

Suoni di sintesi e routing di segnali midi & audio in *Csound*  
e *MaxMSP*

*Ambiente integrato per la produzione di musica elettronica/informatica*

Stefano Cucchi - Numero di matricola 4258

*Corso di laurea specialistica in "Tecnologia del suono"*

Conservatorio G. Verdi - Milano

Anno accademico 2012/2013

## Introduzione

*Csound*<sup>1</sup> e *MaxMSP*<sup>2</sup> sono due tra i principali software per la produzione di musica elettronica e consentono entrambi la quasi totalità delle tecniche di sintesi sia in tempo reale che differito, l'elaborazione live di segnali audio ed una completa implementazione del protocollo MIDI.

I due software hanno però una filosofia completamente diversa: Csound funziona tramite linea di comando, MaxMSP tramite la connessione sul piano di lavoro di oggetti con funzioni differenti.

Ciascuno dei due approcci ha pregi e difetti, sia oggettivi (derivati cioè dalla concezione stessa del software) che soggettivi (derivati cioè dalle preferenze dell'utente).

Nel mio percorso di studi ho esplorato sia Csound che MaxMSP e ho provato a ipotizzare una piattaforma di lavoro che mi consentisse l'uso contemporaneo ed interconnesso dei due software sia riguardo allo scambio di dati midi (sia generati dai software stessi che da controller esterni) che audio, nonché l'utilizzo di file audio generati durante il brano stesso.

Sono riuscito così a costruire un *personale* sistema integrato di produzione di musica elettronica/informatica che mi dà la possibilità di sfruttare al meglio i due software secondo le peculiarità oggettive degli stessi e le mie esigenze.

Ho studiato inoltre la gestione di un sistema di sincronizzazione e una time line comune che consente, nella fase di studio/composizione, l'ascolto di singole parti del brano.

Questo sistema integrato è possibile in linea teorica sulle principali piattaforme informatiche: su Windows con la sostituzione di Soundflower (non presente per questa piattaforma) con un software di routing di dati audio; su Linux utilizzando PD al posto di MaxMSP e Jack al posto di Soundflower.

Ho testato il sistema con le seguenti periferiche:

- Machintosh<sup>3</sup>: MacBook Pro, Processore 2,4 Ghz Intel Core i5, OS Mac OS X Lion 10.7.5
- Csound: Csound5 - QuteCsound 0.6.0
- MaxMSP 6.0.7
- Scheda audio: M-Audio Fast Track C400

La trattazione teorica è accompagnata dalla presentazione di un brano - di natura invero molto didascalica - suddiviso in sezioni in cui, per ciascuna di esse, vengono presentate e attuate le varie tecniche di scambio dati tra i software, e l'acquisizione di dati midi esterni; ogni sezione è caratterizzata inoltre da un tipo particolare di sintesi sonora.

---

<sup>1</sup>Csound © Copyright 1986, 1992 Massachusetts Institute of Technology.

<sup>2</sup>MaxMSP © Copyright Cycling '74.

<sup>3</sup>TM e © 1983-2012 Apple Inc.

## 1 Software e connessioni.

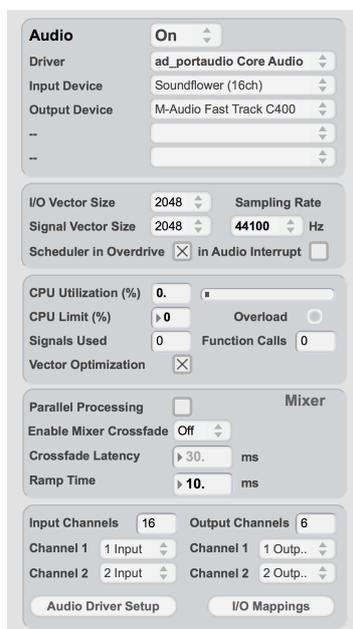
Il flusso dei dati audio tra *MaxMSP* e *Csound* avviene tramite il software *Soundflower*<sup>4</sup>. Quest'ultimo è un driver audio virtuale che si comporta esattamente come una periferica reale (una scheda audio), con i propri input ed output (2 o 16 canali).<sup>5</sup>

Il sistema prevede che i segnali audio generati da *Csound* vengano mandati [output device (-o)] a *MaxMSP* su 16 canali separati e poi inviati alla scheda audio in quadrifonia.

I segnali audio in ingresso in *Csound* [Input device (-i)] possono provenire o dalla scheda audio esterna o dai microfoni integrati: possono essere processati direttamente in *Csound* o inviati a *MaxMSP* per essere elaborati in *MaxMSP*.

Perchè ciò sia possibile è però necessario che i canali di ingresso della interfaccia audio corrispondano al numero di canali (opcode *nchnls*) della orchestra di *Csound*.<sup>6</sup> I segnali audio generati in *MaxMSP* e quelli provenienti da *Csound* vengono poi mandati in uscita alla scheda audio.

Figura 1: Impostazioni audio in MaxMSP.



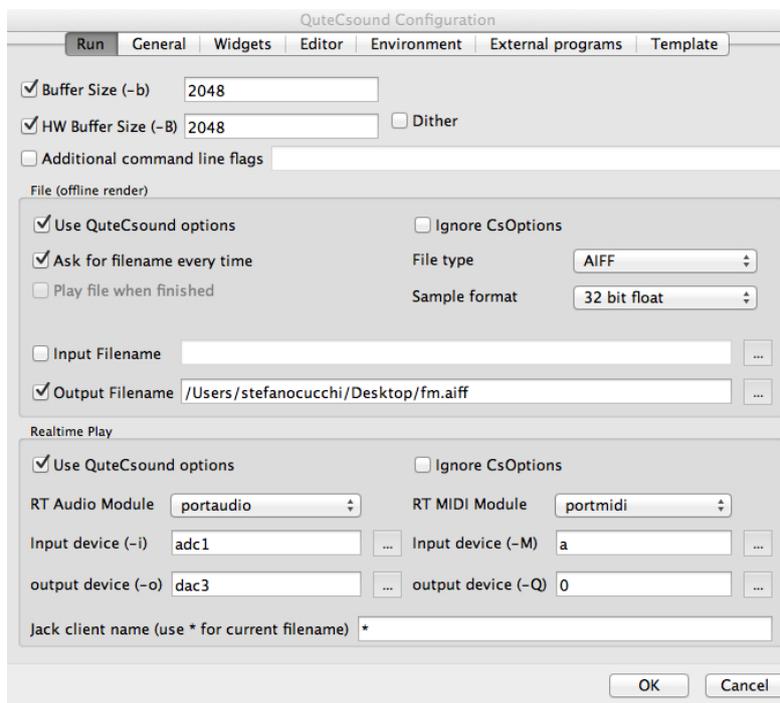
Sono altresì possibili, tramite le impostazioni di sistema, configurazioni di routing audio più complesse, quali “Dispositivi con uscite multiple” e “Dispositivi combinati”

<sup>4</sup>Soundflower © Copyright Cycling '74.

<sup>5</sup>Nelle ultime versioni il software consente di avere fino a 64 canali audio.

<sup>6</sup>Durante la composizione del brano ho preferito utilizzare il routing di sedici canali audio sacrificando la possibilità di utilizzo di segnali audio ripresi “live” avendo la scheda audio utilizzata 4 canali audio in ingresso.

Figura 2: Impostazioni midi e audio in Csound.



che non rappresentano, almeno in questo caso, una reale utilità.<sup>7</sup>

Il flusso di dati midi è gestito dal dispositivo “core midi” in varie combinazioni:

- Da MaxMSP a Csound.
- Da Csound a MaxMSP.
- Da un controller esterno a ciascuno o entrambi i software.
- Dai software a uno/più strumenti esterni.

La contemporaneità di più flussi può essere controllata indirizzando in uscita e filtrando in entrata i dati MIDI su canali differenti, tramite gli oggetti *midiin*, *midiselect* e *route* in MaxMSP e gli opcode *massign*, *moscil* e *midion* in Csound.

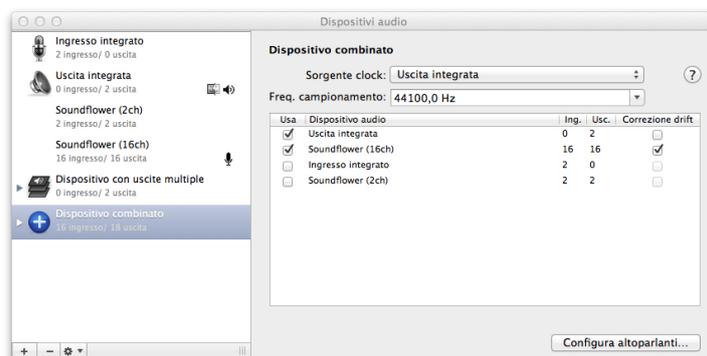
Gerarchia dei software per la sincronizzazione: Csound - master & MaxMSP - slave.

In una composizione la cui esecuzione sia il risultato del processamento operato da due software distinti è fondamentale garantire la sincronia degli stessi.

---

<sup>7</sup>É possibile ad esempio avere segnali audio in ingresso simultaneo sia dai microfoni integrati che dalla scheda audio, oppure segnale audio in uscita su diversi dispositivi (virtuali o reali).

Figura 3: Dispositivi audio.



La simultaneità non è possibile inviando un unico segnale di “start” a Csound e MaxMSP in quanto Csound prevede un tempo seppur minimo per le impostazioni del file e la compilazione dell’orchestra<sup>8</sup>, in MaxMSP invece, una volta attivato l’audio, è sufficiente un “bang” per iniziare i processi praticamente “real-time”.

La soluzione trovata è stata quella di mandare da Csound a MaxMSP un comando “bang” come primo evento della score.

L’effettiva sincronia è altresì influenzata dai “Buffer Size (-b)” e dal “HW Buffer Size (-B)” in Csound e al “I/O Vector Size” e al “Signal Vector Size” in MaxMSP; il ritardo teorico del segnale proveniente da Csound non è comunque avvertibile con Buffer di 2048 campioni in Csound e I/O vector Size di 2048 campioni in MaxMSP.

Una volta avviato il “clocker” in MaxMSP la sincronia viene mantenuta per tutta la durata del brano. Il segnale di start viene mandato attraverso una nota MIDI; all’interno del brano, altri segnali aggiuntivi di sincronia vengono mandati per permettere un playback a sezioni.

In sostanza la *time line* principale è quella di Csound a cui sono agganciati dei “bang” in MaxMSP in punti strategici del brano.<sup>9</sup>

## 2 Lavori preliminari alla composizione del brano.

Prima della composizione del brano ho preparato alcuni strumenti utili per i miei procedimenti quali, fogli excel per il calcolo della time line, strumenti di Csound per la spazializzazione, preparazione di tabelle Csound e patch di MaxMSP, etc...

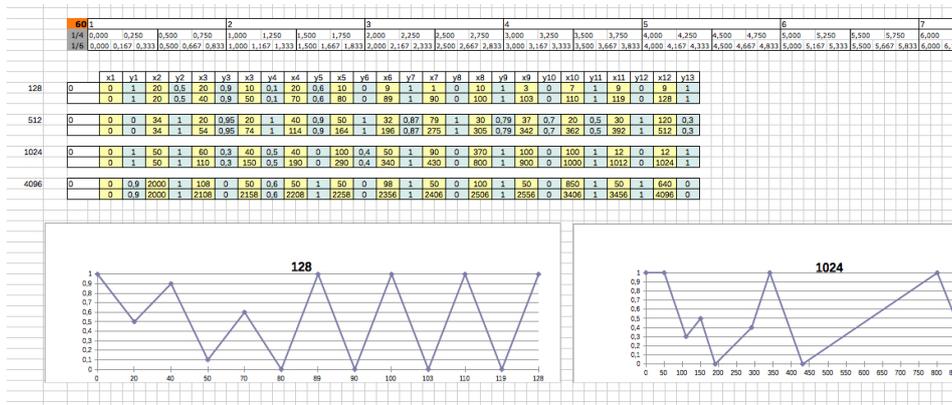
Alcune delle patch di MaxMSP sono state inserite nel progetto finale, come del resto la maggior parte delle subpatch sono state sviluppate dapprima come patch a sè stanti da unire in un momento successivo al progetto generale.

Stesso procedimento è stato usato con Csound attraverso la preparazione degli strumenti

<sup>8</sup>È da tener presente che il tempo necessario alla compilazione dell’orchestra varia durante il proseguo della composizione in relazione al numero di strumenti, alla loro complessità, ai file collegati, etc...

<sup>9</sup>Ho anche ipotizzato un segnale midi fisso ogni secondo, ipotesi perfettamente attuabile, ma non necessaria in questo caso in cui è stato sufficiente una suddivisione a sezioni.

Figura 4: Foglio excel per tabelle e timeline.



in progetti singoli poi riuniti in un unico progetto; ciò consente una lettura più agevole della orchestra e della score nella fase di “liuteria sonora” e sound design.

### 3 Il brano.

Il brano è diviso in sezioni ciascuna delle quali esplora una sintesi diversa, una diversa configurazione del flusso dati audio e MIDI e un diverso controllo da parte dell’esecutore di alcuni parametri della produzione sonora: preset, spazializzazione, volumi, effetti, registrazione di file su hard disk, etc. . .

Il brano è composto *INTEGRALMENTE* con suoni di sintesi, le tecniche usate sono:

- Sintesi sottrattiva (rumore bianco filtrato).
- Modulazione di frequenza - modulazione di fase.
- Sintesi granulare.
- Algoritmi di modellazione fisica (Karplus Strong, opcode *vibes* in Csound).
- Convoluzione.

Il brano è stato pensato per una diffusione quadrifonica, avendo in alcuni casi la spazializzazione valenza timbrica e strutturale. La composizione prevede la generazione di eventi pseudo-casuali<sup>10</sup> nonché utilizzo di segnali di controllo generati random in tempo reale. Si ha così un diverso risultato ad ogni esecuzione del brano.

L’esecutore opera tramite tastiera del PC nella seconda sezione (cambio di preset in MaxMSP) e tramite controller MIDI esterno nella terza sezione (spazializzazione in Csound).

<sup>10</sup>Preferisco definirla *generazione pseudo-casuale* e non *generazione algoritmica* in quanto ho lavorato per “sottrazione” restringendo in maniera empirica, successivamente ad ogni ascolto, le possibilità di scelta riguardo a parametri quali altezze, durate, velocity, etc. . . e non definendo in via preliminare delle classi di altezze, delle combinazioni o delle successioni.

Figura 5: Patch MaxMSP per function.

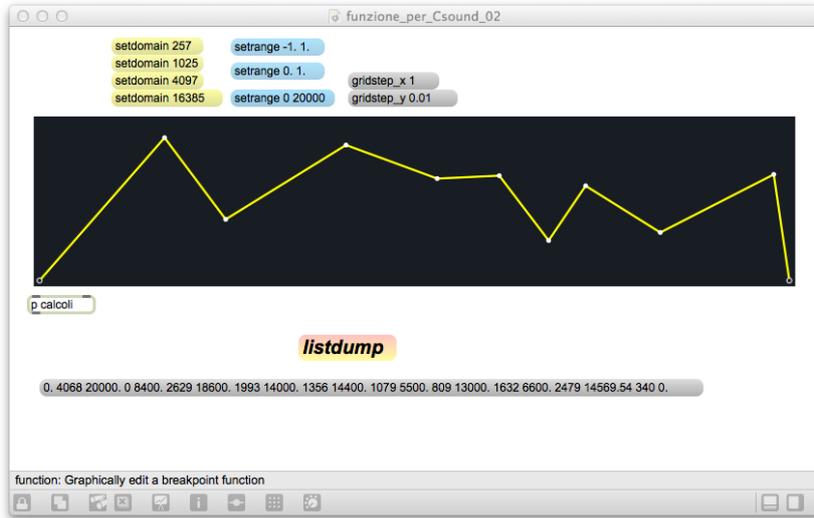


Figura 6: Studio per movimenti Csound.

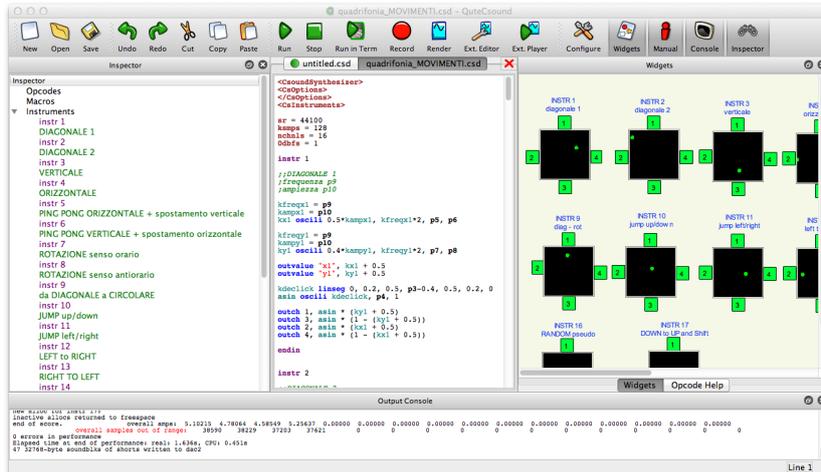


Figura 7: Time line MaxMSP.

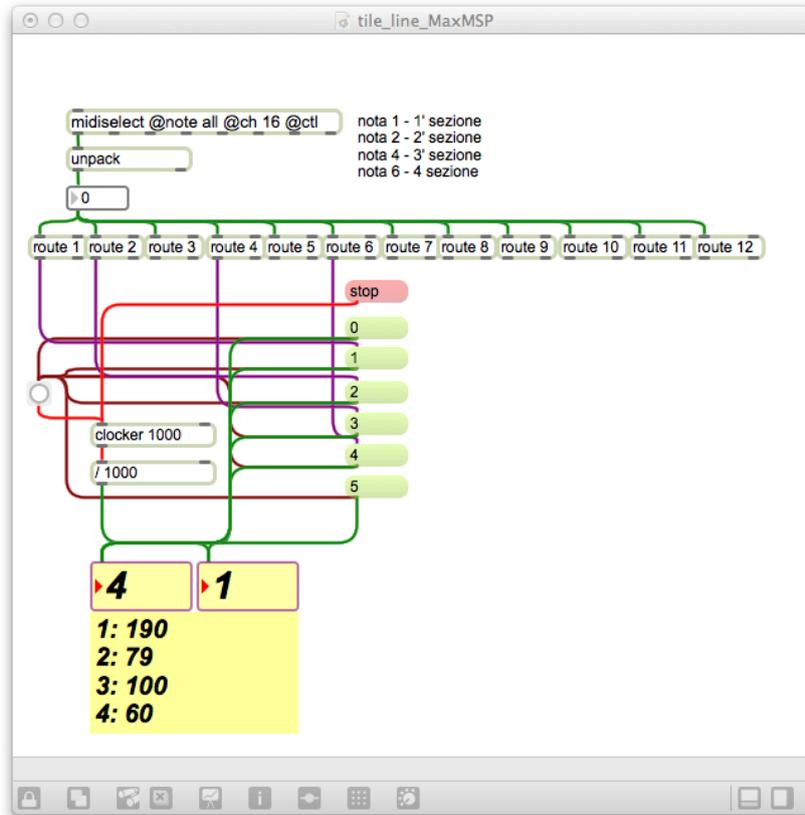


Figura 8: Controllo MaxMSP tramite tastiera.

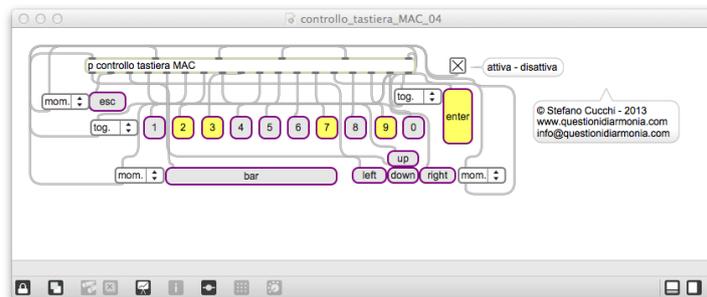
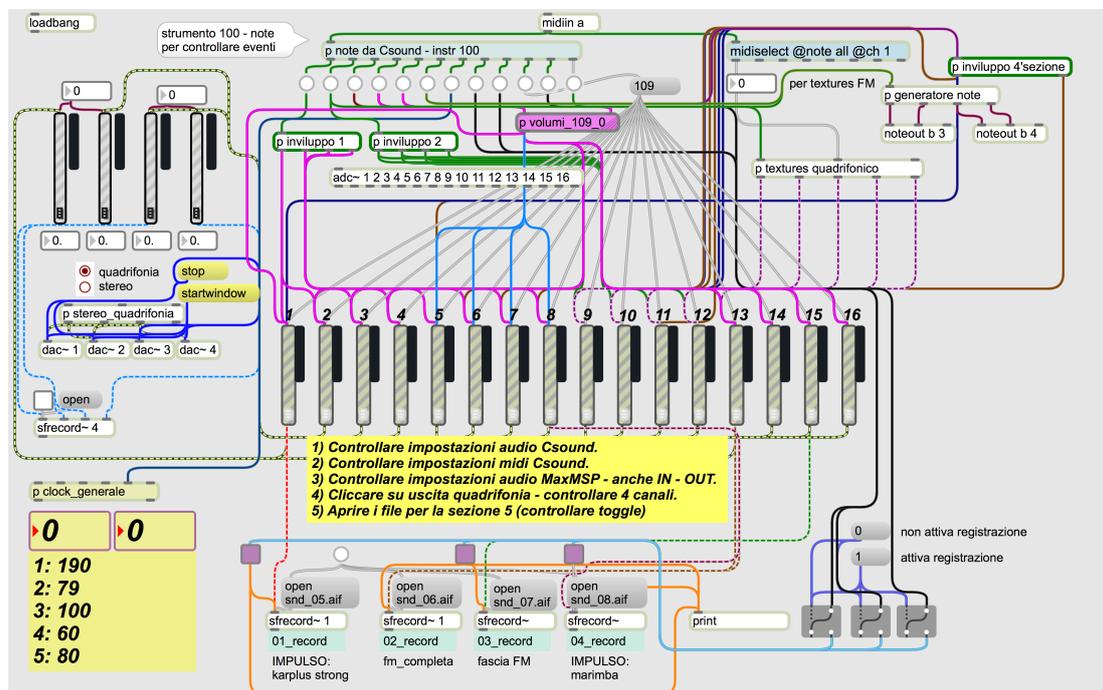


Figura 9: Patch generale MaxMSP.



### Suggerzioni compositive.

Il brano è nato dalle suggestioni derivatemi dalla lettura e dall'ascolto di un madrigale a cinque voci di Gesualdo da Venosa tratto dal libro quinto.

Carlo Gesualdo principe di Venosa (1566 - 1613): *“Languisce al fin”*.

*Languisce al fin  
 Chi da la vita parte  
 E di morte il dolore l'affligge  
 Sì che in crude pene more  
 Ahì che quel son io  
 Dolcissimo cor mio  
 Che da voi parto  
 E per mia crudele sorte la vita lascio  
 E me ne vado a morte.*

Ho voluto sviluppare e trasporre in immagini sonore due concetti cardine della vita: la *pulsazione - battito cardiaco* e il *soffio vitale - respiro*.

Non si tratta di musica a programma o musica descrittiva, quanto più di *musica evocativa*: fasce sonore e bande di rumore rappresentano il soffio vitale/respiro, suoni impulsivi/percussivi indicano la pulsazione. Questi due elementi si trovano, in linea prioritaria giustapposti, in un caso (terza sezione) l'utilizzo della sintesi granulare e del riverbero

permetterà di passare da impulsi/grani a fasce sonore.

Si è cercato inoltre di esplorare e di mettere in luce gli aspetti contigui e non solo opposti/complementari delle due categorie di “oggetti sonori”.<sup>11</sup>

### 3.1 Sezione 1: algoritmo di Karplus-Strong e sintesi sottrattiva.

Uno degli aspetti più complicati della “liuteria elettronica” in Csound è la scrittura di instrument che richiedono numerosi parametri nelle score. Le soluzioni possono essere di vario tipo: dividere il processo in più strumenti successivi, oppure fare un rendering e elaborare il materiale (effetti e spazializzazione) in un passo successivo.

Ho scelto la seconda opzione lasciandomi comunque la possibilità poter modificare il suono prima della esecuzione finale conservando in un file separato gli strumenti e la score originaria.

Questa sezione utilizza due processi di generazione sonora: rumore bianco filtrato passa banda e algoritmo Karplus-Strong. Aspetto fondamentale nella creazione di suoni tramite procedimenti informatici è la “Acoustic viability”: nel filtro passa banda ho fatto un largo uso di opcode *randi* e *jitter* collegati alla frequenza del centro banda e della larghezza di banda, si riesce in questo modo a evitare la staticità di un filtro digitale e del generatore di rumore.<sup>12</sup>

La larghezza di banda è stata tenuta stretta fino ad avere un risultato vicino al suono intonato ma con una certa indeterminatezza.

Anche per l'algoritmo di Karplus-Strong ho preso un rumore bianco come eccitazione iniziale. Ho determinato in via preliminare un'unica nota-frequenza con minime variazioni casuali di intonazione mentre, cambiando la lunghezza del buffer si passa in maniera graduale da un effetto suono/rumore alla simulazione di una corda pizzicata.

L'effetto di delay è ottenuto sfruttando una caratteristica intrinseca dell'algoritmo: aumentando la grandezza del buffer.

Le tracce singole ottenute tramite rendering sono state distribuite sui 4 canali, minime variazioni casuali della velocità di lettura dei file danno vita un effetto di chorus molto leggero.

Attraverso il routing audio tra strumenti Csound (*Zack patch system*) è stata creata una mandata per aggiungere un effetto di risonanza tramite l'opcode *streson*, che simula la risonanza di una corda con un procedimento simile all'algoritmo di Karplus-Strong. La frequenza della risonanza varia in maniera pseudo casuale per ciascuno dei 4 canali, ottenendo così “pennellate” di effetto diverse nei rispettivi punti di diffusione.

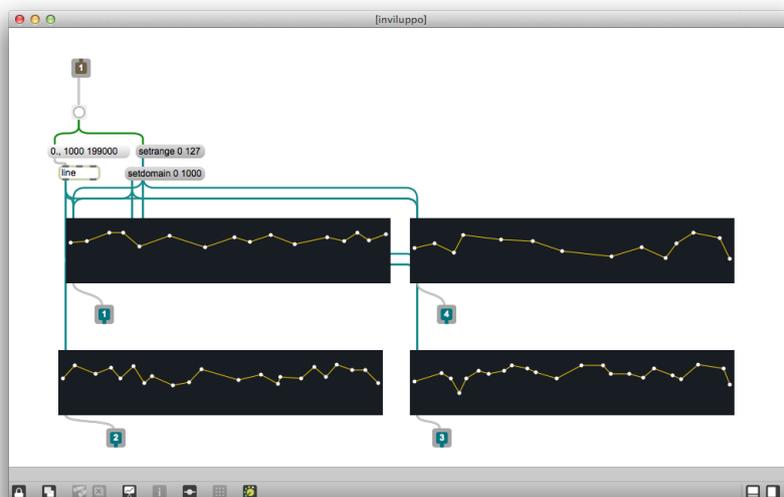
Il mixaggio vero e proprio è stato ottenuto creando 4 *function* in MaxMSP che controllano i volumi dei canali audio.

---

<sup>11</sup>Con il termine *oggetti sonori* non intendo riferirmi alla catalogazione di Shaffer ma ai due caratteri generali di *impulso* e *fascia sonora*.

<sup>12</sup>È percepibile ad un ascolto attento una certa “ripetitività” del rumore bianco generato dagli opcode *rand* e *noise* in Csound. In ambiente informatico un evento “casuale” non è mai completamente decorrelato.

Figura 10: Sezione 01: involuppi.



### 3.2 Sezione 2: sintesi FM-modulazione di fase & segnali midi da Csound a MaxMSP.

La seconda sezione usa esclusivamente la tecnica della *modulazione di fase* in Csound e della *modulazione di frequenza e di fase* in MaxMSP<sup>13</sup>.

In Csound la modulazione di fase è stata implementata in maniera classica tramite la coppia di opcode *phasor & tablei*. La forma d'onda del *carrier* è una sinusoida, la forma d'onda (tempo variante) del *modulator* è ottenuta tramite il morphing di tabelle contenenti forme d'onda diverse (GEN10: sinusoidi con parziali superiori in rapporto armonico con la fondamentale).

Lo strumento produce segnali con leggere varianti sui 4 canali: le differenze sono ottenute tramite variazioni casuali delle frequenze, sia del carrier che del modulator come del volume del modulator.

Analogamente alla prima sezione, dopo aver ottenuto i 4 file audio, gli stessi sono stati letti tramite l'opcode *diskin* con variazioni di velocità ottenuti con l'opcode *jitter* e una diversa combinazione di riverbero e delay.

In MaxMSP ho costruito un patch quadrifonico in cui 4 sinusoidi sono modulate in fase da altrettante sinusoidi, la fase delle sinusoidi modulante viene a sua volta riportata a 0 a intervalli regolabili.

La nota/frequenza del carrier è determinata da uno strumento Csound che genera uno stream di note midi tramite l'opcode *moscil*, le note vengono inviate da Csound sul canale midi 1 e su MaxMSP è stato messo un filtro (oggetto *midiselect*) che legge solamente le note in ricezione sul canale 1.

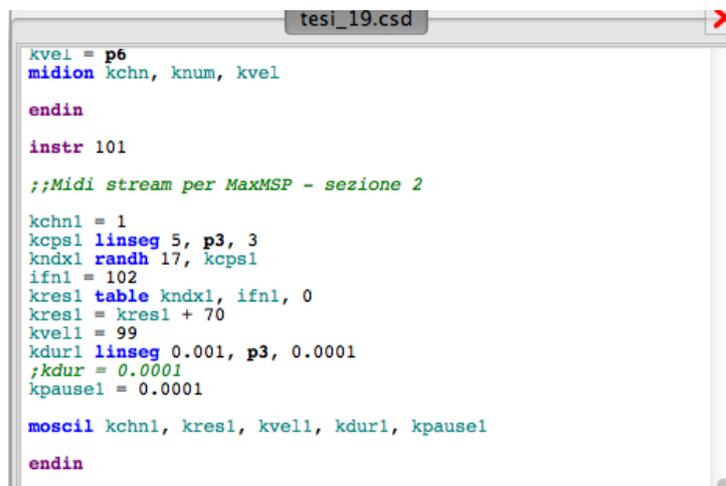
<sup>13</sup>In questo paragrafo si utilizzano termini quali *carrier* e *modulator*, mutuati dalla sintesi FM, anche per la descrizione della *modulazione di fase*, considerando la sostanziale equivalenza delle due tecniche.

Le frequenze del carrier e del modulator sono in rapporti interi tra di loro e il fattore di moltiplicazione è determinato dall'oggetto *drunk*. In ognuno dei 4 canali un carrier diverso ha frequenza multipla intera della nota/frequenza ricevuta da Csound: anche in questo caso le altezze sono determinate in modo pseudo-casuale da degli oggetti *random* non sincronizzati tra di loro.

Quattro preset sono selezionabili tramite i numeri 1, 2, 3 e 4 della tastiera del PC, il tasto 5 porta i volumi a 0 e può essere usato per disattivare lo strumento.

Al termine della sezione da Csound arriva un segnale midi che attiva il preset 5 spegnendo in maniera definitiva lo strumento e il controllo tramite tastiera.

Figura 11: Sezione 02: moscil.



```

kvel = p6
midion kchn, knum, kvel

endin

instr 101

;;Midi stream per MaxMSP - sezione 2

kchn1 = 1
kcps1 linseg 5, p3, 3
kndx1 randh 17, kcps1
ifn1 = 102
kres1 table kndx1, ifn1, 0
kres1 = kres1 + 70
kvel1 = 99
kdur1 linseg 0.001, p3, 0.0001
;kdur = 0.0001
kpause1 = 0.0001

moscil kchn1, kres1, kvel1, kdur1, kpause1

endin

```

### 3.3 Sezione 3: sintesi granulare e controllo tramite controller midi esterno.

In questa sezione sono presenti due strumenti che sfruttano la sintesi granulare implementata in Csound e controllati real-time da un controller midi esterno.

I suoni utilizzati sono:

1. Una *rampa* costruita con la *gen07* e granulazione tramite l'opcode *grain*.
2. Il file audio in FM della sezione 02 e granulazione tramite l'opcode *syncgrain*.

Il suono dello strumento 1 si muove nei 4 canali secondo delle funzioni specificate nella score i cui parametri sono controllati real-time da un controller MIDI<sup>14</sup>, il suono dry viene passato a uno strumento successivo, riverberato e a sua volta spazializzato sui 4 canali sempre tramite controller MIDI.

In questa sezione si può passare gradualmente dagli impulsi singoli a fasce sonore modificando il numero di grani per secondo, gli impulsi vengono trasformati a loro volta in fasce sonore con il riverbero.

I parametri controllati real-time dello strumento 1 sono:

<sup>14</sup>In questo caso ho usato un *KORG nanoKONTROL2*.



- Parametri comuni: numero di grani.
- Parametri comuni: deviazione in frequenza.
- Suono dry: velocità movimento asse X.
- Suono dry: ampiezza movimento asse X.
- Suono dry: velocità movimento asse Y.
- Suono dry: ampiezza movimento asse Y.
- Suono riverberato: velocità movimento asse X.
- Suono riverberato: ampiezza movimento asse X.
- Suono riverberato: velocità movimento asse Y.
- Suono riverberato: ampiezza movimento asse Y.

Il suono FM della seconda è stato effettuato con un riverbero e un flanger con impostazioni diverse per ciascuno dei 4 canali, il controllo real-time è il volume comune ai 4 canali. Grazie all'estensione “*Widgets*” presente in CsoundQt è possibile vedere sul monitor la posizione degli slider e dei knob nonché il movimento del suono nello spazio.

### 3.4 Sezione 4: Strumenti di Csound controllati da MaxMSP.

Nella sezione 4 gli strumenti di Csound sono suonati tramite comandi MIDI (nota on e nota off) inviati da MaxMSP.

Due algoritmi di generazione differenti in MaxMSP inviano note diverse su 2 differenti canali MIDI a 2 strumenti di Csound che filtrano i messaggi secondo il canale di provenienza.

I comandi vengono inviati da MaxMSP attraverso l'oggetto *noteout*; tramite gli argomenti *device* e *channel* specificiamo il “device” e il canale di trasmissione.

In Csound l'opcode *massign* assegna a ogni strumento uno specifico canale midi.

La densità delle note è descritta in una *function* che viene letta in maniera lineare, mentre le altezze inserite in una *table* vengono lette in maniera pseudocasuale.

Ho inoltre randomizzato la *velocity* come variazione rispetto a dei valori predefiniti.

Gli strumenti di Csound convertono il valore-nota in frequenze tramite tabelle definite dal compositore che contengono classi di altezze non temperate.

La non sincronia dei due strumenti e l'utilizzo aggiuntivo di un delay con tempo di ritardo diverso per ciascuno dei quattro canali danno vita a una “texture” polifonica e poliritmica.

I suoni-battiti terminano improvvisamente.

Figura 14: Sezione 04: generazione note MIDI.

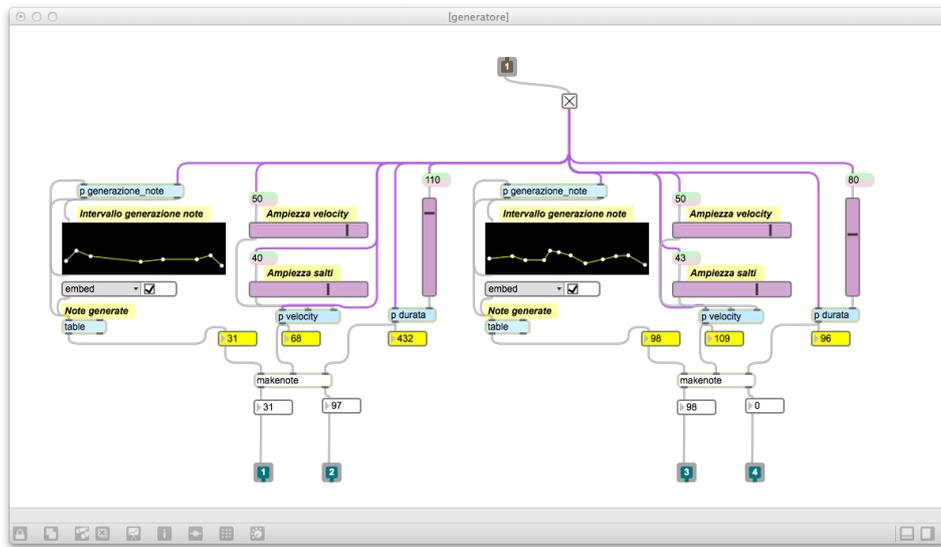
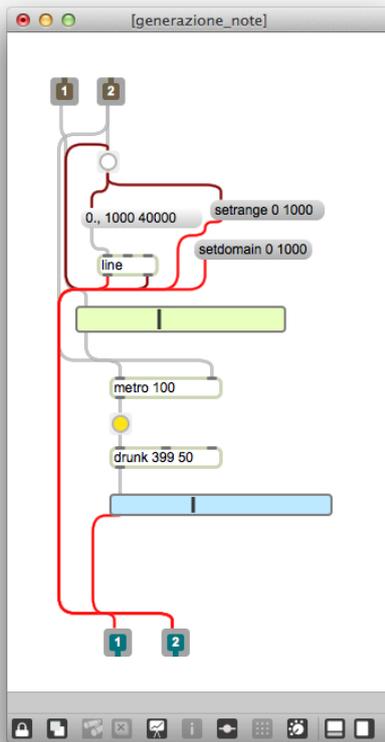


Figura 15: Sezione 04: lettura tabella altezze.



### 3.5 Sezione 5: Utilizzo da parte di Csound di file *.aiff* generati/registrati da MaxMSP nelle sezioni precedenti - sintesi per convoluzione.

Sia Csound che MaxMSP hanno la possibilità di registrare (scrivere su disco) file audio in diversi formati. In questa sezione si è voluto sperimentare l'utilizzo di file audio generati durante le sezioni precedenti del brano.

È necessario l'accorgimento, qualora si voglia utilizzare i file in Csound tramite la *GEN01* di inserire il flag addizionale *-D* nei *Command-line Flags*, ciò permette di ritardare il caricamento della forma d'onda nella *GEN01* fino al suo effettivo utilizzo da parte dello strumento. In caso contrario si rischia che il file non venga trovato o che ne venga utilizzata una versione precedente se la scrittura del file avviene durante la performance.<sup>15</sup> I comandi di “start” e di “stop” della registrazione vengono inviati da Csound o dall'esecutore direttamente dalla patch di MaxMSP, mentre il nome viene scelto manualmente in accordo col nome del file dell'orchestra Csound.

In MaxMSP tramite un *Gate Switch* è possibile decidere se attivare la registrazione o non attivarla utilizzando così i file generati nell'ultimo rendering del brano. Disattivando la registrazione automatica è possibile scegliere quali file registrare tramite comando manuale (attivando e disattivando l'apposito “toggle” nella patch di MaxMSP.)

Negli strumenti di Csound sono stati inseriti dei codici/commento per poter utilizzare versioni alternative dei file cancellando semplicemente alcuni caratteri nella score.

L'ultima sezione del brano rappresenta il riepilogo dei materiali utilizzati in precedenza, e proprio a sottolinearne il senso di ricapitolazione sono stati utilizzati elementi generati durante le sezioni precedenti del brano.<sup>16</sup> Alcune fasce sonore registrate durante sezioni 2 e 3 vengono sottoposte a convoluzione con impulsi registrati nelle sezioni 1 e 5. I comandi di “start” e di “stop” della registrazione degli impulsi per la convoluzione sono stati scelti con cura per essere sicuri di inserire un suono impulsivo nel file. I suoni ottenuti tramite convoluzione vengono fatti ruotare nello spazio con un movimento lento, dopo un crescendo nel registro grave si ha una breve linea acuta che si estingue rapidamente.<sup>17</sup>

## 4 Conclusioni.

Il sistema.

Al termine dello studio si è verificata la stabilità del sistema: MAC con processore Intel Core i5 a 2.4 GHz e 4 GB di RAM è al limite delle sue prestazioni.

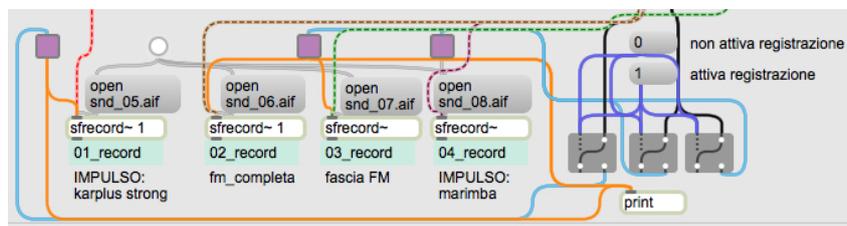
---

<sup>15</sup>É anche necessario considerare il carico sulla RAM nel caso di file di grandi dimensioni.

<sup>16</sup>Non sono stati registrati segnali audio nella sezione 4 in quanto il controllo live dell'esecutore non garantisce di avere un segnale utile.

<sup>17</sup>Ciò nel caso in cui i file audio per la convoluzione siano registrati automaticamente con comandi da Csound; nel caso in cui i comandi di start e stop della registrazione vengano effettuati dall'esecutore si possono avere, pur mantenendo una certa coerenza timbrica, significative variazioni sia delle dinamiche che degli spettri sonori.

Figura 16: Sezione 05: registrazione file audio.



Ciò si vede dal carico della CPU, dall'utilizzo della memoria di sistema e dal surriscaldamento del processore. Si presume comunque la non necessità pratica di avere nello stesso brano tutte le configurazioni qui presentate, potendo ottenere risultati del tutto equivalenti con metodi diversi.

**Non si è presentato comunque alcun conflitto tra le periferiche audio e midi né tra i software.**

Si è riscontrata una possibilità di errori sull'audio (click, perdite di segnale) in caso di batteria poco carica o PC surriscaldato con ventola in funzione.<sup>18 19</sup>

Resta comunque il problema della portabilità del singolo progetto in tempo reale su diverse piattaforme dato l'alto numero di parametri da configurare.

Considerando il carico sul processore dei 2 software si è riscontrato, a parità di processi (o comunque processi molto simili), una pesantezza maggiore di MaxMSP; ciò è evidente dalla lentezza della grafica in parti cui sono attive subpatch complesse, soprattutto in quelle che generano audio in tempo reale.

La grafica di MaxMSP ha evidentemente un costo computazionale molto alto, mentre in Csound le features grafiche sono aggiuntive/opzionali, si può decidere comunque di non usare i Widgets o disattivare la libreria FLTK se non si ha la necessità stringente di monitorare a video segnali o settaggi.

La complessità del progetto lo rende rischioso, pur considerando la bassissima casistica di errori e di crash del sistema, per una eventuale performance "live", mentre non sussiste questo tipo di problema per il lavoro in studio.

A motivo di ciò sono state approntate 2 realizzazioni del brano registrando in MaxMSP i 4 canali tramite l'oggetto *sfrecord*.

### Suoni di sintesi.

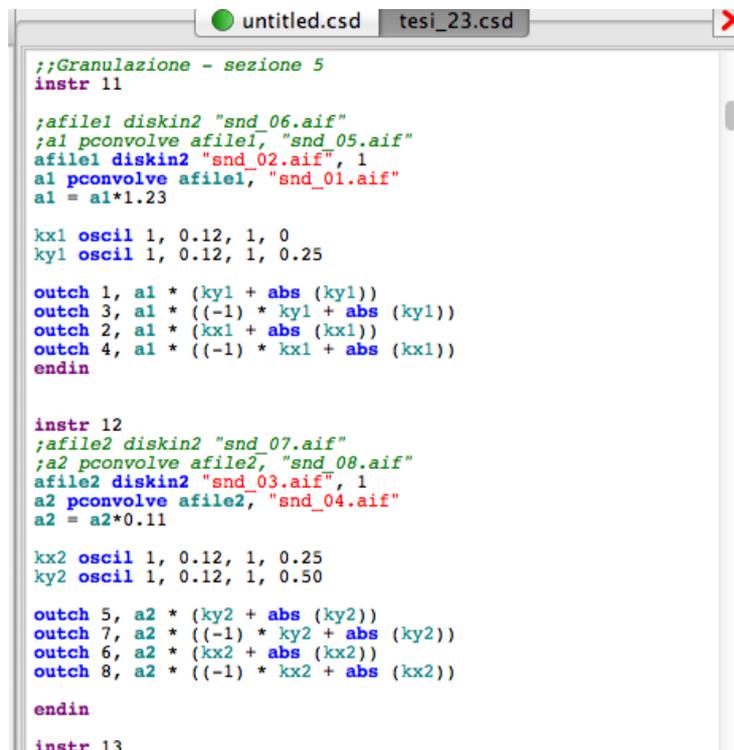
Pur non avendo esplorato per ovvie ragioni la totalità delle tecniche di sintesi, ho trovato che l'utilizzo congiunto dei due software permetta un flusso di lavoro agevole purchè ci si ricordi di commentare il codice di Csound e le patch di MaxMSP.

Per preferenze personali trovo più pratico creare il suono con Csound, la possibilità del *Render to file* permette di avere in pochi secondi un file audio mentre con MaxMSP è

<sup>18</sup>Ciò accade anche aumentando in maniera consistente i buffer audio di entrambi i software.

<sup>19</sup>Errori sull'audio si sono avuti anche usando la scheda audio integrata in una riduzione a stereo dei quattro canali audio.

Figura 17: Sezione 05: utilizzo file da Csound.



```
untitled.csd  tesi_23.csd
;;Granulazione - sezione 5
instr 11

;file1 diskin2 "snd_06.aif"
;a1 pconvolve afile1, "snd_05.aif"
afile1 diskin2 "snd_02.aif", 1
a1 pconvolve afile1, "snd_01.aif"
a1 = a1*1.23

kx1 oscil 1, 0.12, 1, 0
ky1 oscil 1, 0.12, 1, 0.25

outch 1, a1 * (ky1 + abs (ky1))
outch 3, a1 * ((-1) * ky1 + abs (ky1))
outch 2, a1 * (kx1 + abs (kx1))
outch 4, a1 * ((-1) * kx1 + abs (kx1))
endin

instr 12
;file2 diskin2 "snd_07.aif"
;a2 pconvolve afile2, "snd_08.aif"
afile2 diskin2 "snd_03.aif", 1
a2 pconvolve afile2, "snd_04.aif"
a2 = a2*0.11

kx2 oscil 1, 0.12, 1, 0.25
ky2 oscil 1, 0.12, 1, 0.50

outch 5, a2 * (ky2 + abs (ky2))
outch 7, a2 * ((-1) * ky2 + abs (ky2))
outch 6, a2 * (kx2 + abs (kx2))
outch 8, a2 * ((-1) * kx2 + abs (kx2))

endin
instr 13
```

necessario registrare i file “real time” su hard disk. La mia preferenza va a Csound anche in tutti i casi in cui si debba definire il comportamento di ogni singolo suono/nota in maniera precisa.

In MaxMSP si ha l’indubbia facilitazione di vedere graficamente le funzioni, siano essi inviluppi, altezze, etc. . . e di visualizzare il flusso audio tramite cavi virtuali: ciò è molto utile in catene di sintesi particolarmente complesse. Anche alcuni oggetti come *slider* e *number* possono essere utili per monitorare valori e livelli durante la composizione o l’esecuzione del brano.

Ho spesso adottato la tecnica di sperimentare una sintesi in MaxMSP per poi riportarla in Csound avendo l’ausilio dello schema grafico del percorso dei segnali audio e dei segnali di controllo. La possibilità delle uscite multiple in Csound mi è stata utile, in catene di sintesi complesse in serie, per prelevare il segnale audio a ogni stadio successivo o, in processamenti in parallelo, per determinare in via empirica la quantità di effetti da aggiungere ai suoni, utilizzando in pratica le uscite multiple come canali *wet* e *dry* di un multieffetto.

*Al termine di questo percorso di studi desidero ringraziare quanti mi hanno aiutato a portarlo a termine.*

*Stefano.*

## Elenco delle figure

1	Impostazioni audio in MaxMSP. . . . .	3
2	Impostazioni midi e audio in Csound. . . . .	4
3	Dispositivi audio. . . . .	5
4	Foglio excel per tabelle e timeline. . . . .	6
5	Patch MaxMSP per function. . . . .	7
6	Studio per movimenti Csound. . . . .	7
7	Time line MaxMSP. . . . .	8
8	Controllo MaxMSP tramite tastiera. . . . .	8
9	Patch generale MaxMSP. . . . .	9
10	Sezione 01: involuppi. . . . .	11
11	Sezione 02: moscil. . . . .	12
12	Sezione 02 textures FM. . . . .	13
13	Sezione 03: sintesi granulare. . . . .	13
14	Sezione 04: generazione note MIDI. . . . .	15
15	Sezione 04: lettura tabella altezze. . . . .	15
16	Sezione 05: registrazione file audio. . . . .	17
17	Sezione 05: utilizzo file da Csound. . . . .	18

## Indice

1	Software e connessioni. . . . .	3
2	Lavori preliminari alla composizione del brano. . . . .	5
3	Il brano. . . . .	6
3.1	Sezione 1: algoritmo di Karplus-Strong e sintesi sottrattiva. . . . .	10
3.2	Sezione 2: sintesi FM-modulazione di fase & segnali midi da Csound a MaxMSP. . . . .	11
3.3	Sezione 3: sintesi granulare e controllo tramite controller midi esterno. . . . .	12
3.4	Sezione 4: Strumenti di Csound controllati da MaxMSP. . . . .	14
3.5	Sezione 5: Utilizzo da parte di Csound di file <i>.aiff</i> generati/registrati da MaxMSP nelle sezioni precedenti - sintesi per convoluzione. . . . .	16
4	Conclusioni. . . . .	16