 0))
          (dolist (x (spectrum 1 1 (+ r 1))'begin/end)
            (cond
              ((> q 1)
               (setq add (*(- q (*(/(- q (- 2 q))r)(- x 1))) (/(- b a)(+ r
               1))))))
              ((< q 1)
               (setq add (*(+ q (*(/(- (- 2 q) q)r)( - x 1))) (/(- b a)(+ r
               1))))))
              (( = q 1)
               (setq add (/(- b a)(+ r 1))))))
          (format t " ~S"
                (cond
                  ((>= b a)
                   add)
                  ((< b a)
                   (- (- add add )add))))
          ))))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(defun xjinterp (a r q b)
  (let ((toto nil))
    (list a b
          (let((add 0))
            (dolist (x (spectrum 1 1 (+ r 1))'begin/end)
              (cond

```

```

      (> q 1)
      (setq add (*(- q (*(/(- q (- 2 q))r)(- x 1))) (/(- b a)(+
        r 1))))))
      (< q 1)
      (setq add (*(+ q (*(/(- (- 2 q) q)r)(- x 1))) (/(- b a)(+
        r 1))))))
      (= q 1)
      (setq add (/(- b a)(+ r 1))))
(push
 (cond
  ((>= b a)
   add)
  (< b a)
  (- (- add add) add)))toto)
))) (reverse toto)
))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(defun wcopy (List1 List2 )
  (let ((x
        (if (null list1) nil
            (Append(List(Car List1)(Car List2))
                    (wcopy (Cdr List1)(Cdr List2))))))
    (remove nil x)))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(defun zcopy (lista listb)
  (if (null lista) NIL
      (if (null listb) NIL
          (cons(wcopy(car lista)(car listb))
                (zcopy (cdr lista)(cdr listb))))))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;; INTERPOLAZIONE SEMPLICE TRA DUE O PIU' VALORI
;;; (ADDIZIONE DI VALORI)
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

(defun qinterp (list1 list2 list3)
  (if (null list1) NIL
      (if (null list2) NIL
          (if (null list3) NIL
              (append(jinterp (car list1)(car list2)(car
                list3)(car(cdr list1)))
                    (qinterp (cdr list1)(cdr list2)(cdr list3)))
              )))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(defun interplus (lfreq letapes lcoeff)
  (cons (car lfreq)
        ( qinterp lfreq letapes lcoeff)))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(defun qinterp0 (list1 rk qk)
  (if (null (cdr list1)) NIL
      (append(jinterp (car list1)rk qk (car(cdr list1)))
              (qinterp0 (cdr list1)rk qk)))
  ))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(defun interplus0 (list1 rk qk)

```



```

(cons (car list1)
      ( qinterp0 list1 rk qk)))

;;;;;;;;;;;;;
;;; INTERPOLAZIONE COMPLESSA TRA DUE LISTE DI VALORI
;;; (ADDIZIONE DI VALORI)
;;;;;;;;;;;;;

(defun casscopy (lista)
  (if (null lista)nil
      (cons
        (cdr (car lista))
        (casscopy(cdr lista))))))

;;;;;;;;;;;;;
(defun carcopy (lista)
  (if (null lista)nil
      (cons (car(car lista))
            (carcopy(cdr lista))))))

;;;;;;;;;;;;;
(defun popcopy (lista)
  (if (null (car lista))nil
      (cons
        (carcopy lista)
        (popcopy(casscopy lista))))))

;;;;;;;;;;;;;
(defun qinterpxx1 (list1 r q)
  (if (null list1)NIL
      (cons(jinterpol (car list1) r q (car(cdr list1)))
            (qinterpxx1 (cdr(cdr list1)) r q))
  ))

;;;;;;;;;;;;;
(defun qinterpxx1/2 (list1 r q)
  (if (null list1)NIL
      (cons(jinterpol (car list1) r q (car(cdr list1)))
            (qinterpxx1/2 (cdr(cdr list1)) r q))
  ))

;;;;;;;;;;;;;
(defun qinterpxx2 (list1 r list2)
  (if (null list1)NIL
      (if (null list2)NIL
          (cons(jinterpol (car list1) r (car list2)(car(cdr list1)))
                (qinterpxx2 (cdr(cdr list1)) r (cdr list2)))
          ))))

;;;;;;;;;;;;;
(defun interpmax0 (list1 list2 rk qk r q)
  (popcopy(qinterpxx1
            (scopy(interplus0 list1 rk qk)
                  (interplus0 list2 rk qk))
            r q)))

;;;;;;;;;;;;;
(defun interpmax1 (list1 list2 list3 list4 r q)
  (popcopy(qinterpxx1
            (scopy(interplus list1 list3 list4

```

```

        (interplus list2 list3 list4))
      r q)))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(defun interpmax2 (list1 list2 list3 list4 r list5)
  (popcopy(qinterpmax2
    (scopy(interplus list1 list3 list4)
      (interplus list2 list3 list4))
    r list5)))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;; INTERPOLAZIONE SEMPLICE TRA DUE O PIU' VALORI
;;; (DIFFERENZA DI VALORI)
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

(defun kqinterp (list1 list2 list3)
  (if (null list1)NIL
      (if (null list2)NIL
          (if (null list3)NIL
              (append(xjinterp (car list1)(car list2)(car
list3)(car(cdr list1)))
                (kqinterp (cdr list1)(cdr list2)(cdr list3)))
          )))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(defun kqinterp0 (list1 rk qk)
  (if (null(cdr list1))NIL
      (append(xjinterp (car list1)rk qk(car(cdr list1)))
        (kqinterp0 (cdr list1)rk qk))
  ))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;; INTERPOLAZIONE COMPLESSA TRA DUE LISTE DI VALORI
;;; (DIFFERENZA DI VALORI)
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

(defun kqinterpmax1 (list1 r q)
  (if (null list1)NIL
      (cons(xjinterp (car list1) r q (car(cdr list1)))
        (kqinterpmax1 (cdr(cdr list1)) r q))
  ))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(defun kqinterpmax1/2 (list1 r q)
  (if (null list1)NIL
      (cons(xjinterp (car list1) r q (car(cdr list1)))
        (kqinterpmax1/2 (cdr(cdr list1)) r q))
  ))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(defun kqinterpmax2 (list1 r list2)
  (if (null list1)NIL
      (if (null list2)NIL
          (cons(jinterpol (car list1) r (car list2)(car(cdr list1)))
            (kqinterpmax2 (cdr(cdr list1)) r (cdr list2)))
          )))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(defun kinterpmax0 (list1 list2 rk qk r q)
  (popcopy (qinterpmax1

```

```

      (scopy(kqinterp0 list1 rk qk)
        (kqinterp0 list2 rk qk))
      r q)))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(defun kinterpmax1 (list1 list2 list3 list4 r q)
  (popcopy (kqinterp0 list1 list3 list4)
    (scopy(kqinterp list1 list3 list4)
      (kqinterp list2 list3 list4))
    r q)))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(defun kinterpmax2 (list1 list2 list3 list4 r list5)
  (popcopy (kqinterp0 list1 list3 list4)
    (scopy(kqinterp list1 list3 list4)
      (kqinterp list2 list3 list4))
    r list5)))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;; INTERPOLAZIONE SEMPLICE VALORE-TEMPO
;;; (CREAZIONE DI UN INVILUPPO)
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

(defun interpline0 (k n q list1 r q1 list2 q2 nom-fichier )
  (let((listx(interplus0 list1 r q1)))
    (let((listy(kqinterp0 list2 r q2)))
      (if (null listx) nil
        (if (null listy) nil
          (append(list
            (format nom-fichier "~S~S ~S ~8F,"k n q(car listx))
            (format nom-fichier "~8F,"(car listy))
            (SScopy (cdr listx)(cdr listy) nom-fichier)
            (format nom-fichier "~8F ~%-%"(car(reverse listx)))
            ))))))))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(defun endinterpline0 (k n q list1 r q1 list2 q2 nom-fichier )
  (let((listx(interplus0 list1 r q1)))
    (let((listy(kqinterp0 list2 r q2)))
      (if (null listx) nil
        (if (null listy) nil
          (append(list
            (format nom-fichier "~S~S ~S ~8F,"k n q (car listx))
            (format nom-fichier "~8F,"(car listy))
            (SScopy (cdr listx)(cdr listy) nom-fichier)
            (format nom-fichier "~8F,.01,0.00001~%-%"(car(reverse
listx))))
          ))))))))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(defun interpline (k n q list1 list2 list3 list4 list5 nom-fichier )
  (let((listx(interplus list1 list2 list3)))
    (let((listy(kqinterp list4 list2 list5)))
      (if (null listx) nil
        (if (null listy) nil
          (append(list
            (format nom-fichier "~S~S ~S ~8F,"k n q (car listx))
            (format nom-fichier "~8F,"(car listy))
            (SScopy (cdr listx)(cdr listy) nom-fichier)
            (format nom-fichier "~8F ~%-%"(car(reverse listx)))
            ))))))))

```

```

))))))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;; INTERPOLAZIONE COMPLESSA VALORE-TEMPO
;;; (CREAZIONE DI DUE O PIU' INVILUPPI)
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

(defun kcopy (k list nom-fichier)
  (if (null list) nil
      (append (list (format nom-fichier "~S~S " k (car list))))))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(defun xxxcopy (k q Lista listb nom-fichier)
  (if (Null Lista) Nil
      (if (Null Listb) Nil
          (append (list
                    (kcopy k lista nom-fichier)
                    (cons (lhhcopy q (Car Listb) nom-fichier)
                          (xxxcopy k q (cdr lista) (Cdr Listb) nom-fichier)))))))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(defun endlhhcopy (q Lista nom-fichier)
  (let ((lista-1 (reverse (cdr (reverse lista)))))
    (if (Null Lista) Nil
        (append (list
                  (format nom-fichier " ~S ~8F," q (Car Lista-1))
                  (lcopy (Cdr Lista-1) nom-fichier)
                  (format nom-fichier "~8F,.01,0.00001-&" (car (reverse lista)))))))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(defun endxxxcopy (k q Lista listb nom-fichier)
  (if (Null Lista) Nil
      (if (Null Listb) Nil
          (append (list
                    (kcopy k lista nom-fichier)
                    (cons (endlhhcopy q (Car Listb) nom-fichier)
                          (endxxxcopy k q (cdr lista) (Cdr Listb) nom-fichier)))))))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(defun env0 (k n q list1 list2 rk qk1 r q1
             list6 list7 qk2 q2 nom-fichier)
  (let ((bbb
        (let ((lista (reverse (cdr (reverse (interpmax0 list1 list2 rk qk1 r
                                                         q1)))))
              (let ((listb (reverse (cdr (reverse (kinterpmax0 list6 list7 rk qk2 r
                                                                q2)))))
                    (if (null lista) nil
                        (if (null listb) nil
                            (cons
                             (wcopy (car lista) (car listb))
                             (zcopy (cdr lista) (cdr listb))
                             ))))))))
    (endxxxcopy k q (addiz n (spectrum 1 1 (+ r 2))) bbb nom-fichier)))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(defun env1 (k n q list1 list2 list3 list4 r q1
             list6 list7 list8 q2 nom-fichier)
  (let ((bbb
        (let ((lista (interpmax1 list1 list2 list3 list4 r q1))
              (listb (interpmax0 list6 list7 list8 q2 nom-fichier)))
          (if (null lista) nil
              (if (null listb) nil
                  (cons
                   (wcopy (car lista) (car listb))
                   (zcopy (cdr lista) (cdr listb))
                   ))))))))
    (endxxxcopy k q (addiz n (spectrum 1 1 (+ r 2))) bbb nom-fichier)))

```

```

      (let((listb (kinterpmax1 list6 list7 list3 list8 r q2)))
        (if (null lista) nil
            (if (null listb) nil
                (cons
                 (wcopy (car lista) (car listb))
                 (zcopy (cdr lista) (cdr listb))
                 ))))))
      (xxxcopy k q (addiz n (spectrum 1 1 (+ r 2))) bbb nom-fichier))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(defun env2 (k n q list1 list2 list3 list4 r list5
            list6 list7 list8 list9 nom-fichier)
  (let((bbb
        (let((lista(interpmax2 list1 list2 list3 list4 r list5)))
          (let((listb (kinterpmax2 list6 list7 list3 list8 r list9)))
            (if (null lista) nil
                (if (null listb) nil
                    (cons
                     (wcopy (car lista) (car listb))
                     (zcopy (cdr lista) (cdr listb))
                     ))))))))
    (xxxcopy k q (addiz n (spectrum 1 1 (+ r 2))) bbb nom-fichier))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(defun molt (lista x)
  (if (null lista) nil
      (cons(*(car lista) x)
            (molt (cdr lista) x))))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(defun superenv (k q list1 list2 list3 rk list4 list5 list6 list7 nom-
fichier)
  (if (null (cdr list3)) nil
      (list (env0 k (car list1) q (molt list2 (car list3)) (molt
list2(car(cdr list3)))
            rk (car list4) (- (car(cdr list1))(car list1) 1) (car list5)
            list6 list6 (car list7) 1 nom-fichier)
            (superenv k q (cdr list1) list2(cdr list3) rk (cdr list4)
            (cdr list5) list6 (cdr list7) nom-fichier)
            )))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(defun superenvelope (k q list1 list2 list3 rk list4 list5 list6 list7
nom-fichier)
  (list
   (superenv k q list1 list2 list3 rk list4 list5 list6 list7 nom-fichier)
   (interpline0 k
    (car(reverse list1)) q
    (molt list2 (car(reverse list3)))
    rk (car(reverse list4))
    list6 (car(reverse list7)) nom-fichier)))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(defun iperenv (k q list1 list2 list3 rk list4 list5
               list6 list7 list8 list9 nom-fichier)
  (if (null (cdr list3)) nil
      (list (env0 k (car list1) q (molt list2 (car list3)) (molt
list2(car(cdr list3)))
            rk (car list4) (- (car(cdr list1))(car list1) 1) (car list5)

```

```

      (molt list6 (car list7)) (molt list6 (car (cdr list7))) (car
list8) (car list9) nom-fichier)
      (iperenv k q (cdr list1) list2(cdr list3) rk (cdr list4)
      (cdr list5) list6 (cdr list7) (cdr list8) (cdr
list9) nom-fichier)
    )))

```

```

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(defun iperenvelope (k q list1 list2 list3 rk list4 list5
                    list6 list7 list8 list9 nom-fichier)
  (list
    ( iperenv k q list1 list2 list3 rk list4 list5 list6 list7 list8 list9
nom-fichier)
    (endinterpline0 k
      (car(reverse list1)) q
      (molt list2 (car(reverse list3)))
      rk (car(reverse list4))
      (molt list6 (car(reverse list7))) (car(reverse list8))
nom-fichier)))

```

```

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(defun line (k n q list1 list2 nom-fichier )
  (let((listx(interplus0 list1 0 1)))
    (let((listy(kqinterp0 list2 0 1)))
      (if (null listx) nil
          (if (null listy) nil
              (append(list
                (format nom-fichier "~%-S-S ~S ~8F," k n q (car listx))
                (format nom-fichier "~8F," (car listy))
                (SScopy (cdr listx)(cdr listy) nom-fichier)
                (format nom-fichier "~8F"(car(reverse listx))
                ))))))))

```

```

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(defun linend (k n q list1 list2 nom-fichier )
  (let((listx(interplus0 list1 0 1)))
    (let((listy(kqinterp0 list2 0 1)))
      (if (null listx) nil
          (if (null listy) nil
              (append(list
                (format nom-fichier "~%-S-S ~S ~8F," k n q (car listx))
                (format nom-fichier "~8F," (car listy))
                (SScopy (cdr listx)(cdr listy) nom-fichier)
                (format nom-fichier "~8F, .01,0.00001"(car(reverse
listx)))
                ))))))))

```

```

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(defun interpmin (list1 list2 r q)
  (popcopy (qinterpxx1
            (scopy(interplus0 list1 0 1)
                  (interplus0 list2 0 1))
            r q)))

```

```

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(defun kinterpmin (list1 list2 r q)
  (popcopy (qinterpxx1
            (scopy(kqinterp0 list1 0 1)
                  (kqinterp0 list2 0 1))
            r q)))

```

```

;;;;;;;;;;;;;
(defun qenv0 (k n q list1 list2 r q1
             list6 list7 q2 nom-fichier)
  (let((bbb
        (let((lista(reverse(cdr(reverse(interpmin list1 list2 r q1))))))
              (let((listb(reverse (cdr(reverse(kinterpmin list6 list7 r q2))))))
                    (if (null lista) nil
                        (if (null listb) nil
                            (cons
                             (wcopy (car lista) (car listb))
                             (zcopy (cdr lista) (cdr listb))
                             ))))))))
        (xxxcopy k q (addiz n (spectrum 1 1 (+ r 2))) bbb nom-fichier)))

;;;;;;;;;;;;;
(defun endqenv0 (k n q list1 list2 r q1
                list6 list7 q2 nom-fichier)
  (let((bbb
        (let((lista(reverse(cdr(reverse(interpmin list1 list2 r q1))))))
              (let((listb(reverse (cdr(reverse(kinterpmin list6 list7 r q2))))))
                    (if (null lista) nil
                        (if (null listb) nil
                            (cons
                             (wcopy (car lista) (car listb))
                             (zcopy (cdr lista) (cdr listb))
                             ))))))))
        (endxxxcopy k q (addiz n (spectrum 1 1 (+ r 2))) bbb nom-fichier)))

;;;;;;;;;;;;;
(defun superqenv (k q list1 list2 list3 list4 list5 nom-fichier)
  (if (null (cdr list3)) nil
      (list (qenv0 k (car list1) q (molt list2 (car list3)) (molt
list2(car(cdr list3)))
            (- (car(cdr list1))(car list1) 1) (car list4)
              list5 list5 1 nom-fichier)
            (superqenv k q (cdr list1 ) list2 (cdr list3)
              (cdr list4) list5 nom-fichier)
            )))

;;;;;;;;;;;;;
(defun superqenvelope (k q list1 list2 list3 list4 list5 nom-fichier)
  (list (format nom-fichier "~%")
        ( superqenv k q list1 list2 list3 list4 list5 nom-fichier)
        (interpline0 k
          (car(reverse list1)) q
            (molt list2 (car(reverse list3)))
              0 1
                list5 1 nom-fichier)))

;;;;;;;;;;;;;
(defun iperqenv (k q list1 list2 list3 list4 list5 list6 list7 nom-fichier)
  (if (null (cdr list3)) nil
      (list (endqenv0 k (car list1) q (molt list2 (car list3)) (molt
list2(car(cdr list3)))
            (- (car(cdr list1))(car list1) 1) (car list4)
              (molt list5 (car list6)) (molt list5 (car(cdr list6))) (car
list7) nom-fichier)
            (iperqenv k q (cdr list1 ) list2 (cdr list3)

```

```

                                (cdr list4) list5  (cdr list6) (cdr list7) nom-
fichier)
)))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(defun ipergenvelope (k q list1 list2 list3 list4 list5 list6 list7 nom-
fichier)
  (list
    ( ipergenv k q list1 list2 list3  list4 list5 list6 list7 nom-fichier)
    (endinterpline0 k
      (car(reverse list1)) q
      (molt list2 (car(reverse list3)))
      0 1
      (molt list5 (car(reverse list6)))
      (car(reverse list7)) nom-fichier)))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;; MOLTIPLICAZIONE-ADDIZIONE-SOTTRAZIONE DI LISTE
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

(defun chorus (lista listb)
  (if (null lista ) NIL
      (append(molt  listb (car lista))
              (chorus (cdr lista ) listb))))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(defun summa(n list)
  (if(null list)nil
      (cons (+ n (car list))
            (summa n (cdr list)))))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(defun sott(n list)
  (if(null list)nil
      (cons (- (car list) n)
            (sott n (cdr list)))))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(defun sumplus (lista listb)
  (if (null listb ) NIL
      (append(summa (car listb) lista)
              (sumplus lista (cdr listb)))))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(defun sotplus (lista listb)
  (if (null listb ) NIL
      (append(sott (car listb) lista)
              (sotplus lista (cdr listb)))))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;; INIZIO ORCHESTRA
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

(defun enter (x1 x2 x3 nom-fichier)
  (append
    (format nom-fichier " ~%sr = ~S" x1)
    (format nom-fichier " ~%kr = ~S" x2)
    (format nom-fichier " ~%ksmps = ~S" (/ x1 x2))
    (format nom-fichier " ~%chnls = ~S ~%-%-%" x3)))

```



```

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(defun ninstr (x1 x2 x3 nom-fichier)
  (append (format nom-fichier "~%~%instr ~S" x1 )
    (format nom-fichier " ~%iscaling=32767/(~S*~S)~%~%" x2 x3)))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;; ELENCO OSCILLATORI
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

(defun oscilix (lista listb nom-fichier)
  (if (null lista)nil
    (append
      (format nom-fichier "~%a~S oscili "(car lista))
      (format nom-fichier "kenv~S*iscaling,"(car lista))
      (format nom-fichier "~8F+kvib+kjitter,1"(car listb))
      (oscilix (cdr lista)(cdr listb)nom-fichier))))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(defun oscili (lista nom-fichier)
  (oscilix (spectrum 1 1 (length lista)) lista nom-fichier))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;; ELENCO OSCILLATORI CON P4 E P5
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

(defun osciliP (lista listb nom-fichier)
  (if (null lista)nil
    (append
      (format nom-fichier "~%a~S oscili "(car lista))
      (format nom-fichier "p5 * (kenv~S*iscaling),"(car lista))
      (format nom-fichier "(p4 * ~8F)+kvib+kjitter,1"(car listb))
      (osciliP (cdr lista)(cdr listb)nom-fichier))))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(defun PXoscili (lista nom-fichier)
  (oscilip (spectrum 1 1 (length lista)) lista nom-fichier))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;; FINE ORCHESTRA
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

(defun aout1 (lista y nom-fichier)
  (if (null lista)nil
    (append
      (format nom-fichier "+a~S" (+ (car lista) y))
      (aout1 (cdr lista) y nom-fichier))))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(defun acopy(x listb y nom-fichier)
  (if (null listb)nil
    (append (format nom-fichier "~%aout~S=0.0" x)
      (aout1 (spectrum 1 1 (car listb)) y nom-fichier)
      (acopy (+ 1 x) (cdr listb) (+ y (car listb)) nom-fichier))))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(defun exit (k lista nom-fichier)
  (if( null lista)nil
    (append
      (acopy 1 lista 0 nom-fichier)
      (format nom-fichier"~%~%asortiel=0.0"))
  )

```

```

      (dolist (x(spectrum 1 1 (length lista))nil)
        (format nom-fichier "+aout~S"x)
        (format nom-fichier "~%-S (0.0+asortiel)~%endin"k)))

;;;;;
;;; VIBRATO
;;;;;

(defun vibr (q list1 list2 list3 nom-fichier)
  (append
    (line 'kprof 1 q list1 list3 nom-fichier)
    (line 'kvit 1 q list2 list3 nom-fichier)
    (format nom-fichier "kvib oscili kprof1,kvit1,1~%-%" )))

;;;;;
(defun vibrlea (q x list1 y list2 list3 list4 nom-fichier)
  (append
    (format nom-fichier "~%kalea1 randi ~S,~S~%"(nth 0 list4)(nth 1 list4))
    (format nom-fichier"kalea2 randi ~S,~S~%"(nth 2 list4)(nth 3 list4))
    (line 'kprof 1 q (molt list1 x)list3 nom-fichier)
    (line 'kvit 1 q (molt list2 y)list3 nom-fichier)
    (format
      nom-fichier
      "~%kvib
      kprof1+(kprof1*kalea1),kvit1+(kvit1*kalea2),1~%-%" )))

;;;;;
(defun vibr+ (n nom-fichier)
  (append
    (dolist (x(spectrum 1 1 n)nil)
      (format nom-fichier "~%kvib~S oscili "x)
      (format nom-fichier "kprof~S,"x)
      (format nom-fichier "kvit~S,~1~%"x)
    )))

;;;;;
;;; JITTER
;;;;;

(defun jitter (q list1 list2 list3 nom-fichier)
  (append
    (line 'khalbfd 1 q list1 list3 nom-fichier)
    (line 'ktimes 1 q list2 list3 nom-fichier)
    (format nom-fichier "kjitter randi khalbfd1,ktimes1~%" )))

;;;;;
(defun jitterplus(x list1 nom-fichier)
  (if (null list1)nil
    (append
      (format nom-fichier "kjitter~S randi ~S,~S~%"x (nth 0 list1)(nth 1
list1))
      (jitterplus (+ x 1) (cdr(cdr list1)) nom-fichier))))

;;;;;
(defun jitterplus ( list1 nom-fichier)
  (append
    (jitterplus 1 list1 nom-fichier)
    (format nom-fichier "kjitter=0.0")
    (dolist (x(spectrum 1 1 (/(length list1)2)) nil)
      (format nom-fichier "+kjitter~S" x ))
    (format nom-fichier "~%" )))

```

```

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(defun jitter+ (n nom-fichier)
  (append
    (dolist (x(spectrum 1 1 n)nil)
      (format nom-fichier "~%kjitter~S randi"x)
      (format nom-fichier "khalfdb~S,"x)
      (format nom-fichier "ktimes~S~%"x)
    )))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;; BRUIT ,FONTI PERIODICHE E FILTRI
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

(defun abruit (q list1 list2 list3 nom-fichier)
  (append
    (line 'khalfdb 1 q list1 list3 nom-fichier)
    (line 'ktimes 1 q list2 list3 nom-fichier)
    (format nom-fichier "~%aimpuls randi khalfdb1,ktimes1~%")))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(defun aperiod (q list1 list2 f nom-fichier)
  (if(null list1)NIL
    (append (line 'kenv 1 q list1 list2 nom-fichier)
      (format nom-fichier "~%aimpuls oscili kenv1,~8F,2~%" f))))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(defun resonx (lista listb listc nom-fichier)
  (if (null lista)nil
    (append
      (format nom-fichier "~%a~S reson "(car lista))
      (format nom-fichier "(aimpuls*~8F)*iscaling,~8F,"(car listb)(car
listc))
      (format nom-fichier "kband~S+(kband~S*kbalea)"(car lista)(car
lista))
      (resonx (cdr lista)(cdr listb)(cdr listc)nom-fichier))))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(defun reson (lista listb listc nom-fichier)
  (append
    (format nom-fichier "~%kbalea randi ~S,~S~%"(nth 0 listc)(nth 1 listc))
    (resonx (spectrum 1 1 (length lista)) lista listb nom-fichier)))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(defun x*1 (lista listb)
  (if (null lista ) NIL
    (cons(molt listb (car lista))
      (x*1 (cdr lista ) listb))))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;; FORMANTI PER SINTESI ADDITIVA
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

(defun formantx (lista listb listc nom-fichier)
  (if (null lista)nil
    (append
      (format nom-fichier "~%aformant~S reson "(car lista))
      (format nom-fichier "(asortiel*~8F)*iscaling,~8F,"(car listb)(car
listc))
      (format nom-fichier "kband~S+(kband~S*kbalea)"(car lista)(car
lista))
    )))

```

```

(formantx (cdr lista)(cdr listb)(cdr listc)nom-fichier)))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(defun formant (lista listb listc nom-fichier)
  (append
    (format nom-fichier "~%kbalea randi ~S,~S~%"(nth 0 listc)(nth 1 listc))
    (formantx (spectrum 1 1 (length lista)) lista listb nom-fichier)))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(defun kformantx (lista listb nom-fichier)
  (if (null lista)nil
    (append
      (format nom-fichier "~%aformant~S reson "(car lista))
      (format nom-fichier "(asortiel*~8F)*iscaling,kgliss~S,"(car
listb)(car lista))
      (format nom-fichier "kband~S+(kband~S*kbalea)"(car lista)(car
lista))
      (kformantx (cdr lista)(cdr listb)nom-fichier))))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(defun kformant (lista listb nom-fichier)
  (append
    (format nom-fichier "~%kbalea randi ~S,~S~%"(nth 0 listb)(nth 1 listb))
    (kformantx (spectrum 1 1 (length lista)) lista nom-fichier)))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(defun choose (x lista)
  (if(null lista) nil
    (cons(nth x(car lista))
      (choose x (cdr lista)))))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(defun kbandx (k x q listn lista listb nom-fichier)
  (if (null listn) nil
    (append
      (line k (car listn) q (choose x lista) listb nom-fichier)
      (kbandx k (+ 1 x) q (cdr listn) lista listb nom-fichier)
      )))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(defun kelenc (q lista listb listc nom-fichier)
  (if (null lista)nil
    (append
      (kbandx 'kgliss 0 q (spectrum 1 1 (length (car lista))) lista listc
nom-fichier)
      (kbandx 'kband 0 q (spectrum 1 1 (length (car lista))) listb listc
nom-fichier))))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;; RM
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

(defun RMoscilix (lista listb nom-fichier)
  (if (null lista)nil
    (append
      (format nom-fichier "~%a~S oscili "(car lista))
      (format nom-fichier "a0+(kenvl*iscaling),~7F+kvib+kjitter,1"(car
listb))
      (RMoscilix (cdr lista)(cdr listb)nom-fichier))))

```

```

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(defun RMoscili (lista nom-fichier)
  (rmoscilix (spectrum 1 1 (length lista)) lista nom-fichier))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(defun RM (lista listb listc nom-fichier)
  (if (null lista)NIL
      (append
       (line 'kgliiss 1 listb listc nom-fichier)
       (format nom-fichier "a0 oscili kenv1*iscaling,kgliiss1,1")
       (RMoscili lista nom-fichier))))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(defun modamx (lista listb listc nom-fichier)
  (if (null lista)nil
      (append
       (format nom-fichier "~%am~S oscili "(car lista))
       (format nom-fichier "kenv0,"(car lista))
       (format nom-fichier "~7F,1"(-(* (car listb)(car listc))(car
listb))))
      (modamx(cdr lista)(cdr listb)(cdr listc)nom-fichier)))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(defun modam (lista listb nom-fichier)
  (modamx (spectrum 1 1 (length lista)) lista listb nom-fichier))
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(defun am (lista listb nom-fichier)
  (append (oscili lista nom-fichier)
          (modam lista listb nom-fichier)))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;; FINE ORCHESTRA AM
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

(defun amout1 (lista y nom-fichier)
  (if (null lista)nil
      (append
       (format nom-fichier "+a-S+(a-S*am-S)" (+ (car lista) y)(+ (car
lista)y)(+ (car lista)y))
       (amout1 (cdr lista) y nom-fichier))))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(defun amcopy(x listb y nom-fichier)
  (if (null listb)nil
      (append (format nom-fichier "~%aout~S=0.0" x)
              (amout1 (spectrum 1 1 (car listb)) y nom-fichier)
              (amcopy (+ 1 x) (cdr listb) (+ y (car listb)) nom-fichier))))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(defun amexit (k lista nom-fichier)
  (if( null lista)nil
      (append
       (amcopy 1 lista 0 nom-fichier)
       (format nom-fichier"~%~%asortiel=0.0")
       (dolist (x(spectrum 1 1 (length lista))nil)
         (format nom-fichier "+aout~S"x))
       (format nom-fichier "~%~S (0.0+asortiel)~%endin"k))))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;; FM

```

```

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(defun foscilix (lista listb nom-fichier)
  (if (null lista) nil
      (append
        (format nom-fichier "~%a-S foscili "(car lista))
        (format nom-fichier "kenv~S*iscaling,"(car lista))
        (format nom-fichier "1,kport~S,kport~S*~8F,kindex1,1"
          (car lista)(car lista)(car listb))
        (foscilix (cdr lista)(cdr listb) nom-fichier))))
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(defun foscili (lista nom-fichier)
  (foscilix (spectrum 1 1 (length lista)) lista nom-fichier))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;; FORMANTI E FINE ORCHESTRA PER SINT ADD + FOMANTI
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

(defun formantexit (lista listb listc k listout nf nom-fichier)
  (if (null lista) nil
      (append
        (acopy 1 listout 0 nom-fichier)
        (format nom-fichier "~%-%asortiel=0.0")
        (dolist (x (spectrum 1 1 (length listout))) nil)
        (format nom-fichier "+aout~S"x))
        (formant lista listb listc nom-fichier)
        (format nom-fichier "~%-%asortie2=0.0")
        (dolist (x (spectrum 1 1 nf)) nil)
        (format nom-fichier "+aformant~S"x))
        (format nom-fichier "~%-S (0.0+asortiel+asortie2)~%endin"k))))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(defun kformantexit (lista listb k listout nf nom-fichier)
  (if (null lista) nil
      (append
        (acopy 1 listout 0 nom-fichier)
        (format nom-fichier "~%-%asortiel=0.0")
        (dolist (x (spectrum 1 1 (length listout))) nil)
        (format nom-fichier "+aout~S"x))
        (kformant lista listb nom-fichier)
        (format nom-fichier "~%-%asortie2=0.0")
        (dolist (x (spectrum 1 1 nf)) nil)
        (format nom-fichier "+aformant~S"x))
        (format nom-fichier "~%-S (0.0+asortiel+asortie2)~%endin"k))))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;; DUPLICAZIONE DI LISTE
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

(defun du (lista)
  (if (null lista) nil
      (append (list (car lista) (car lista))
              (du (cdr lista)))))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;; RIPETIZIONE DELLO STESSO VALORE
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

(defun idem (x ntimes)
  (jinterpol x (- ntimes 2) 1 x))

```

```

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;; DEFINIZIONE SCORE COMPLESSO
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

(defun scoresa (x lista nom-fichier)
  (if (null lista) nil
      (append
       (format nom-fichier " ~S" (nth x (car lista)))
       (scoresa x (cdr lista) nom-fichier))))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(defun scora (x n lista nom-fichier)
  (append
   (format nom-fichier "~%i~S"n)
   (scoresa x lista nom-fichier)))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(defun scoreson ( j n lista nom-fichier)
  (if (> j (-(length (car lista))1)) nil
      (append
       (scora j n lista nom-fichier)
       (scoreson (+ 1 j) n lista nom-fichier))))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(defun sco (n lista nom-fichier)
  (append (scoreson 0 n lista nom-fichier)
          (format nom-fichier "~%e")))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;; DEFINIZIONE ENVELOPE ORC PER SCORE COMPLESSO
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

(defun pSScopy (list1 list2 nom-fichier)
  (if (null list1) NIL
      (if (null list2) NIL
          (append(list(format nom-fichier "~8F*p4,"(car list1))
                      (format nom-fichier "~8F*p5,"(car list2))
                      (pSScopy (cdr list1)(cdr list2) nom-fichier))))))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(defun pline (k n q list1 list2 nom-fichier )
  (let((listx(interplus0 list1 0 1))
        (listy(kqinterp0 list2 0 1)))
    (if (null listx) nil
        (if (null listy) nil
            (append(list
                     (format nom-fichier "~%-S-S ~S ~8F*p4,"k n q (car
listx))
                     (format nom-fichier "~8F*p5,"(car listy))
                     (pSScopy (cdr listx)(cdr listy) nom-fichier)
                     (format nom-fichier "~8F*p4"(car(reverse listx))
                     ))))))))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(defun plinend (k n q list1 list2 nom-fichier )
  (let((listx(interplus0 list1 0 1))
        (listy(kqinterp0 list2 0 1)))
    (if (null listx) nil
        (if (null listy) nil
            (append(list
                     (format nom-fichier "~%-S-S ~S ~8F*p4,"k n q (car
listx))
                     (format nom-fichier "~8F*p5,"(car listy))
                     (pSScopy (cdr listx)(cdr listy) nom-fichier)
                     (format nom-fichier "~8F*p4"(car(reverse listx))
                     ))))))))

```

```

      (append(list
listx))      (format nom-fichier "~%-S-S ~S ~8F*p4,"k n q (car
              (format nom-fichier "~8F*p5,"(car listy))
              (SScopy (cdr listx)(cdr listy) nom-fichier)
              (format nom-fichier "~8F*p4, .01,0.00001"(car(reverse
listx)))
              ))))

;;;;;;;;;;;;;
(defun plinex (k n q p1 p2 list1 list2 nom-fichier )
  (let((listx(interplus0 list1 0 1))
        (listy(kqinterp0 list2 0 1))
        (if (null listx) nil
              (if (null listy) nil
                    (append(list
listx)p1)      (format nom-fichier "~%-S-S ~S ~8F*p-S,"k n q (car
                  (format nom-fichier "~8F*p-S,"(car listy) p2)
                  (pSScopy (cdr listx)(cdr listy) nom-fichier)
                  (format nom-fichier "~8F*p-S"(car(reverse listx))p1)
                  ))))

;;;;;;;;;;;;;
;ESEMPIO
;;;;;;;;;;;;;

;(with-open-file (nom-fichier "Cursus FX1:Users:FR:getStarted:7sco.sco"
;                          :direction :output :if-exists :overwrite
;                          :if-does-not-exist :create)

;;;;;;;;;;;;;
;(SCO
; ;;-P1: numero instr
;   1 '(
; ;;-P2: attacco
;   (jinterpol 0 7 1.5 1)
; ;;-P3: durata
;   (idem 12 9)
; ;;-P4: ratio amp
;   (jinterpol 1 7 1 .1)
; ;;-P5: ratio time
;   (jinterpol 1 7 1 .6)
; ;;-evaluate.
; )nom-fichier))

;;;;;;;;;;;;;
;;;FREQ E BAND VOCALI MASCHILI
;;;;;;;;;;;;;

(setq fa1 '(609 1000 2450 2700))
(setq fe '(400 1700 2300 2900))
(setq fi '(238 1741 2450 2900))
(setq fo '(325 700 2550 2850))
(setq fu '(360 750 2400 2675))
(setq foe '(415 1400 2200 2800))
(setq fa3 '(300 1600 2150 2700))
(setq fa2 '(400 1050 2200 2650))
(setq ba1 '(78 88 123 128))
(setq be '(64 81 101 119))

```



```

(setq bi '(73 108 123 132))
(setq bo '(73 80 125 131))
(setq bu '(51 61 168 184))
(setq boe '(45 64 93 114))
(setq ba3 '(66 93 108 122))
(setq ba2 '(73 90 118 127))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(defun sumplusx (lista listb)
  (if (null listb) nil
      (cons (summa (car listb) lista)
            (sumplusx lista (cdr listb)))))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(defun sumplusx2 (lista listb)
  (if (null listb) nil
      (cons (summa (car lista) (car listb))
            (sumplusx2 (cdr lista) (cdr listb)))))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(PW::defun score ((file list(:value '() :type-list nil))
                 (gen list(:value '() :type-list nil))
                 (lista list(:value '() :type-list nil)))
  nil "" (with-open-file (nom-fichier file
                          :direction :output :if-exists :overwrite
                          :if-does-not-exist :create)

(list
  (format nom-fichier GEN)
  (SCO
  1 lista nom-fichier))))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;; DISTORSIONI
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

(defun dist(part freq dist)
  (let ((res))
    (dotimes (n part res)
      (setf res (cons (* freq (expt(+ n 1) dist))res)))
    (reverse res)))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;; LISTA DI SPETTRI DIST SULLO STESSO FOND E COEFF. DIFFERENTI
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

(defun distplus (part freq lista)
  (if (null lista)nil
      (cons (dist part freq (car lista))
            (distplus part freq (cdr lista)))))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;; LISTA DI SPETTRI DIST SU DIFFERENTI FOND. E STESSO COEFF.
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

(defun distcord (p list0 dist)
  (if (null list0)nil
      (cons (dist p (car list0) dist)
            (distcord p (cdr list0) dist))))

```

```

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;; LISTA DI SPETTRI DIST SU DIFFERENTI FOND E DIFFERENTI COEFF.
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

(defun distcord2 (p list0 list1)
  (if (null list0) nil
      (cons (dist p (car list0) (car list1))
            (distcord2 p (cdr list0) (cdr list1)))))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;; A PARTIRE DA UNA LISTA DI FREQ E DI FOND , ESCE UNA LISTA
;;; DI SPETTRI DIST NEI QUALI LE FREQ CORRISPONDONO AI
;;; PARZIALI N (LPART) DEGLI SPETTRI DIST SULLE FOND DATE
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

(defun distcordfond (np lfz lfond lpart)
  (if (null lfz) nil
      (cons (dist np (car lfond)
                  (finddist (car lfz) (car lfond) (car lpart)))
            (distcordfond np (cdr lfz) (cdr lfond) (cdr lpart)))))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(defun distcord2*2 (p list0 list1 list2 list3)
  (if (null list0) nil
      (cons (append (dist p (car list0) (car list1))
                    (dist p (car list2) (car list3)))
            (distcord2*2 p (cdr list0) (cdr list1) (cdr list2) (cdr
list3)))))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;; TROVA IL COEFFICIENTE DI DISTORSIONE
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

(defun finddist (freq fond npart)
  (log (/ freq fond) npart))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;; DATO UN ACCORDO, LO INTEGRA IN SPETTRI DISTORTI
;;; DI UN FONDAMENTALE DATO ABBASSANDO IL COEFF.DIST.
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

(defun goback (np fond coeff)
  (if (< coeff .1) nil
      (cons (dist np fond coeff)
            (goback np fond (- coeff .05)))))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(defun brr (i lista x)
  (if (null lista) nil
      (if (and (< i (* (car lista) 1.059463))
               (> i (* (car lista) 0.94388636))) x
          (brr i (cdr lista) x))))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(defun brr2 (listi lista x)
  (if (null listi) nil
      (let ((qq (brr (car listi) lista x)))
        (if (not (eql qq x)) nil
            (cons x (brr2 (cdr listi) lista x))))))

```

```

;;;;;;;;;;;;;
(defun gogo (listi lista x)
  (nth (- (length listi) 1) (brr2 listi lista x)))

;;;;;;;;;;;;;
(defun findcoeff (listi lista listx)
  (if (null lista) nil
      (cons
        (gogo listi (car lista) (car listx))
        (findcoeff listi (cdr lista) (cdr listx)))))

;;;;;;;;;;;;;
(defun HH ( listi np fond coeff )
  (remove nil (findcoeff listi (goback np fond coeff)
                          (goback np fond coeff))))

;;;;;;;;;;;;;
;;; DATO UN ACCORDO, LO INTEGRA IN SPETTRI DISTORTI
;;; DI UN COEFF.DIST. DATO ABBASSANDO IL FONDAMENTALE
;;;;;;;;;;;;;

(defun gobackfond (np fond coeff)
  (if (< fond 27.5) nil
      (cons (dist np fond coeff)
            (gobackfond np (* fond .94388) coeff))))

;;;;;;;;;;;;;
(defun HHfond ( listi np fond coeff )
  (remove nil (findcoeff listi (gobackfond np fond coeff)
                          (gobackfond np fond coeff))))

;;;;;;;;;;;;;
;;; DATA UNA LISTA D'ACCORDI, LI INTEGRA IN SPETTRI DISTORTI
;;; DI UN FONDAMENTALE DATO ABBASSANDO IL COEFF.DIST.
;;;;;;;;;;;;;

(defun carhh (listi np fond coeff)
  (car (hh listi np fond coeff)))

;;;;;;;;;;;;;
(defun multihh (lista np fond coeff)
  (if (null lista) nil
      (cons (carhh (car lista) np fond coeff)
            (multihh (cdr lista) np fond coeff))))

;;;;;;;;;;;;;
;;; DATA UNA LISTA D'ACCORDI, LI INTEGRA IN SPETTRI DISTORTI
;;; DI UN FONDAMENTALE DATO ABBASSANDO IL FONDAMENTALE
;;;;;;;;;;;;;

(defun carhhfond(listi np fond coeff)
  (car (hhfond listi np fond coeff)))

;;;;;;;;;;;;;
(defun multihhfond (lista np fond coeff)
  (if (null lista) nil
      (cons (carhhfond (car lista) np fond coeff)
            (multihhfond (cdr lista) np fond coeff))))

```

```

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;; DATA UNA LISTA DI ACCORDI, LI INTEGRA IN SPETTRI DISTORTI
;;; DI UN FONDAMENTALE CORRISPONDENTE ALLA NOTA PIU' GRAVE DI
;;; OGNI ACCORDO, ABBASSANDO IL COEFF.DIST.
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

(defun contrhh(lista np coeff)
  (if (null lista) nil
      (cons (carhh (car lista) np (first (car lista)) coeff)
            (contrhh (cdr lista) np coeff))))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(defun contr-spectr (lista np coeff)
  (append lista (contrhh lista np coeff)))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;; TROVA IL COEFF DI DISTORSIONE
;;; DI UNA LISTA D'ACCORDI (PER UN FONDAMENTALE DATO)
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

(defun finddistchord (chord fond )
  (log (/ (nth 1 chord) fond) 2))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(defun findmultidist (lista fond )
  (if (null lista) nil
      (cons (finddistchord (car lista) fond)
            (findmultidist (cdr lista) fond))))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(defun finddist-contr(lista)
  (if (null lista) nil
      (cons (finddistchord (car lista) (first (car lista)))
            (finddist-contr (cdr lista)))))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;; FORNISCE UNA LISTA DI SPETTRI D'INTERPOLAZIONE TRA UNA
;;; LISTA DI SPETTRI DATA
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

(defun listcar (lista)
  (if (null lista) nil
      (cons (car (car lista))
            (listcar (cdr lista)))))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(defun interp-contr (Lspectr etapes coeff)
  (distcord2 (length (car Lspectr))
             (interplus0 (listcar Lspectr) etapes coeff )
             (interplus0 (finddist-contr Lspectr) etapes coeff)))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;; MODULAZIONE DI FREQUENZA
;;; ADDIZIONE
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

(defun fmspectrum1 (port mod index)
  (let ((l nil))
    (do ((i index (1- i))
        ((<= i 0) l)
        (push (fmspectrum1 port mod i) l))))

```

```

      (setq l (cons (+ port (* mod i))l))))))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;; SOTTRAZIONE
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

(defun fmspectrum2 (port mod index)
  (let ((l nil))
    (do ((i index (1- i))
        ((=> i 0)l)
        (setq l (cons (- port (* mod i))l))))))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;; RENDE POSITIVI I VALORI NEGATIVI DI UNA LISTA
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

(defun rempl (lista)
  (if (null lista) nil
      (cons
       (if (> 0 (car lista))
           (* (car lista) -1)
           (car lista))
       (rempl (cdr lista)))))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;; ADD.+SOTTR.con REMPL.(formula base)
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

(defun fm (port mod index)
  (append (fmspectrum1 port mod index)
          (remove 0 (rempl (fmspectrum2 port mod index)))))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;; FM con ratio
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

(defun fmr (port ratio index)
  (append (fmspectrum1 port (* port ratio) index)
          (remove 0 (rempl (fmspectrum2 port (* port ratio) index)))))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;; DATO UN ACCORDO, LO INTEGRA IN SPETTRI FM
;;; CON PORT FISSA E ABBASSANDO IL RATIO
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

(defun gobackratio (port ratio index)
  (if (< ratio .1) nil
      (cons (fmr port ratio index)
            (gobackratio port (- ratio .05) index))))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(defun HHratio (listi port ratio index)
  (remove nil (findcoeff listi (gobackratio port ratio index)
                        (gobackratio port ratio index))))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;; TROVA IL RATIO FM
;;; DI UNA LISTA DI SPETTRI FM (PER UNA PORTANTE DATA)
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

```

```

(defun findratio (lista port)
  (/ (- (car lista) port) port))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(defun findmultiratio (lista port)
  (if (null lista) nil
      (cons (findratio (car lista) port)
            (findmultiratio (cdr lista) port))))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;; DATO UN ACCORDO, LO INTEGRA IN SPETTRI FM
;;; CON RATIO FISSO E ABBASSANDO LA PORT
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

(defun gobackport (port ratio index)
  (if (< port 27.5) nil
      (cons (fmr port ratio index)
            (gobackport (* port .94388) ratio index))))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(defun HHport (listi port ratio index)
  (remove nil (findcoeff listi (gobackport port ratio index)
                          (gobackport port ratio index))))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;; TROVA LA PORT
;;; DI UNA LISTA DI SPETTRI
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

(defun findport (lista)
  (- (car lista) (/ (- (car lista) (nth 1 lista)) -1)))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(defun findmulport (lista)
  (if (null lista) nil
      (cons (findport (car lista))
            (findmulport (cdr lista)))))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;; DATA UNA LISTA D'ACCORDI, LI INTEGRA IN SPETTRI FM
;;; CON PORT FISSA E ABBASSANDO IL RATIO
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

(defun carhhratio (listi port ratio index)
  (car (hhratio listi port ratio index)))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(defun multihhratio (lista port ratio index)
  (if (null lista) nil
      (cons (carhhratio (car lista) port ratio index)
            (multihhratio (cdr lista) port ratio index))))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;; DATA UNA LISTA D'ACCORDI, LI INTEGRA IN SPETTRI FM
;;; CON RATIO FISSO E ABBASSANDO LA PORT
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

(defun carhhport (listi port ratio index)
  (car (hhport listi port ratio index)))

```

```

;;;;;;;;;;;;;
(defun multihport (lista port ratio index)
  (if (null lista) nil
      (cons (carhport(car lista) port ratio index)
            (multihport (cdr lista) port ratio index))))

;;;;;;;;;;;;;
;;; DATA UNA LISTA DI ACCORDI, LI INTEGRA IN SPETTRI FM
;;; DI UNA PORT CORRISPONDENTE ALLA NOTA X(NPORT) DI
;;; OGNI ACCORDO, ABBASSANDO IL RATIO
;;;;;;;;;;;;;

(defun contrhhratio (lista nport ratio index)
  (if (null lista) nil
      (cons (carhhratio (car lista) (nth nport (car lista)) ratio index )
            (contrhhratio (cdr lista) nport ratio index))))

;;;;;;;;;;;;;
(defun contr-fm (lista nport ratio index)
  (append lista (contrhhratio lista nport ratio index)))

;;;;;;;;;;;;;
;;; DATO UN ACCORDO, LO INTEGRA IN SPETTRI FM
;;; CON PORT FISSA E ABBASSANDO IL RATIO1 FINO AL RATIO2
;;;;;;;;;;;;;

(defun gobackratio2 (port ratio1 ratio2 index)
  (if (< ratio1 ratio2) nil
      (cons (fmr port ratio1 index)
            (gobackratio2 port (- ratio1 .03) ratio2 index))))

;;;;;;;;;;;;;
(defun HHratio2 ( listi port ratio1 ratio2 index )
  (remove nil (findcoeff listi (gobackratio2 port ratio1 ratio2 index)
                          (gobackratio2 port ratio1 ratio2 index))))

;;;;;;;;;;;;;
;;; DATA UNA LISTA DI INTERVALLI, LI INTEGRA IN SPETTRI DISTORTI
;;; ABBASSANDO IL COEFF.; E' POSSIBILE DEFINIRE QUANTI DEGLI INTERVALLI
;;; DELLA LISTA DEVONO ESSERE OBBLIGATORIAMENTE PRESENTI NELLO
;;; SPETTRO (NI) E IN QUALE DENSITA' MINIMA (NN ) RISPETTO
;;; AL NUMERO COMPLESSIVO DEGLI INTERVALLI DELLO SPETTRO
;;;;;;;;;;;;;

(defun popo (i lista x)
  (if (null (cdr lista)) nil
      (if (and (< i (* (/ (nth 1 lista) (nth 0 lista)) 1.0293022))
             (> i (* (/ (nth 1 lista) (nth 0 lista)) .97153834)))
          (cons x (popo i (cdr lista) x ))
          (cons nil (popo i (cdr lista) x )))))

;;;;;;;;;;;;;
(defun popo2 (listi lista )
  (if (null listi) nil
      (cons
        (if
          (< (length (remove nil (popo (car listi) lista 'ok))) 1) nil
          lista)
        (popo2 (cdr listi) lista ))))

```

```

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(defun popo2bis (listi lista )
  (if (null listi) nil
      (append (popo (car listi) lista 'ok)
              (popo2bis (cdr listi) lista ))))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(defun popo3/a (listi lista ni)
  (let (( gg (popo2 listi lista ))
        (if (<= ni (length (remove nil gg)))
            lista
            nil)))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(defun popo3/b (listi lista ni)
  (if (null lista) nil
      (remove nil (cons
                   (popo3/a listi (car lista) ni)
                   (popo3/b listi (cdr lista) ni)))))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(defun YY1 ( listi ni np fond coeff )
  (remove nil (popo3/b listi (goback np fond coeff)
                          ni)))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(defun popo3 (listi lista ni)
  (let (( gg (popo2bis listi lista ))
        (if (<= ni (length (remove nil gg)))
            lista
            nil)))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(defun popo4 (listi lista ni)
  (if (null lista) nil
      (cons
       (popo3 listi (car lista) ni)
       (popo4 listi (cdr lista) ni))))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(defun YY2 (listi ni nn np fond coeff)
  (remove nil (if (null (YY1 listi ni np fond coeff)) nil
                 (popo4 listi (YY1 listi ni np fond coeff) nn))))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;; DATA UNA LISTA DI INTERVALLI, LI INTEGRA IN SPETTRI FM
;;; ABBASSANDO IL RATIO; E' POSSIBILE DEFINIRE QUANTI DEGLI INTERVALLI
;;; DELLA LISTA DEVONO ESSERE OBBLIGATORIAMENTE PRESENTI NELLO
;;; SPETTRO (NI ) E IN QUALE DENSITA' MINIMA (NN )
;;; RISPETTO AL NUMERO COMPLESSIVO
;;;DEGLI INTERVALLI DELLO SPETTRO (INDEX * 2)
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

(defun fmr/sort (port ratio index)
  (sort (append (cons port (fmspectrum1 port (* port ratio) index))
              (remove 0 (repl (fmspectrum2 port (* port ratio) index))) '<))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(defun gobackratio/sort (port ratio index)

```



```

(if (< ratio .1) nil
    (cons (fmr/sort port ratio index)
          (gobackratio/sort port (- ratio .01) index)))

;;;;;;;;;;;;;
(defun YY1ratio( listi ni port ratio index )
  (remove nil (popo3/b listi (gobackratio/sort port ratio index)
                        ni)))

;;;;;;;;;;;;;
(defun YY2ratio (listi ni nn port ratio index)
  (remove nil (if (null (YY1ratio listi ni port ratio index)) nil
                  (popo4 listi (YY1ratio listi ni port ratio index) nn))))

;;;;;;;;;;;;;
(setq 2mi '1.059)
(setq 2ma '1.122)
(setq 3mi '1.189)
(setq 3ma '1.260)
(setq 4j  '1.335)
(setq 4ex '1.414)
(setq 5j  '1.498)
(setq 6mi '1.587)
(setq 6ma '1.682)
(setq 7mi '1.782)
(setq 7ma '1.888)
(setq 8j  '2)
(setq 9mi '2.119)
(setq 9ma '2.245)
(setq 10mi '2.378)

;;;;;;;;;;;;;
;;; DATA UNA LISTA DI LISTE D'INTERVALLI, LI INTEGRA IN
;;; SPETTRI DIST ABBASSANDO IL COEFF; E' POSSIBILE DEFINIRE
;;; QUANTI DEGLI INTERVALLI DELLA LISTA DEVONO ESSERE
;;; OBBLIGATORIAMENTE PRESENTI NELLO SPETTRO (NI)
;;; E IN QUALE DENSITA' MINIMA (NN ) RISPETTO AL NUMERO
;;; COMPLESSIVO DEGLI INTERVALLI DELLO SPETTRO (NP)
;;;;;;;;;;;;;

(defun caryy2 (listi ni nn np fond coeff)
  (car (yy2 listi ni nn np fond coeff)))

;;;;;;;;;;;;;
(defun multiyy2 (lista ni nn np fond coeff)
  (if (null lista) nil
      (cons (caryy2 (car lista) ni nn np fond coeff)
            (multiyy2 (cdr lista) ni nn np fond coeff))))

;;;;;;;;;;;;;
;;; CALCOLA L'INTERPOLAZIONE TRA DUE O PIU' SPETTRI DIST
;;; IN BASE AD UNA LISTA D'INTERVALLI.
;;; E' POSSIBILE DEFINIRE QUANTI DEGLI INTERVALLI
;;; DELLA LISTA DEVONO ESSERE OBBLIGATORIAMENTE PRESENTI
;;; NELLO SPETTRO (NI) E IN QUALE DENSITA' MINIMA (NN )
;;; RISPETTO AL NUMERO COMPLESSIVO DEGLI INTERVALLI DELLO
;;; SPETTRO, E QUANTE TAPPE INTERMEDIARIE DELL'INT
;;; DEVONO RISPONDERE A QUESTE CARATTERISTICHE AFFINCHE'
;;; L'INTERP VENGA EDITA (gogackplus + superxxx)
;;;;;;;;;;;;;

```

```

(defun gobackplus (Lspectr etapes coeff)
  (if (< coeff .1) nil
      (cons (interp-contr Lspectr etapes coeff)
            (gobackplus Lspectr etapes (- coeff .05)))))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(defun xxx (listi listone ni nn )
  (let ((ww (remove nil (popo3/b listi (car listone) ni))))
    (remove nil (if (null ww) nil
                    (popo4 listi ww nn)))))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(defun selectxxx (listi listone ni nn nr)
  (let ((qqq ( xxx listi listone ni nn)))
    (if (>= (length qqq) nr) qqq
        nil)))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(defun superxxx (listi iperlista ni nn nr)
  (if (null iperlista) nil
      (remove nil (append (selectxxx listi iperlista ni nn nr)
                          (superxxx listi (cdr iperlista) ni nn nr)))))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(defun interp-list-interval (Lspectr etapes coeff listi ni nn nr)
  (superxxx listi (gobackplus lspectr etapes coeff) ni nn nr))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;; DATA UNA LISTA DI LISTE D'INTERVALLI, LI INTEGRA IN SPETTRI FM
;;; ABBASSANDO IL RATIO; E' POSSIBILE DEFINIRE QUANTI DEGLI
;;; INTERVALLI DELLA LISTA DEVONO ESSERE OBBLIGATORIAMENTE
;;; PRESENTI NELLO SPETTRO (NI) E IN QUALE DENSITA'
;;; MINIMA (NN ) RISPETTO AL NUMERO COMPLESSIVO
;;; DEGLI INTERVALLI DELLO SPETTRO (INDEX * 2)
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

(defun caryy2 (listi ni nn np fond coeff)
  (car (yy2 listi ni nn np fond coeff)))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(defun caryy2ratio (listi ni nn port ratio index)
  (car (yy2ratio listi ni nn port ratio index)))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(defun multiyy2ratio (lista ni nn port ratio index)
  (if (null lista) nil
      (cons (caryy2ratio (car lista) ni nn port ratio index)
            (multiyy2ratio (cdr lista) ni nn port ratio index))))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;; TROVA IL RATIO FM DI UNO SPETTRO FM I CUI PARZIALI
;;; SONO IN ORDINE DI FREQUENZA CRESCENTE
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

(defun findratio/sort (lista port)
  (let ((listar (remove port lista)))
    ( / (/ ( - (car(reverse listar)) port) (/ (- (length lista)1) 2))
        port )))

```

```

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;; TROVA IL RATIO FM DI UNA LISTA DI SPETTRI FM I CUI PARZIALI
;;; SONO IN ORDINE DI FREQUENZA CRESCENTE
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

(defun fmulratiosort (lista port)
  (if (null lista) nil
      (cons (findratio/sort (car lista ) port)
            (fmulratiosort (cdr lista ) port))))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;; CALCOLA L'INTERPOLAZIONE TRA DUE O PIU' SPETTRI FM
;;; IN BASE AD UNA LISTA D'INTERVALLI.
;;; E' POSSIBILE DEFINIRE QUANTI DEGLI INTERVALLI
;;; DELLA LISTA DEVONO ESSERE OBBLIGATORIAMENTE PRESENTI
;;; NELLO SPETTRO (NI) E IN QUALE DENSITA' MINIMA (NN )
;;; RISPETTO AL NUMERO COMPLESSIVO DEGLI INTERVALLI
;;; DELLO SPETTRO, E QUANTE TAPPE INTERMEDIARIE DELL'INT
;;; DEVONO RISPONDERE A QUESTE CARATTERISTICHE AFFINCHE'
;;; L'INTERP VENGA EDITA (gobackratioplus + superxxx)
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

(defun many-fmr/sort (port list1 index)
  (if (null list1) nil
      (cons (fmr/sort port (car list1) index)
            (many-fmr/sort port (cdr list1) index))))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(defun interp-fmr (lspectr port etapes coeff)
  (many-fmr/sort port
    (interplus0 (fmulratiosort lspectr port) etapes coeff)
    (/(- (length (car lspectr)) 1) 2)))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(defun gobackratioplus (lspectr port etapes coeff)
  (if (< coeff .1) nil
      (cons (interp-fmr lspectr port etapes coeff)
            (gobackratioplus lspectr port etapes (- coeff .05)))))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(defun interpFM-list-interval
  (lspectr port etapes coeff listi ni nn nr)
  (superxxx listi (gobackratioplus lspectr port etapes coeff) ni nn nr))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;DATA UNA LISTA DI INTERVALLI, LI TROVA ALL'INTERNO DI UNO SPETTRO,
;;;NELL'ORDINE STABILITO DALLA LISTA,E LI FA APPARIRE CANCELLANDO
;;;LE NOTE DELLO SPETTRO ESTRANEE
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

(defun kul (i f0 lista )
  (if (null lista) nil
      (if (and (< i (* (/ (car lista) f0) 1.0293022))
              (> i (* (/ (car lista) f0) .97153834)))
          (list f0 (car lista ))
          (kul i f0 (cdr lista)))))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(defun kul8ème (i f0 lista )
  (if (null lista) nil

```

```

      (if (and (< i (* (/ (car lista) f0) 1.015))
            (> i (* (/ (car lista) f0) .985)))
          (list f0 (car lista) )
          (kul i f0 (cdr lista))))))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(defun removeplus (lista listb)
  (let (( zzz(remove (car lista) listb)))
    (if (null lista) zzz
        (removeplus (cdr lista) zzz))))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(defun only+ (n lista)
  (if (null lista) nil
      (if (>= (car lista) n) lista
          (only+ n (remove (car lista) lista)))))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(defun kul2 (i lista)
  (if (null lista) nil
      (if ( null (kul i (car lista) (cdr lista)))
          (kul2 i (cdr lista))
          (kul i (car lista) (cdr lista)))))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(defun kkk/a (listk lista)
  (if (null listk) nil
      (let ((may (kul (car listk) (car lista) (cdr lista))))
        (if (null may) nil
            (remove-duplicates (append may (kkk/a (cdr listk)
                                                    (only+ (nth 1 may) lista))))))))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(defun cherche-int0 (listint listfreq)
  (if (null listfreq) nil
      (let ((bobo (kul2 (car listint) listfreq)))
        (let ((momo (kkk/a (cdr listint) (only+ (nth 1 bobo) listfreq))))
          (if (null momo) (cherche-int0 listint ( cdr listfreq))
              (remove-duplicates (append bobo momo)
                                  ))))))))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(defun cherche-int (listint listfreq)
  (if (null listfreq) nil
      (if (= (- (length (cherche-int0 listint listfreq)) 1)
              (length listint))
          (cherche-int0 listint listfreq)
          (cherche-int listint (cdr listfreq)))))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;; DATA UNA LISTA DI INTERVALLI, LI TROVA ALL'INTERNO
;;; DI UNA LISTA DI SPETTRI, NELL'ORDINE STABILITO DALLA LISTA,
;;; E LI FA APPARIRE CANCELLANDO LE NOTE DELLO SPETTRO
;;; ESTRANEE AGLI INTERVALLI
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

(defun multicherche (listint listspectr)
  (if (null listspectr) nil
      (cons (cherche-int listint (car listspectr))
            (multicherche listint (cdr listspectr)))))

```

```

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;; DATA UNA LISTA DI LISTE D'INTERVALLI, LI TROVA ALL'INTERNO DI UNO
;;; SPETTRO, NELL'ORDINE STABILITO DALLA LISTA, E LI FA APPARIRE
;;; CANCELLANDO LE NOTE DELLO SPETTRO ESTRANEE AGLI INTERVALLI
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

```

```

(defun multicherche1 (listinterplus listfreq)
  (if (null listinterplus) nil
      (cons (cherche-int0 (car listinterplus) listfreq)
            (multicherche1 (cdr listinterplus) listfreq))))

```

```

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;; DATA UNA LISTA DI LISTE D'INTERVALLI, LI TROVA ALL'INTERNO
;;; DI UNA LISTA DI SPETTRI, NELL'ORDINE STABILITO DALLA LISTA,
;;; E LI FA APPARIRE CANCELLANDO LE NOTE DELLO SPETTRO
;;; ESTRANEE AGLI INTERVALLI
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

```

```

(defun multicherche2 (listinterplus listspectr)
  (if (null listinterplus) nil
      (cons (cherche-int (car listinterplus) (car listspectr))
            (multicherche2 (cdr listinterplus) (cdr listspectr)))))

```

```

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;; DATA UNA LISTA DI INTERVALLI, LI TROVA IN UNO SPETTRO AGGIUNGENDO
;;; DELLE FREQUENZE, SE NECESSARIO, TRA UN PARZIALE E L'ALTRO
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

```

```

(defun quasi4 (i f0 f1)
  (if (and (< i (* (/ f1 f0) 1.0293022))
        (> i (* (/ f1 f0) .97153834)))
      (list f0 f1) nil))

```

```

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(defun quasi4plus (lista f0 f1)
  (if (null lista) nil
      (if (null (quasi4 (car lista) f0 f1))
          (quasi4plus (cdr lista) f0 f1)
          (list f0 f1))))

```

```

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(defun coco ( f0 lista)
  (cons f0
        (let ((toto nil))
          (let ((add f0))
            (dolist (x lista )
              (setq add (* add x ))
                (push add toto )))) (reverse toto))))

```

```

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(defun densdens (lista f0 f1 www)
  (let ((zzz (cons 1 lista)))
    (if (null www) nil
        (cons (car www )
              (if (null (quasi4plus zzz (car www) f1))
                  (densdens (cdr zzz) f0 f1 (cdr www))
                  nil))))))

```

```

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

```

```

(defun densdens1 (lista f0 f1)
  (let ((byby (densdens lista f0 f1 (coco f0 lista))))
    (if (null (quasi4plus lista
                (nth (- (length byby)1) byby)
                f1)) nil
        (append byby (cons f1 nil)))))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(defun alldensdens (lista f0 f1)
  (let ((byby (densdens lista f0 f1 (coco f0 lista))))
    (if (null (quasi4plus lista
                (nth (- (length byby)1) byby)
                f1)) (list f0 f1)
        (append byby (cons f1 nil)))))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;; ESCONO SOLO GLI INTERVALLI VOLUTI
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

(defun densifier (listint listfreq)
  (if (null (cdr listfreq)) nil
      (remove-duplicates
       (append (densdens1 listint (nth 0 listfreq) (nth 1 listfreq))
               (densifier listint (cdr listfreq))))))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;; ESCONO TUTTE LE NOTE DELLO SPETTRO+LE NOTE AGGIUNTE
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

(defun alldensifier(listint listfreq)
  (if (null (cdr listfreq)) nil
      (remove-duplicates
       (append (alldensdens listint (nth 0 listfreq) (nth 1 listfreq))
               (densifier listint (cdr listfreq))))))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;; DATA UNA LISTA DI INTERVALLI, LI TROVA IN UNA SERIE DI SPETTRI
;;; AGGIUNGENDO DELLE FREQUENZE, SE NECESSARIO, TRA UN PARZIALE
;;; E L'ALTRO ESCONO SOLO GLI INTERVALLI VOLUTI
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

(defun multidensifier (listint listspectr)
  (if (null listspectr) nil
      (cons (densifier listint (car listspectr))
            (multidensifier listint (cdr listspectr)))))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;; DATA UNA LISTA DI INTERVALLI, LI TROVA IN UNA SERIE DI SPETTRI
;;; AGGIUNGENDO DELLE FREQUENZE, SE NECESSARIO, TRA UN PARZIALE E
;;; L'ALTRO ESCONO TUTTE LE NOTE DELLO SPETTRO+LE NOTE AGGIUNTE
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

(defun multialldensifier (listint listspectr)
  (if (null listspectr) nil
      (cons (alldensifier listint (car listspectr))
            (multidensifier listint (cdr listspectr)))))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;; FREQUENCY-SHIFTING;PLUS/MINUS: 1 = ADD.,0 = SOTTR.,
;;; 10 = ADD. + SOTTR.

```

```

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(defun freq-shifting (lista plus/minus fz)
  (cond
    ((= plus/minus 1)
     (summa fz lista))
    ((= plus/minus 0)
     (sott fz lista))
    ((= plus/minus 10)
     (append (summa fz lista) (sott fz lista))))))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(defun shiftplus (lista plus/minus listfz)
  (if (null listfz) nil
      (cons (freq-shifting lista plus/minus (car listfz))
            (shiftplus lista plus/minus (cdr listfz)))))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;; SHIFTING SU LISTA D'ACCORDI CON LISTA DI FZ
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

(defun multishift (listchord plus/minus listfz)
  (if (null listchord) nil
      (cons (freq-shifting (car listchord) plus/minus (car listfz))
            (multishift (cdr listchord) plus/minus listfz))))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;; DIST DI SOLO UNA PORZIONE DELLO SPETTRO, ENTRO NMIN E NMAX
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

(defun exptplus (freq listb dist)
  (if (null listb) nil
      (cons (* freq (expt (car listb) dist))
            (exptplus freq (cdr listb) dist))))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(defun selectdist (np freq nmin nmax dist)
  (let ((bigolo
        (summa (- nmin 1) (spectrum 1 1 (- (+ nmax 1) nmin))))))
    (sort
     (append (exptplus freq bigolo dist)
             (removeplus (molt bigolo freq) (spectrum freq 1 np))) '<)))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(defun selectonlydist (freq nmin nmax dist)
  (let ((bigolo
        (summa (- nmin 1) (spectrum 1 1 (- (+ nmax 1) nmin))))))
    (exptplus freq bigolo dist)))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;; DATA UNA LISTA DI LISTE DI FZ LE INTEGRA IN SPETTRI
;;; DISTORTI SULLO STESSO FOND
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

(defun segmHH (listseg np fond coeff)
  (if (null listseg) nil
      (append (cons (car listseg) (hh (car listseg) np fond coeff))
              (segmHH (cdr listseg) np fond coeff))))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

```

```

(defun nthplus (lista listb)
  (if (null lista) nil
      (cons (nth (car lista) listb)
            (nthplus (cdr lista) listb))))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;; DATA UNA LISTA DI LISTE DI FZ LE INTEGRA IN SPETTRI
;;; FM SULLA STESSA PORT
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

(defun segmHHratio ( listseg port ratio index)
  (if (null listseg) nil
      (append(cons (car listseg)(hhratio (car listseg) port ratio index))
              (segmHHratio (cdr listseg) port ratio index))))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;; DATA UNA LISTA DI FZ E LA SEGMENTAZIONE IN INSIEMI,
;;; LI INTEGRA IN SPETTRI DISTORTI SULLO STESSO FOND
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

(defun segm-nthHH ( listfreq listnth np fond coeff)
  (if (null listnth) nil
      (append(cons (nthplus (sott 1 (car listnth)) listfreq)
                    (hh (nthplus (sott 1 (car listnth)) listfreq) np fond coeff))
              (segm-nthHH listfreq (cdr listnth) np fond coeff))))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;; DIVIDE L'ACCORDO IN SEGMENTI
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

(defun dividcord (listfreq listnth)
  (if (null listnth) nil
      (cons (nthplus (sott 1 (car listnth)) listfreq)
            (dividcord listfreq (cdr listnth)))))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;; TRASFORMA UN LISTA DI FZ IN UNA LISTA D'INTERVALLI
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

(defun cord-int (lista)
  (if (null (cdr lista)) nil
      (cons (/ (nth 1 lista) (nth 0 lista))
            (cord-int (cdr lista)))))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;; TRASFORMA UN LISTA DI LISTE DI FZ
;;; IN UNA LISTA DI LISTE D'INTERVALLI
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

(defun cord-intplus (listspectr)
  (if (null listspectr) nil
      (cons (cord-int (car listspectr))
            (cord-intplus (cdr listspectr)))))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;FMR SU DIFFERENTI PORT E RATIOS
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

(defun fmrbis (port ratio index)
  (cons port (append (fmspectrum1 port (* port ratio) index)
                    (cdr listspectr))))

```



```

(remove 0 (repl (fmspectrum2 port (* port ratio) index))))

;;;;;;;;;;;;;
(defun fmrratioplus (port listratio index)
  (if (null listratio) nil
      (cons (fmrbis port (car listratio) index)
            (fmrratioplus port (cdr listratio) index))))

;;;;;;;;;;;;;
(defun fmrportplus (listport ratio index)
  (if (null listport) nil
      (cons (fmrbis (car listport) ratio index)
            (fmrportplus (cdr listport) ratio index))))

;;;;;;;;;;;;;
(defun fmrplusplus (listport listratio index)
  (if (null listratio) nil
      (cons (fmrbis (car listport) (car listratio) index)
            (fmrplusplus (cdr listport) (cdr listratio) index))))

;;;;;;;;;;;;;
(defun fmmodplus (port listmod index)
  (if (null listmod) nil
      (cons (fm port (car listmod) index)
            (fmmodplus port (cdr listmod) index))))

;;;;;;;;;;;;;
;;;DISTORCE UN AGGREGATO DI FREQUENZE
;;;;;;;;;;;;;

(defun deform (list dist)
  (if (null list) nil
      (cons (expt (car list) dist)
            (deform (cdr list) dist))))

;;;;;;;;;;;;;
(defun deformplus (listfz listdist)
  (if (null listdist) nil
      (cons (deform listfz (car listdist))
            (deformplus listfz (cdr listdist))))
;;;;;;;;;;;;;

```

C.2 WORKLISP²³²

```

;;;                               INSTR 1 - ADDITIVA SEMPLICE
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(with-open-file (nom-fichier "Macintosh HD:Users:FR:getStarted:lorc.orc"
                  :direction :output :if-exists :overwrite
                  :if-does-not-exist :create)

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(ENTER 48000 1000 1
 nom-fichier)

      (NINSTR
;;;/ninstr                               -max ampiezza
      1                                   100
;;;-nparz.
      20
 nom-fichier )(IPERQENVELOPE 'kenv
;;;/tipo curva                             -numero sezioni parziali
      'expseg                             '(1 10 20)
;;;-modello env amp                         -rapp sez parz con env amp
      '(1 20 1)                             '(.5 1 .2)
;;;-q verticale amp                         -mod env time
      '(1 1 1)                               '(0 2 4)
;;; -rapp sez parz con mod env time        -q verticale time.
      '(1 .8 .6)                             '(1 1 1)
 nom-fichier )(VIBRALEA
;;;/tipo curva                             -prof vibr
      'expon                                 2.2
;;;-env prof vibr                           -vit vibr
      '(1 2 1)                               7.3
;;;-env vit vibr                            -env time.
      '(1 1.8 1)                             '(0 2 4)
;;;-alea vibr in % .
      '(.2 .8 .3 .5)
 nom-fichier )(JITTERPLUS
;;;/jitter.
      '(1 .8 .5 1.3 1.2 1.1)
 nom-fichier )(OSCILI
;;;-frequenze.
      (chorus (spectrum 98 1 10) '(1 1.414))
 nom-fichier )(EXIT
;;;/tipo out                               -numero out
      'out                                   '(6 6 6 2)
;;;/evaluate.
 nom-fichier ))
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

;;;                               INSTR 2 - ADDITIVA CON FORMANTI
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(with-open-file (nom-fichier "hammer1280:Fausto:Csound:brr.orc"
                  :direction :output :if-exists :overwrite
                  :if-does-not-exist :create)

```

²³² Si veda *infra*, par. 3.2.3.

```

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(ENTER 48000 1000 1
nom-fichier)
      (NINSTR
;;;/ninstr                               -max ampiezza
      1                                   100000
;;;-nparz.
      50
nom-fichier )(IPERQENVELOPE 'kenv
;;;/tipo curva                            -numero sezioni parziali
      'expseg                             '(1 32 50)
;;;-modello env amp                       -rapp sez parz con env amp
      '(1 50000 1)                         '(1 .8 .5)
;;;-q verticale amp                       -mod env time
      '(1 1 1)                             '(0 3 6 )
;;; -rapp sez parz con mod env time      -q verticale time.
      '(1 .9 .8)                           '(1 1 1)
nom-fichier )(VIBRALEA
;;;/tipo curva                            -prof vibr
      'linseg                              2.2
;;;-env prof vibr                         -vit vibr
      '(1 1.5 1)                           7.3
;;;-env vit vibr                          -env time.
      '(1 1.3 1)                           '(0 3 6)
;;;-alea vibr in % .
      '(.2 .8 .3 .5)
nom-fichier )(JITTERPLUS
;;;/jitter.
      '(1 .8 .9 1.2 1.2 .5)
nom-fichier )(OSCILI
;;;/frequenze.
      (chorus (spectrum 65.4 1 25) '(1 1.09))
nom-fichier )(KELENC
;;;/tipo curva                            -elenco freq centrali formanti
      'linseg                              '((609 1000 2450)(587 987 2450)(587
987 2450)(554 686 2450)(554 686 2450))
;;;-elenco amp banda formanti.           -env time.
      '((78 88 123)(78 88 123 )(3 3 3 )(78 88 123 )(78 88 123))
      '(0 2 3 4 5 )
nom-fichier )(KFORMANTEXTIT
;;;/amp filtri                            -alea banda in %
      '(.6 .8 .3)                          '(.4 .9)
;;;-tipo out                              -n out
      'out                                   '(8 8 8 8 8 8 2 )
;;;-n formanti.
      3
;;;/evaluate.
nom-fichier))
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

;;;                                     INSTR 2 - FILTRAGGIO BRUIT
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(with-open-file (nom-fichier "taranis-
1Giga:Users:Pottier:Fausto:csound:xxh.orc"
:direction :output :if-exists :overwrite
:if-does-not-exist :create)
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(ENTER 44100 100 1

```

```

nom-fichier)

                                (NINSTR
;;;/ninstr                      -max ampiezza
    1                          100000
;;;-nparz.
    16
    nom-fichier ) (SUPERQENVELOPE 'kband
;;;/tipo curva                  -numero sezioni parziali
    'expseg                     '(1 20 16)
;;;-modello env band           -rapp sez parz con env band
    '(5 .5 5)                   '(.2 .5 1)
;;;-q verticale band           -mod env time.
    '(1 1 1)                     '(0 2 4)
    nom-fichier ) (ABRUIT
;;;-tipo curva                  -amp banda bruit
    'linseg                       '(0 20 0)
;;;-vitesse bruit              -env time.
    '(5000 5000 5000 )          '(0 2 4 )
    nom-fichier ) (RESON
;;;/amp filtri                  -freq centrali
    (jinterpol 2 14 1 .5)        (chorus (dist 20 98 1.212) '(1
1.12))
;;;-alea banda in %
    '(.3 .9)
    nom-fichier ) (EXIT
;;;/tipo out                    -numero out
    'out                          '(8 8)
;;;/evaluate.
    nom-fichier ))

;aimpuls soundin "cymroll", 0
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

;;;                               INSTR 3 - sndin BRUIT
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(with-open-file (nom-fichier "taranis-
1Giga:Users:Pottier:Fausto:csound:xxhRTSGX99.orc"
                :direction :output :if-exists :overwrite
                :if-does-not-exist :create)

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(ENTER 44100 100 1
nom-fichier)

                                (NINSTR
;;;/ninstr                      -max ampiezza
    1                          100000
;;;-nparz.
    60
    nom-fichier ) (SUPERQENVELOPE 'kband
;;;/tipo curva                  -numero sezioni parziali
    'expseg                     '(1 60)
;;;-modello env band           -rapp sez parz con env band
    '(5 5)                       '(1 1)
;;;-q verticale band           -mod env time.
    '(1 1)                         '(0 2)
    nom-fichier ) (ASNDIN
;;;-fichier                      -temps départ

```

```

'cymroll                                0
nom-fichier) (RESON-SF
;;;/amp filtri                            -freq centrali
      (jinterpol 2 58 1 .1)              (SORT (PW::FLAT (distcord 10 '(311.
466.16 659.25 1108.7 1396.9 1975.5) 1)) '<)
;;;-alea banda in %
      '(.3 .9)
nom-fichier ) (EXIT
;;;/tipo out                              -numero out
      'out                                '(8 8 8 8 8 8 8 4)
;;;/evaluate.
nom-fichier ))

;aimpuls soundin "cymroll", 0
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

;;;                                     INSTR 4 - FILTRAGGIO FONTE PERIODICA
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(with-open-file (nom-fichier "Macintosh HD:Users:FR:getStarted:lorc.orc"
                  :direction :output :if-exists :overwrite
                  :if-does-not-exist :create)

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(ENTER 48000 1000 1
nom-fichier)

      (NINSTR
;;;/ninstr                                -max ampiezza
      1                                    100
;;;-nparz.
      3
nom-fichier )(SUPERQENVELOPE 'kband
;;;/tipo curva                            -numero sezioni parziali
      'expseg                              '(1 5 11)
;;;-modello env band                      -rapp sez parz con env band
      '(1 20 1)                            '(.5 1 .2)
;;;-q verticale band                      -mod env time.
      '(1 1 1)                              '(0 6 12)
nom-fichier )(LINE 'kgliss 1
;;;/tipo curva                            -env frequenze
      'expseg                              '(68 100)
;;;-env time.
      '(0 12)

nom-fichier )(APERIOD
;;;-tipo curva                            -env amp
      'linseg                              '(0 20 0)
;;;-env time                              -frequenza.
      '(0 6 12)                            'gkliss1
nom-fichier )(RESON
;;;/amp filtri                            -freq centrali
      '(.2 .3)                              '(200 300)
;;;-alea banda in %
      '(.3 .9)
nom-fichier )(EXIT
;;;/tipo out                              -numero out
      'out                                '(2 )
;;;/evaluate.

```

```

nom-fichier ))
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;
                                INSTR 5 - ADDITIVA CON MOD D'AMP
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(with-open-file (nom-fichier "Cursus FX1:Users:FR:getStarted:6orc.orc"
                        :direction :output :if-exists :overwrite
                        :if-does-not-exist :create)

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(ENTER 48000 1000 1
nom-fichier)

                                (NINSTR
;;;/ninstr                                -max ampiezza
    1                                      100
;;;-nparz.
    20
nom-fichier )(IPERQENVELOPE 'kenv
;;;/tipo curva                                -numero sezioni parziali
    'expseg                                '(1 2)
;;;-modello env amp                          -rapp sez parz con env amp
    '(1 20 1)                              '(.5 1 )
;;;-q verticale amp                          -mod env time
    '(1 1 1)                                '(0 6 12)
;;; -rapp sez parz con mod env time          -q verticale time.
    '(1 .8 )                               '(1 1 1)
nom-fichier )(VIBRALEA
;;;/tipo curva                                -prof vibr
    'expon                                  2.2
;;;-env prof vibr                            -vit vibr
    '(1 2 1)                                7.3
;;;-env vit vibr                             -env time.
    '(1 1.8 1)                             '(0 7 12)
;;;-alea vibr in % .
    '(.2 .8 .3 .5)
nom-fichier )(JITTERPLUS
;;;/jitter.
    '(1 .8 .5 1.3 1.2 1.1)
nom-fichier )(LINEND 'kenv 0
;;;/tipo curva                                -env amp mod
    'linseg                                '(0 0 1 1)
;;;-env time mod.
    '(0 6 7 12)
nom-fichier )(AM
;;;-frequenze portanti                      -ratio portante/modulante
    '(261 830)                             '(1.414 1.26)
nom-fichier )(AMEXIT
;;;/tipo out                                -numero out
    'out                                    '(2 )
;;;/evaluate.
nom-fichier ))
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;
                                INSTR 6 - FM
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(with-open-file (nom-fichier "Macintosh HD:Users:FR:getStarted:lorc.orc"
                        :direction :output :if-exists :overwrite
                        :if-does-not-exist :create)

```

```

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(ENTER 48000 1000 1
nom-fichier)

(NINSTR

;;;/ninstr -max ampiezza
1 100
;;;-nparz.
20
nom-fichier )(IPERQENVELOPE 'kenv
;;;/tipo curva -numero sezioni parziali
'expseg '(1 2 3)
;;;-modello env amp -rapp sez parz con env amp
'(1 20 1) '(.5 1 .2)
;;;-q verticale amp -mod env time
'(1 1 1) '(0 6 12)
;;; -rapp sez parz con mod env time -q verticale time.
'(1 .8 .6) '(1 1 1)
nom-fichier )(LINE 'kindex 1
;;;/tipo curva -env indice fm
'expseg '(10 1 1)
;;;-env time indice.
'(0 0.2 12)
nom-fichier )(SUPERQENVELOPE 'kport
;;;/tipo curva -numero sezioni parziali
'expseg '(1 2 )
;;;-modello env freq -rapp sez parz con env freq
'(100 261 261) '(1.028 1.414)
;;;-q verticale freq -mod env time.
'(1 1 ) '(0 6 12)
nom-fichier)(FOSCILI
;;;-ratio portante/modulante
'(1 1.12 )
nom-fichier )(EXIT
;;;/tipo out -numero out
'out '(3 )
;;;/evaluate.
nom-fichier ))
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

;;; INSTR 7 - FM CON FORMANTI
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(with-open-file (nom-fichier "Macintosh HD:Users:FR:getStarted:lorc.orc"
:direction :output :if-exists :overwrite
:if-does-not-exist :create)

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(ENTER 48000 1000 1
nom-fichier)

(NINSTR

;;;/ninstr -max ampiezza
1 100
;;;-nparz.
20
nom-fichier )(IPERQENVELOPE 'kenv
;;;/tipo curva -numero sezioni parziali
'expseg '(1 2 3)
;;;-modello env amp -rapp sez parz con env amp
'(1 10 1) '(.5 1 .2)

```

```

;;;-q verticale amp -mod env time
      '(1 1 1) '(0 6 12)
;;; -rapp sez parz con mod env time -q verticale time.
      '(1 .8 .6) '(1 1 1)
nom-fichier )(LINE 'kindex 1
;;;/tipo curva -env indice fm
      'expseg '(10 1 1)
;;;-env time indice.
      '(0 0.2 12)
nom-fichier )(SUPERQENVELOPE 'kport
;;;/tipo curva -numero sezioni parziali
      'linseg '(1 2 )
;;;-modello env freq -rapp sez parz con env freq
      '(100) '(1.028 1.414)
;;;-q verticale freq -mod env time.
      '(1 1 1) '(0 6 12)
nom-fichier)(FOSCILI
;;;-ratio portante/modulante(per ogni port)
      '(1 1.12 )
nom-fichier )(KELENC
;;;/tipo curva -elenco freq centrali(kgliss)
      'linseg '((609 1000 2450)(360 750 2400))
;;;-elenco amp banda(kband). -env time.
      '((78 88 123)(51 61 168)) '(0 12)
nom-fichier )(KFORMANTEXTIT
;;;/amp filtri 005 .003 '(.4 .9)
;;;-tipo out -n out
      'out '( 2)
;;;-n formanti.
      3
;;;/evaluate.
nom-fichier))
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(with-open-file (nom-fichier "Macintosh HD:Users:FR:getStarted:6orc.orc"
                      :direction :output :if-exists :overwrite
                      :if-does-not-exist :create)

(ENTER 48000 1000 1
nom-fichier)
      (NINSTR
;;;/ninstr -max ampiezza
      1 100
;;;-nparz.
      50
nom-fichier )(IPERQENVELOPE 'kenv
;;;/tipo curva -numero sezioni parziali
      'linseg '(1 20 40)
;;;-modello env amp -rapp sez parz con env amp
      '(0 15 0) '(.2 .4 1)
;;;-q verticale amp -mod env time
      '(1 1 1) '(0 6 12)
;;; -rapp sez parz con mod env time -q verticale time.
      '(.8 .9 1) '(1 1 1)
nom-fichier )(VIBRALEA
;;;/tipo curva -prof vibr
      'linseg 2.2
;;;-env prof vibr -vit vibr
      '(1 2 1) 7.3
;;;-env vit vibr -env time.

```



```

      '(1 1.8 1)                                '(0 7 12)
;;;-alea vibr in % .
      '(.2 .8 .3 .5)
      nom-fichier )(JITTERPLUS
;;;/jitter.
      '(1 .8 .9 1.2 1.2 .5)
NOM-FICHIER )(OSCILI
;;;/frequenze.
      (chorus (spectrum 98 1 18) '(1 1.189))
      nom-fichier )(KELENC
;;;/tipo curva
      'expseg
;;;-elenco freq centrali formanti
      '((360 750 2400)(609 1000 2450)(400 1700 2300)
(360 750 2400)(609 1000 2450)(360 750 2400))

;;;-elenco amp banda formanti.
      '((51 61 168)(78 88 123)(64 81 101)(51 61 168)(78 88 123)(51 61 168))
;;;-env time.
      '(0 2 2.5 8 10 12 )
nom-fichier )(KFORMANTEXTIT
;;;/amp filtri                                -alea banda in %
      '(.005 .004 .003)                        '(.4 .9)
;;;-tipo out                                  -n out
      'out                                        '(6 6 6 6 6 6 4)
;;;-n formanti.
      3
;;;/evaluate.
      nom-fichier))
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;;;;;;;;;;;;;;;;;;;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;; INSTR 2BIS - ADDITIVA CON FORMANTI E SCORE COMPLESSO
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(with-open-file (nom-fichier "CMAO:People:Fausto:orc10"
                        :direction :output :if-exists :overwrite
                        :if-does-not-exist :create)

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(ENTER 48000 1000 1
nom-fichier)
      (NINSTR
;;;/ninstr                                -max ampiezza
      1                                    100
;;;-nparz.
      70
      nom-fichier )(PLINEX 'kenv 1
;;;/tipo curva
      'linseg
;;;-numero p*ampl                            -numero p*time
      4                                        5
;;; -modello env ampl                        -modello env time
      '(20 0)                                '(0 12)
nom-fichier )(VIBRALEA
;;;/tipo curva                                -prof vibr
      'linseg                                2.2
;;;-env prof vibr                            -vit vibr
      '(1 1.5 1)                            7.3
;;;-env vit vibr                            -env time.
      '(1 1.3 1)                            '(0 7 8)

```

```

;;;-alea vibr in % .
      '(.2 .8 .3 .5)
nom-fichier )(JITTERPLUS
;;;/jitter.
      '(1 .8 .9 1.2 1.2 .5)
nom-fichier )(PLINEX 'kfreq 1
;;;/tipo curva
      'linseg
;;;-numero p*freq1          -numero p*freq2
      6                      7
;;;-env frequenze          -env time
      '(1 1)                  '(0 .5)
nom-fichier)(format      nom-fichier  "-%a1      oscili      kenv*iscaling,
kfreq*kvib*kjitter,1~%"
      (KELENC
;;;/tipo curva          -elenco freq centrali formanti
      'linseg          (list fi fi fu fu)
;;;-elenco amp banda formanti.
      (list bi bi bu bu)          -env time.
                                '(0 2 6 12 )
nom-fichier )(KFORMANTEXTIT
;;;/amp filtri          -alea banda in %
      '(.006 .008 .003 .002)          '(.4 .9)
;;;-tipo out          -n out
      'out          '(7 7 7 7 7 7 7 7 7 7)
;;;-n formanti.
      4
;;;/evaluate.
nom-fichier))
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

```

BIBLIOGRAFIA

Bibliografia generale

CARLOS AUGUSTO AGON AMADO, *OpenMusic : Un langage visuel pour la composition musicale assistée par ordinateur*, tesi di dottorato sostenuta presso l'Université Paris 6, 1998.

ANSI CL standard₁₉₉₄. ANSI INCITS 226-1994, *Information Technology - Programming Language - Common Lisp*.

Alessandro Arbo (a cura di), *Anamorphoses. Études sur l'œuvre de Fausto Romitelli*, Parigi, Hermann, 2015.

Alessandro Arbo (a cura di), *Oltre le periferie dell'impero. Omaggio a Fausto Romitelli*, Torino, Trauben, 2014.

ALESSANDRO ARBO, "Risonanze e anamorfosi. Note sulla poetica di Fausto Romitelli", in Amalia Collisani, Gabriele Garilli, Gaetano Mercadante, *Italia/Francia. Musica e cultura nella seconda metà del XX secolo*, Palermo, L'Epos, 2009, pp. 325-345.

Alessandro Arbo (a cura di), *Le corps électrique. Voyage dans le son de Fausto Romitelli*, Parigi, L'Harmattan, 2005.

Alessandro Arbo (a cura di), *Il corpo elettrico. Viaggio nel suono di Fausto Romitelli*, Monfalcone, Teatro comunale di Monfalcone, 2003.

GERARD ASSAYAG, MICHELE CASTELLENGO, CLAUDY MALHERBE, "Functional Integration of Complex Instrumental Sounds in Music Writing", *Proceedings of the International Computer Music Conference*, Burnaby, Canada, 1985.

GERARD ASSAYAG, CAMILO RUEDA et al., "Computer-Assisted Composition at IRCAM: From *PatchWork* to *OpenMusic*", in «Computer Music Journal», v. 23 n. 3, 1999, pp. 59-72.

GERARD ASSAYAG, "Computer Assisted Composition today", in «Proceedings of 1st Symposium on Music and Computers», Corfu, 1998. <recherche.ircam.fr/equipements/repmus/RMPapers/Corfou98/>, ultima visita 5 marzo 2017.

GERARD ASSAYAG, “CAO : vers la partition potentielle”, in *Les cahiers de l'IRCAM : la composition assistée par ordinateur*, Parigi, IRCAM, 1993, pp. 13-41.

MILTON BABBITT, “The Use of Computers in Musicological Research”, in «Perspectives of New Music», v. 3, n. 2, 1965, pp. 74-83.

BRUNO BACHIMONT, “Préservation culturelle numérique”, in Evelyne Gayou (a cura di), *Musique et Technologie. Préserver - Archiver - Re-produire*, Parigi, INA, 2013, pp. 11-32.

PIERRE-FRANÇOIS BAISNEE, YVES POTARD, JEAN-BAPTISTE BARRIERE, “Methodologies de synthèse du timbre : l'exemple des modèles de résonance”, in Jean-Baptiste Barrière (a cura di), *Le timbre, métaphore pour la composition*, Parigi, Christian Bourgois, 1990, pp. 135-163.

Jean-Baptiste Barrière (a cura di), *Le timbre, métaphore pour la composition*, Parigi, Christian Bourgois, 1991.

JEAN-BAPTISTE BARRIERE, “Devenir de l'écriture musicale assistée par ordinateur : formalisme, forme, aide à la composition”, in «Analyse musicale», 20, 1990, pp. 52-68.

JEAN-BAPTISTE BARRIERE, *Pour une esthétique de la musique avec ordinateur*, tesi di dottorato sotto la direzione di Olivier Revault d'Allonnes presso l'Université de Paris I – Panthéon-Sorbonne, 1990.

MARC BATTIER, *Electroacoustic music studies and the danger of loss*, in «Organised Sound», v. 9 n. 1, 2004, pp. 47-53.

MARC BATTIER, LAURENT POTTIER, *Fausto Romitelli, En trance (1995), documentation d'exploitation*, Parigi, IRCAM, 1998.

WALTER BENJAMIN, *Aura e choc. Saggi sulla teoria dei media*, Torino, Einaudi, 2012.

NICOLA BERNARDINI, ALVISE VIDOLIN, “Sustainable Live Electro-Acoustic Music”, in «Proceedings of Sound and Music Computing 2005», Salerno, 2005, <http://econtact.ca/8_3/bernardini_vidolin.html>, ultima visita 5 marzo 2017.

RICCARDO BIANCHINI, ALESSANDRO CIPRIANI, *Il suono virtuale*, ConTempoNet, 2011.

DANIEL G. BOBROW, LINDA G. DEMICHIEL et al., “Common Lisp Object System specification”, in «Higher-order And Symbolic Computation», v. 1, n. 3-4, 1989.

ALAIN BONARDI, “Pérenniser pour transmettre, transmettre pour pérenniser. Destins de l'œuvre mixte interactive – Autour de *En Echo*, pièce de Philippe Manoury”, in Evelyne Gayou (a cura di), *Musique et Technologie. Préserver - Archiver - Re-produire*, Parigi, INA, 2013, pp. 105-125.

GEORGE BOSTON, *Guide to the basic technical equipment required by audio, film and television archives*, Londra, Coordinating Committee for the Technical Commissions of the International Organisations for Audio, Film and Television Archives., & International Federation of Film Archives, 1991.

PIERRE BOULEZ, “Timbre and composition - timbre and language”, in «Contemporary Music Review», v. 2 n. 1, 1987, pp. 161-171.

GUILLAUME BOUTARD, FABRICE MARANDOLA, “Mixed music creative process documentation methodology: outcomes of the DiP-CoRE project”, in *Proceeding of the 9th Conference on Interdisciplinary Musicology (CIM'14)*, 2014. Disponibile online all'indirizzo <https://www.academia.edu/20970403/Mixed_music_creative_process_documentation_methodology_outcomes_of_the_DiP-CoRE_project>, ultima visita 5 marzo 2017.

Kevin Bradley (a cura di), *Guidelines on the Production and Preservation of Digital Audio Objects – IASA-TC 04*, Auckland Park, IASA, 2009.

FRITJOF CAPRA, PIER LUIGI LUISI, *Vita e natura. Una visione sistemica*, Sansepolcro, Aboca, 2014.

MICHELE CASTELLENGO, *Ecoute musicale et acoustique*, Parigi, Eyrolles, 2015.

CCSDS: The Consultative Committee for Space Data Systems, *Reference model for an open archival information system (OAIS). Recommended practice*, CCSDS 650.0-M-2, Magenta book, 2012.

MARIO CERUTTI, *Il vincolo e la possibilità*, Milano, Raffaello Cortina, 2009.

GREGORY CHAITIN, *Alla ricerca di Omega*, Milano, Adelphi, 2007.

JOHN CHOWNING, “The Synthesis of Complex Audio Spectra by Means of Frequency Modulation”, in «Journal of the Audio Engineering Society», v. 21, n. 7, 1973, pp. 526-534.

EMMANUEL CHUILON, “L'univers des claviers électroniques : enjeux et limites”, in «Inharmonique» n. 7, *Musique et authenticité*, Parigi, Librairie Séguier/IRCAM, 1991, pp. 224-236.

JOEL E. COHEN, “Information Theory and Music”, in «Behavioral Science», v. 7 n. 2, 1962, pp. 137-163.

JACOPO CONTI, “*Corroded by noise: Il suono di Fausto Romitelli*”, in «Musica/realtà», n. 96, 2011, pp. 91-116.

OMER CORLAIX, “Le Coursus de composition et d'informatique musicale à l'Ircam”, in «Résonance», n. 14, 1998, pp. 12-15.

ANTHONY CORNICELLO, *Timbral Organization in Tristan Murail's Désintégrations and Rituals by Anthony Cornicello*, tesi di dottorato presso la Faculty of the Graduate School of Arts and Sciences, Brandeis University, Waltham, 2000.

LUCA COSSETTINI, ANGELO ORCALLI, *L'invenzione della fonologia musicale. Saggi sulla musica sperimentale di Luciano Berio e Bruno Maderna*, Lucca, LIM, 2015.

LUCA COSSETTINI, ANGELO ORCALLI, "Da *Sortie vers la lumière du jour* a *Jour, Contre-jour*. Riscrittura di un mito dell'antico Egitto", in Luca Cossettini (a cura di), *Scritture e ri-mediazioni / Écritures et re-productions*, Lucca, LIM, 2013, pp. 271-308.

LUCA COSSETTINI, ANGELO ORCALLI, "Témoignages sonores et critique des sources audiovisuelles", in «Techné», n. 37, 2013, pp. 72-78.

LUCA COSSETTINI, *Opere chiuse in sistemi aperti. Autopoiesi nella musica elettronica*, Lucca, LIM, 2013.

Luca Cossettini (a cura di), *Luigi Nono. Studi, edizione, testimonianze*, Lucca, LIM, 2010.

HUGH DAVIES, "A Simple Ring-modulator", in «Musics», n. 6, 1976, pp. 3-5.

VINCENT DEHOUX, "Ethnomusicologie II : Tibet", in «Musique en jeu», n. 22, 1976, pp. 119-126.

CHARLES DODGE, THOMAS JERSE, *Computer Music. Synthesis, Composition, and Performance*, Schirmer, 1997.

HUGUES DUFOURT, *Musique, pouvoir, écriture*, Parigi, Christian Bourgois, 1991. Trad. it. *Musica, potere, scrittura*, Milano/Lucca, Ricordi/LIM, 1997.

HALSON V. EAGLESON, ORAN W. EAGLESON, "Identification of Musical Instruments When Heard Directly and over a Public-Address System", in «The Journal of the Acoustical Society of America», v. 19 n. 2, 1947, pp. 338-342.

ROBERT ESCARPIT, *Théorie générale de l'information et de la communication*, Parigi, Hachette, 1976. Trad. it. *Teoria dell'informazione e della comunicazione*, Roma, Editori Riuniti, 1979.

NICOLAS ESPOSITO, "L'exemple de la préservation des jeux vidéo", in EVELYNE GAYOU (a cura di), *Musique et Technologique. Préserver - Archiver - Re-produire*, Parigi, INA, 2013, pp. 197-217.

NICOLAS ESPOSITO, YANN GESLIN, "Long-term preservation of acousmatic works: Toward a generic model of description", *Electrotechnical Conference, 2008*, MELECON 2008. The 14th IEEE Mediterranean, pp. 270-274.

Fausto Romitelli. *23° Festival di Milano Musica – Percorsi di musica d'oggi*, Milano, Milano Musica – Teatro alla Scala, 2014.

DOMENICO FIORMONTE, *Scrittura e filologia nell'era digitale*, Torino, Bollati Boringhieri, 2003.

MAURIZIO GABBRIELLI, SIMONE MARTINI, *Programming Languages: Principles and Paradigms*, Londra, Springer, 2010

Evelyne Gayou (a cura di), *Musique et Technologique. Préserver – Archiver – Re-produire*, Parigi, INA, 2013.

ANASTASIA GEORGAKI, “The grain of Xenakis technological thought on the computer music research of our days”, *Proceedings of the International Symposium Iannis Xenakis*, University of Athens, 2005. <<http://cicm.mshparisnord.org/ColloqueXenakis/papers/Georgaki.pdf>>, ultima visita 5 marzo 2017.

ANDREW GERZSO, “Paradigms and Computer Music”, in «Leonardo Music Journal», v. 2 n. 1, 1992, p. 73-79.

YANN GESLIN, “*Saturne*, d'Hugues Dufourt ; ou la préservation d'une œuvre avec électronique”, in Evelyne Gayou (a cura di), *Musique et Technologique. Préserver – Archiver – Re-produire*, Parigi, INA, 2013, pp. 127-144.

JOHN GREY, “Timbre discrimination in musical patterns”, in «Journal of the Acoustical Society of America», v. 62 n. 2, 1978, pp. 467-472.

JOHN GREY, “Scaling of musical timbre”, in «Journal of the Acoustical Society of America», v. 61 n. 5, 1977, pp. 1270-1277.

GERARD GRISEY, *Écrits, ou l'invention de la musique spectrale*, a cura di Guy Lelong, Parigi, Éditions MF, 2008.

GERARD GRISEY, “Structuration des timbres dans la musique instrumentale”, in *Écrits, ou l'invention de la musique spectrale*, a cura di Guy Lelong, Parigi, Éditions MF, 2008.

MARIA GUERCIO, “La notion d'authenticité en conservation numérique”, in Evelyne Gayou (a cura di), *Musique et Technologique. Préserver – Archiver – Re-produire*, Parigi, INA, 2013, pp. 75-92.

PETER HANAPPEUNDER, MARIE-HELENE SERRA, *AudioSculpt User's manual*, Parigi, IRCAM, 1996.

HERMANN L.F. HELMHOLTZ, *On the Sensations of Tone as a Physiological Basis for the Theory of Music*, traduzione in inglese di Alexander J. Ellis, London / New York, Longmans, Green, and Co., 1895.

C.W. HEWLETT, "Analysis of Complex Sound Waves", in «Physical Review, serie I», v. 35 n. 5, pp. 359-372.

LEJAREN A. HILLER, ROBERT A. BAKER, "Computer Cantata: A Study in Compositional Method", in «Perspectives of New Music», v. 3/1, 1964, pp. 62-90.

FRANCISCO IOVINO, MIKAEL LAURSON, LAURENT POTTIER, *PW-Chant – Reference*, Parigi, IRCAM, 1996.

FRANCISCO IOVINO, MIKAEL LAURSON, LAURENT POTTIER, *PW-Chant – Tutorial*, Parigi, IRCAM, 1996.

La composition assistée par ordinateur, «Le cahiers de l'IRCAM» n. 3, Parigi, IRCAM, 1993.

La synthèse sonore, «Le cahiers de l'IRCAM» n. 2, Parigi, IRCAM, 1993.

Composition et environnements informatiques, «Le cahiers de l'IRCAM» n. 1, Parigi, IRCAM, 1991.

Rapport IRCAM 1993, consultabile presso la Médiathèque de l'Ircam.

Rapport IRCAM 1995, consultabile presso la Médiathèque de l'Ircam.

MARTIN KALTENECKER, "Fausto Romitelli en son temps", in «Circuit», v. 24 n. 3, 2014, pp. 9-20.

SONYA E. KEENE, *Object-Oriented Programming in Common Lisp: A Programmer's Guide to CLOS*, Addison-Wesley, 1989.

ERNST KRENEK, "Extents and Limits of Serial Techniques", in «The Musical Quarterly», v. 46, n. 2, pp. 210-232.

JULIA KURSELL, ARMIN SCHÄFER, "Microsound and Macrocosm: Gérard Grisey's Explorations of Musical Sound and Space", in Yael Kaduri (a cura di), *The Oxford Handbook of Sound and Image in Western Art*, Oxford, Oxford University Press, 2016, pp. 192-213.

OTTO LASKE, "Composition Theory in Koenig's Project One and Project Two", in «Computer Music Journal», v. 5, n. 4, MIT-Press, 1981, pp. 54-65.

ÉMILE LEIPP, *Acoustique et Musique, données physiques et technologiques, problèmes de l'audition des sons musicaux, principes de fonctionnement et signification acoustique des principaux archétypes d'instruments de musique, les musiques expérimentales, l'acoustique des salles*, Parigi, Masson, 1984.

SERGE LEMOUTON, ALAIN BONARDI, RAFFAELE CIAVARELLA, “Peut-on envisager une organologie des instruments virtuels de l’informatique musicale ?”, in *Proceeding of the Fifth Conference on Interdisciplinary Musicology (CIM’09)*, 2009. <<http://architexte.ircam.fr/textes/Lemouton09b/index.pdf>>, ultima visita 5 marzo 2017.

VINCENZO LOMBARDO, ANDREA VALLE, *Audio e multimedia*, Santarcangelo di Romagna, Apogeo, 2014.

DENIS LORRAIN, *Fausto Romitelli EnTrance_2005 Performance Handbook*, IRCAM, Parigi, 2005.

ERIC MAESTRI, *La formation du style de Fausto Romitelli, lecture de ses quatre premières œuvres à travers le filtre de la texture et de la figure*, tesi di Master Recherche sotto la direzione di Alessandro Arbo, Université de Strasbourg, 2009.

ERIC MAESTRI, “*Natura morta con Fiamme* de Fausto Romitelli. La musique mixte et l’analyse aurale”, in *Corpus et méthodes : traductions théoriques de l’hétérogénéité musicale*, atti del colloquio internazionale Journées d’Analyse Musicale 2013 de la SFAM, Université de Rennes II, Sampzon, Delatour, 2013. <https://www.academia.edu/4460470/_Natura_morta_con_fiamme_de_Fausto_Romitelli._La_musique_mixte_et_la_analyse_aurale_>, ultima visita 5 marzo 2017.

CLAUDY MALHERBE, “Vertiges informatiques”, in «Entretemps», n. 10, 1992, pp. 117-131.

MIKHAIL MALT, “La composition assistée par ordinateur”, <<http://musinfo.fr/archives/anciensite/texte/Malt-I.pdf>>, ultima visita 5 marzo 2017.

MIKHAIL MALT, CURTIS ROADS, *PatchWork – Reference*, Parigi, IRCAM, 1996.

MIKHAIL MALT, CURTIS ROADS, *PatchWork – Introduction*, Parigi, IRCAM, 1996.

MIKHAÏL MALT, LAURENT POTTIER, *PW Csound/Edit-sco – Library of Modules for Generating Csound scores – reference*, Parigi, Ircam, 1996.

PETER MANNING, *Electronic and Computer Music*, New York, Oxford University Press, 2004.

MAX MATHEWS, “The Digital Computer as a Musical Instrument”, in «Science», v. 142 n. 3592, 1963, pp. 553-557.

MAX MATHEWS, “An Acoustic Compiler for Music and Psychological Stimuli”, in «The Bell System Technical Journal», v. 40, n. 3, 1961, pp. 667-694.

MAX MATHEWS, JOAN E. MILLER, E.E. DAVID JR, “Pitch Synchronous Analysis of Voiced Sounds”, in «The Journal of the Acoustical Society of America», v. 33 n. 2, 1961, pp. 179-186.

- JOHN A. MAURER, "A Brief History of Algorithmic Composition", 1999, <<https://ccrma.stanford.edu/~blackrse/algorithm.html>>, ultima visita 5 marzo 2017.
- MARCO MAZZOLINI, "The String Quartet in Italy After 1945", «Contemporary Music Review», v. 32 n. 4, 2013, pp. 353-417.
- Pactes faustiens. L'hybridation des genres musicaux après Romitelli*, «Circuit», v. 24 n. 3, 2014.
- PETER MCCALLUM, "An Interview with Pierre Boulez", in «The Musical Times», v. 130 n. 1751, 1989, pp. 8-10.
- PIERRE MICHEL, "Professor Bad Trip (Lessons I, II, III)", in Alessandro Arbo (a cura di), *Le corps électrique. Voyage dans le son de Fausto Romitelli*, Parigi, L'Harmattan, 2005, pp. 51-77.
- TRISTAN MURAIL, "La révolution des sons complexes", in *Darmstädter Beiträge zur neuen Musik*, XVIII, Mainz, Schott, 1980, poi raccolto in Tristan Murail, *Modèles & Artifices*, textes réunis par Pierre Michel, Strasbourg, Presses universitaires de Strasbourg, 2004.
- TRISTAN MURAIL, "Spectre et lutins", in TRISTAN MURAIL, *Modèles & Artifices*, textes réunis par Pierre Michel, Strasbourg, Presses universitaires de Strasbourg, 2004.
- TRISTAN MURAIL, *Modèles & Artifices*, textes réunis par Pierre Michel, Strasbourg, Presses universitaires de Strasbourg, 2004.
- TRISTAN MURAIL, "After-thoughts", in «Contemporary Music Review», v. 24, n. 2/3, 2005, pp. 269-272.
- PETER NORVIG, *Paradigms of Artificial Intelligence Programming: Case Studies in Common Lisp*, Morgan Kaufmann Publishers, San Mateo, California, 1992.
- ANGELO ORCALLI, "La bellezza dell'invisibile", in Gérard Grisey: *Intonare la luce. 25° Festival di Milano Musica – Percorsi di musica d'oggi*, Milano, Milano Musica – Teatro alla Scala, 2016, pp. 33-43.
- ANGELO ORCALLI, "La pensée spectrale", in Nicolas Donin, Laurent Feneyrou (a cura di), *Théories de la composition musicale au XXe siècle. Volume 2*, Lione, Symetrie, 2013, pp. 1511-1573.
- ANGELO ORCALLI, "Traces sonores du XXe siècle. Pour une critique des sources audiovisuelles", in Evelyne Gayou (a cura di), *Musique et Technologique. Préserver – Archiver – Re-produire*, Parigi, INA, 2013, pp. 33-74.
- ANGELO ORCALLI, *Fenomenologia della musica sperimentale*, Potenza, Sonus, 1993.
- JOHN R. PIERCE, *The Science of Musical Sound*, New York, W.H. Freeman & Co, 1992.

JOHN R. PIERCE, *Symbols, signal and noise: The Nature and Process of Communication*, New York, Harper & Brothers, 1961.

RICHARD C. PINKERTON, "Information Theory and Melody", in «Scientific American», v. 194/2, 1956, pp. 77-87.

LAURENT POTTIER, "La « régénération » des sons de Turenas de John Chowning", in Evelyne Gayou (a cura di), *Musique et Technologie. Préserver - Archiver - Re-produire*, Parigi, INA, 2013, pp. 145-196.

Laurent Pottier (a cura di), *Le calcul de la musique*, Saint-Étienne, Publications de l'Université Saint-Étienne, 2009.

LAURENT POTTIER, *Le contrôle de la synthèse sonore informatique. Cas particulier: l'utilisation du programme Patchwork*, tesi con la supervisione di Marc Battier presso l'École des hautes études en sciences sociales de Paris, 2001.

LAURENT POTTIER, *Patchwork. Csound/Edit-sco. Library of Modules for Generating Csound scores. Reference*, Parigi, IRCAM – Centre Georges Pompidou, 1996.

LAURENT POTTIER, *Patchwork. SpData*, Parigi, IRCAM, 1997.

Henri Pousseur (a cura di), *La musica elettronica*, Milano, Feltrinelli, 1976.

J. K. RANDALL, "A Report from Princeton", in «Perspectives of New Music», v. 3 n. 2, 1965, pp. 84-92.

E.G. RICHARDSON, "The Transient Tones of Wind Instruments", in «The Journal of the Acoustical Society of America», v. 26 n. 6, 1954, pp. 960-962.

JEAN-CLAUDE RISSET, DAVID L. WESSEL, "Exploration of Timbre by Analysis and Synthesis", in Diana Deutsch (a cura di), *Psychology of Music*, New York-Londra, Academic Press, 1982, pp. 25-58.

JEAN-CLAUDE RISSET, "Musique, calcul secret ?", in «Critique», n.359, Paris, 1977, pp. 414-429.

JEAN-CLAUDE RISSET, MAX MATHEWS, "Analysis of Musical-Instruments Tones", in «Physics Today», v. 22 n. 2, 1969, pp. 23-30.

CURTIS ROADS, *The Computer Music Tutorial*, Cambridge-Londra, The MIT Press, 1996.

ANTONIO RODÀ, "Ri-mediazione e ricostruzione di un tessuto sonoro: il restauro del nastro da concerto della *Fabbrica illuminata* di Luigi Nono", in Luca Cossettini (a cura di), *Luigi Nono. Studi, edizione, testimonianze*, Lucca, LIM, 2010, pp. 189-212.

XAVIER RODET, PIERRE COINTE, *FORMES – composition et ordonnancement de processus*, Rapport Ircam 36/85, Parigi, IRCAM, 1985.

XAVIER RODET, YVES POTARD, JEAN-BAPTISTE BARRIERE, “The CHANT Project: From the Synthesis of the Singing Voice to the Synthesis in General”, «Computer Music Journal», v. 8 n. 3, 1984, pp. 15-31.

FRANÇOIS ROSE, “Introduction to the Pitch Organization of French Spectral Music”, in «Perspectives of New Music», v. 34 n. 2, 1996, pp. 6-39.

GILBERT ROUGET, *La musique et la transe*, Parigi, Gallimard, 1980.

KAJJA SAARIAHO, “Timbre and harmony: Interpolations of timbral structures”, in «Contemporary Music Review», v. 2 n. 1, 1987, pp. 93-133.

Vincenzo Santarcangelo (a cura di), *Have your trip. La musica di Fausto Romitelli*, Auditorium edizioni, 2014.

PIERRE SCHAEFFER, *Traité des objets musicaux*, Parigi, Editions du seuil, 1966.

ALBRECHT SCHNEIDER, “Change and Continuity in Sound Analysis: A Review of Concepts in Regard to Musical Acoustics, Music Perception, and Transcription”, in Rolf Bader (a cura di), *Sound – Perception – Performance*, Springer, 2013, pp. 71-111.

XAVIER SIRVEN, *Authenticité et accessibilité des archives électroniques. MUSTICA. Le cas de la création musicale numérique*, IRCAM internship report, 2004. <<http://polaris.gseis.ucla.edu/blanchette/papers/RapportSirven.pdf>>, ultima visita 5 marzo 2017.

GUY L. STEELE, *Common Lisp the Language*, Newton (MA), Digital Press, 1984.

GUY L. STEELE, RICHARD P. GABRIEL, “The Evolution of Lisp”, *History of Programming languages II*, New York, 1996, pp. 233-330.

BERNARD STIEGLER, “Bouillonnements organologiques et enseignement musical”, in «Les dossiers de l'ingénierie éducative», 43, Paris, CNDP, 2003, pp. 11-15.

KARLHEINZ STOCKHAUSEN, “Notes on Mixtur (1964)”, in «Electronic Music Review», n. 1, 1967, pp. 18-21.

HEINRICH TAUBE, “Common Music: A Music Composition Language in Common Lisp and CLOS”, in «Computer Music Journal», v. 15 n. 2, 1991, pp. 21-32.

HEINRICH TAUBE, TOBIAS KUNZE, “Capella: A Graphical Interface for Algorithmic Composition”, in *Proceedings of the 1995 International Computer Music Conference*, San Francisco, International Computer Music Association, 1995, pp. 377-380.

HEINRICH TAUBE, "An Introduction to Common Music", in *Computer Music Journal*, v. 21 n. 1, 1997, pp. 29-34.

LIVIA TEODORESCU-CIOCANEA, "Timbre versus spectralism", in «*Contemporary Music Review*», v. 22 n. 1-2, 2003, pp. 87-104.

JAMES TENNEY, "Sound-Generation by means of a Digital Computer", in «*Journal of Music Theory*», v. 7 n. 1, 1963, pp. 24-70.

BARRY TRUAX, "Computer Music Composition: The Polyphonic POD System", in «*IEEE Computer*», v. 11 n. 8, 1978, pp. 40-49.

FRANCK VARENNE, *Qu'est-ce que l'informatique ?*, Parigi, Vrin, 2009.

Giovanni Verrando (a cura di), *La nuova liuteria: orchestrazione, grammatica, estetica*, Milano, Suvini-Zerboni, 2012.

ANTOIN VINCENT, *Préservation d'œuvres musicales. Étude du processus de production*, tesi presso l'Université de Technologie de Compiègne, sotto la direzione di Bruno Bachimont e Alain Bonardi, 2010.

FRITZ WINCKEL, *Music, Sound and Sensation: A Modern Exposition*, New York, Dover, 1967.

IANNIS XENAKIS, *Formalized Music*, Pendragon, 1992.

Yamaha Music Synthesizer SY99. Manuale d'istruzioni.

LAURA ZATTRA, *Analyse de Stria de John Chowning*, <<http://brahms.ircam.fr/analyses/Stria/>>, ultima visita 5 marzo 2017.

LAURA ZATTRA, *Studiare la computer music*, Padova, Libreria Universitaria, 2011.

Scritti, note di sala e interviste di Fausto Romitelli

“Professor Bad Trip : presentation”, in Alessandro Arbo (a cura di), *Le corps électrique. Voyage dans le son de Fausto Romitelli*, Parigi, L’Harmattan, 2005, pp. 135-137.

“Résonance”, in Christine Buci Blucksman, Michaël Lévinas (a cura di), *L’idée musical*, Vincennes-Saint-Denis, Presses Universitaires de Vincennes, 1993, pp. 43-45. L’articolo è stato poi ripreso in Alessandro Arbo (a cura di), *Le corps électrique. Voyage dans le son de Fausto Romitelli*, Parigi, L’Harmattan, 2005, pp. 127-129.

“Il compositore come virus”, in Alessandro Arbo (a cura di), *Il corpo elettrico. Viaggio nel suono di Fausto Romitelli*, Monfalcone, Teatro comunale di Monfalcone, 2003, pp. 81-84.

Progetto di ricerca inviato all’IRCAM nel 2003, in ERIC MAESTRI, *La formation du style de Fausto Romitelli, lecture de ses quatre premières œuvres à travers le filtre de la texture et de la figure*, tesi di Master Recherche sotto la direzione di Alessandro Arbo, Université de Strasbourg, 2009.

Progetto di ricerca elaborato durante il suo soggiorno all’IRCAM, in ALESSANDRO ARBO, “Risonanze e anamorfofi. Note sulla poetica di Fausto Romitelli”, in Amalia Collisani, Gabriele Garilli, Gaetano Mercadante, *Italia/Francia. Musica e cultura nella seconda metà del XX secolo*, Palermo, L’Epos, 2009, pp. 343-345.

Programma di sala per la prima esecuzione di *EnTrance*, il 26 gennaio 1996.
<<http://brahms.ircam.fr/works/work/11518/>>, ultima visita 5 marzo 2017.

Presentazione per *La sabbia del tempo* (1991/92 – nuova versione) e per *Les idoles du soleil* (1992), in ALESSANDRO ARBO, “Risonanze e anamorfofi. Note sulla poetica di Fausto Romitelli”, in Amalia Collisani, Gabriele Garilli, Gaetano Mercadante, *Italia/Francia. Musica e cultura nella seconda metà del XX secolo*, Palermo, L’Epos, 2009, pp. 341-342.

Presentazione di *Mediterraneo I – Les idoles du soleil* e *Mediterraneo II – L’Azur des Déserts*, in ALESSANDRO ARBO, “Risonanze e anamorfofi. Note sulla poetica di Fausto Romitelli”, in Amalia Collisani, Gabriele Garilli, Gaetano Mercadante, *Italia/Francia. Musica e cultura nella seconda metà del XX secolo*, Palermo, L’Epos, 2009, pp. 342-343.

Programma di sala per la prima esecuzione di *Natura morta con fiamme*, Parigi, IRCAM, 1992.

Presentazione di *Natura morta con fiamme*, in occasione del festival Agora 2000, <http://medias.ircam.fr/x28b1e0_natura-morta-con-fiamme-presentation>, ultima visita 5 marzo 2017.

Presentazione di *Natura morta con fiamme*, in occasione della prima esecuzione dell'opera.
<http://medias.ircam.fr/embed/media/x371b59_presentation-de-loeuvre-natura-morta-con>, ultima visita 5 marzo 2017.

“Pour une pratique visionnaire”, in DANIELLE COHEN-LÉVINAS, *Causeries sur la musique. Entretiens avec des compositeurs*, Parigi, L'Harmattan, 1999, p. 287-294.

“Attacquons le réel à sa racine. Entretien de Fausto Romitelli”, in Danielle Cohen-Lévinas (a cura di), *La création après la musique contemporaine*, Parigi, L'itinéraire/L'Harmattan, 1999, pp. 87-90.

OMER CORLAIX, “L'insurgé. Entretien avec Fausto Romitelli”, in «Musica Falsa», n. 11, 2000, pp. 84-85.

VÉRONIQUE BRINDEAU, “Entretien avec Fausto Romitelli”, in «Accents. Le journal de l'Ensemble Intercontemporain», n. 15, 2001, pp. 6-7.