

# Superstereo Synth

## Relazione STALPM

GIANMARCO ROMITI

Conservatorio G. Nicolini Piacenza

gianmarco.romiti@conservatorio.piacenza.it

### Sommario

L'obiettivo del corso di "Sistemi, tecnologie, applicazioni e linguaggi di programmazione per la multimedialità" è stata la realizzazione di un synthesizer digitale che permette attraverso la programmazione di creare uno strumento di applicazione musicale. La realizzazione tecnica e grafica è facilitata da un prodotto finito di semplice ed immediato utilizzo, grazie a Faust (Functional Audio Stream), linguaggio di programmazione appositamente progettato per descrivere processi di sintesi e di elaborazione del suono, fruibile direttamente tramite applicazione web sull'interfaccia WebAssembly.

### INTRODUZIONE

Inizialmente si è resa necessaria la ricerca di un linguaggio informatico per cui fosse possibile uno studio ed un'asimilazione che si concretizzasse nella finalità di un progetto comprensibile a noi studenti, nonché duttile nella lavorazione ed adattabile al percorso di studi musicale che stiamo intraprendendo. In un primo momento, la semplicità nella concezione di un sintetizzatore digitale ci ha fatto comprendere come in realtà uno strumento di questo tipo si apre ad infinite possibilità di sviluppo, scelte personali volte a ricercare risultati adatti alle nostre esigenze. Tutte queste possibilità, accrescono consciamente le abilità nel gestire i linguaggi base, nonostante quindi gli stretti tempi di analisi per argomenti e impieghi complessi, come detto dalle infinite vie percorribili. Il progetto si è indirizzato verso la realizzazione di uno strumento complesso e personalizzabile, per la composizione digitale, un sintetizzatore digitale. A piccoli passi abbiamo sviluppato elaborando ed aggiungendo i vari moduli che vanno ad arricchire l'esperienza interattiva, siamo partiti da preset già concepiti per modellarne di nostri ed appunto integrarli nell'insieme. Senza conoscenze pregresse, abbiamo avuto come obiettivo lo sviluppo del nostro "Superstereo Synth".

### SVILUPPO DEL CODICE

La prima sezione Graphics User Interfaces (GUI) rende possibile la visualizzazione in moduli dei tools ordinati ed accessibili, nella parte centrale il controllo Master per l'intensità sonora e il ritorno visivo di verifica dell'uscita audio, nella parte sinistra abbiamo il macro modulo di sintesi sottrattiva "SUBTRACTIVE" che ci permette di applicare un filtro alla una sorgente sonora da un punto di vista spettrale "sottraendo" o modulando da essa bande di frequenze o singole parziali, infine nella parte destra abbiamo il macro modulo denominato "Phaser" che ci permette attraverso la duplicazione del segnale originale, ritardando di

qualche millisecondo la copia, di variare la fase in uscita sommando le fasi ottenute dalle due copie.



Figura 1: GUI. Interfaccia grafica del synth.

Di seguito la struttura primaria del programma, sia in forma di codice che rappresentata attraverso un diagramma a blocchi

```
process = subtractive : phaser : master : mutes :  
meters;
```

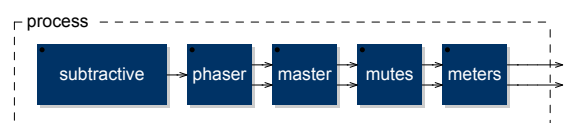


Figura 2: Process. Diagramma a blocchi della struttura del synth.

## GUI

Per facilitare la comprensione, abbiamo delineato in prima istanza l'interfaccia grafica, nominando i moduli e assegnandovi gli spazi noti. Lo studio sull'utilizzo dell'interfaccia utente ci ha permesso di ottimizzare gli spazi, rendendo simmetrico ed ordinato il tutto, il codice amplia le possibilità di personalizzare valori numerici, letterali e degli spazi a schermo.

```
// GUI
main(x) = hgroup("[01] Superstereo Synth",x);
s_g(x) = main(vgroup("[01] SUBTRACTIVE",x));
m_g(x) = main(vgroup("[03] MASTER",x));
p_g(x) = main(vgroup("[05] PHASER",x));
lmeter(x) = main(attach(
  x,an.amp_follower(0.150,x) : ba.linear2db :
  vbargraph("[02] L [unit:dB]", -70,0)));
rmeter(x) = main(attach(
  x,an.amp_follower(0.150,x) : ba.linear2db :
  vbargraph("[04] R [unit:dB]", -70,0)));
meters = lmeter, rmeter;
```

## MASTER

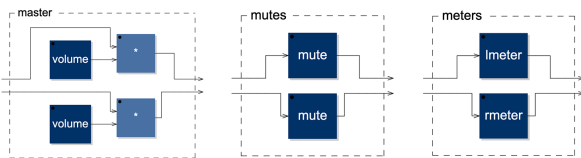


Figura 3: **MASTER**. Diagramma a blocchi del modulo Master.

Il modulo "Master" posto nella parte centrale, è adibito al controllo in entrata di tutto il flusso e le sue variazioni applicate in fase di costruzione del segnale. Come si può vedere dal diagramma a blocchi, l'imposizione del pulsante "Mute" che interrompe il segnale, chiudendo il canale di uscita era necessario, tanto quanto il fader verticale per il controllo del volume, pratico ed al centro del controller, ci permette di agire direttamente sui valori in scala logaritmica più consona a questo utilizzo. In ultimo ai lati troviamo i due Meters che ci permettono di verificare visivamente l'uscita audio dal canale di sinistra e di destra (Left Meters Right Meters).

```
// MASTER CONTROLS
mute = m_g(*(1-(checkbox("[04] Mute")))) : si.smoo;
volume = m_g(vslider("[02] VOLUME ", -6, -70, 12, 0.1)) :
  ba.db2linear : si.smoo;
bpc = p_g(checkbox("[01] Bypass"));
phaser = ba.bypass1to2(bpc, phchop);
mutes = mute, mute;
master = *(volume), *(volume);
```

## SUBTRACTIVE

Come sopracitato nell'introduzione sullo sviluppo del codice, il modulo "Subtractive" riguarda la sintesi sottrattiva del segnale sonoro, abbiamo arricchito questo modulo

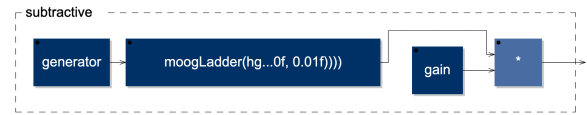


Figura 4: **SUBTRACTIVE**. Diagramma a blocchi primario del modulo Subtractive.

con vari applicativi, un pulsante di switch tra white noise e un'onda non sinusoidale a dente di sega, questo ci permette di creare due segnali completamente diversi e variare tra un segnale che può essere costante o che abbia cambiamenti periodici nella sua forma d'onda. Dopo la generazione primaria il segnale entra nel filtro dinamico "moog ladder" dove è possibile variare il controllo f-cut per filtrare le frequenze e il Q factor, il tutto è aiutato da un controllo di ampiezza per il gain sonoro.

```
// SUBTRACTIVE
// Sawfreq Custom
sawfreq = 123;
generator = (no.noise * switch),
  (os.sawtooth(sawfreq)*(1-switch)) :> _;
switch = s_g(checkbox("[01] Saw/Noise")) : si.smoo;
gain = s_g(
  vslider("[04] Gain [style:knob]", -12, -96, +12, 0.01)) :
  ba.db2linear : si.smoo;
Q = s_g(
  vslider("[03] Q [style:knob]", 5, 0.7072, 25, 0.01));
fcut = s_g(
  vslider("[02] Cut [style:knob]", 0.65, 0, 1, 0.001)) :
  si.smoo;
subtractive = generator : ve.moogLadder(fcut, Q) :
  *(gain);
```

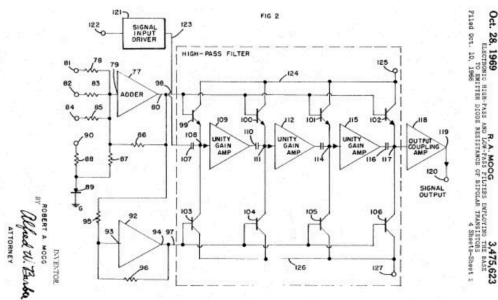
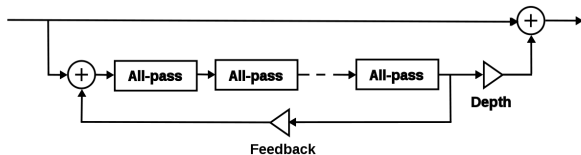


Figura 5: **MOOG LADDER FILTER**. L'originale brevettato risalente al 1966 depositato da Robert Moog.

## PHASER

In ultimo il "PHASER" che si posiziona nella parte sinistra dell'interfaccia, come un phaser tradizionale è composto da una serie di variabili N-allpass che alterano le fasi delle diverse componenti di frequenza del segnale. Il modulo accoglie come ci mostra il diagramma a blocchi, il "Superstereophaser" per la gestione dei 3 controlli principali: Low frequency oscillator "LFO" che genera forme d'on-

da a frequenza infrasonora, modulando gli effetti applicati e creando ad esempio un tremolo. ✕ Feedback, in uscita reintroducendo il segnale all'ingresso possiamo ottenere un effetto più intenso, creando risonanza enfatizzando le frequenze. ✕ Delay per creare ulteriori variazioni da applicare al nostro ciclo sonoro. ✕✕



**Figura 6: FEEDBACK.** Come agisce il feedback all'interno del ciclo phaser.

Uno shortcut utile posizionato sopra ai tre comandi principali è il bypass, che ci permette appunto di escludere i nostri controlli del phaser. Il segnale, modulato in uscita verrà accolto dall'hard limiter restringendolo secondo quanto voluto, come i filtri all pass che renderanno il segnale più netto o morbido.

```
// PHASER
lff = p_g(
  vslider("[02] Lfo [style:knob]",0.358, 0, 16, 0.001)
) : si.smoo;
fbk = p_g(
  vslider("[03] Feedback [style:knob]",
    -0.689, -0.999, 0.999, 0.001) : si.smoo);
del = p_g(
  nentry("[04] Delay [style:knob]",1, 1, 100, 1));

lfo = os.osc(lff);

phaserLR(N,x,d,g,fb) = x <: l,r
with{
  allpassL(d,g) = (+ <: de.fdelay(
    (ma.SR/2),d),*(-g)) ~ *(g) : mem,_ : +;
  allpassR(d,g) = (+ <: de.fdelay(
    (ma.SR/2),d),*(g)) ~ *(-g) : mem,_ : +;
  apseqL(N,d,g) = seq(i,N,allpassL(d,g));
  apseqR(N,d,g) = seq(i,N,allpassR(d,g));

  l= _, (+:apseqL(N,d,g))~*(fb):> _;
  r= _, (+:apseqR(N,d,g))~*(-fb):> _;
};

// STEREO SHUFFLER
pot = m_g(
  vslider("[02] WIDE [style:knob]",100,0,200,0.1)) :
/(100) : si.smoo;
somma = + : /(2);
diff = - : /(2);
sdm = somma,diff;
wide = _, * (sqrt(pot));
stereoshuffle= __ <: sdm : wide <: sdm;

superstereophaser = phaserLR(4,_,del,lfo,fbk) :
stereoshuffle;

// HARD LIMITER
chopper(a) = min(a) : max(-a);

hardlimiter = chopper(0.7) :
```

```
fi.lowpass(12,15000): chopper(0.9) :
fi.lowpass6e(20000);
phchop = superstereophaser :
hardlimiter, hardlimiter;
```

## CONCLUSIONE

Sarò sincero, in un primo momento, data la situazione poco piacevole degli ultimi mesi e le conseguenti difficoltà intercorse nel normale svolgimento delle lezioni, ero perplesso sul buon esito di questo come altri corsi. In particolare modo, questo corso pretende molto e tocca argomenti il cui approfondimento può essere infinito, d'altro canto, questa mole di possibilità ed informazioni di certo, per chi ha il piacere di interessarsene può regalare piacevoli soddisfazioni, sia per applicazioni basiche sia via via più ricche ed elaborate. ✕ Avendo rapporti pregressi con alcuni linguaggi informatici, posso affermare quanto il percorso scelto è stato sì impervio, ma guidato e correttamente calcolato per riuscire nel suo intento, creare da zero un applicativo utile e funzionale per le nostre competenze informatiche e musicali. ✕ Il corso ci ha permesso di apprendere costruendo piano piano, seguendo così un apprendimento passo passo che aiuta a destreggiarsi ed a porre le basi per ulteriori sviluppi futuri. Il Synthesizer ha fatto riaffiorare conoscenze e sistemi di funzionamento non solo digitale ma anche analogico altrimenti dimenticati, ci ha permesso di metterci alla prova e godere poi di un applicativo sempre utile e personalizzabile.

## BIBLIOGRAFIA E SITOGRAFIA

Blumlein, Alan Dower - British Patent Specification 394,325 (1958), JAES, vol. 6, no. 2, p.91. Curtis ROADS - The computer music tutorial (1996), The MIT press. Faust - Faust documentation, <https://faustdoc.grame.fr> Portale Produzione Musicale - <https://produzionemusica.com>