

.SSS: Synth/Sampler Stocastico controllabile via MIDI

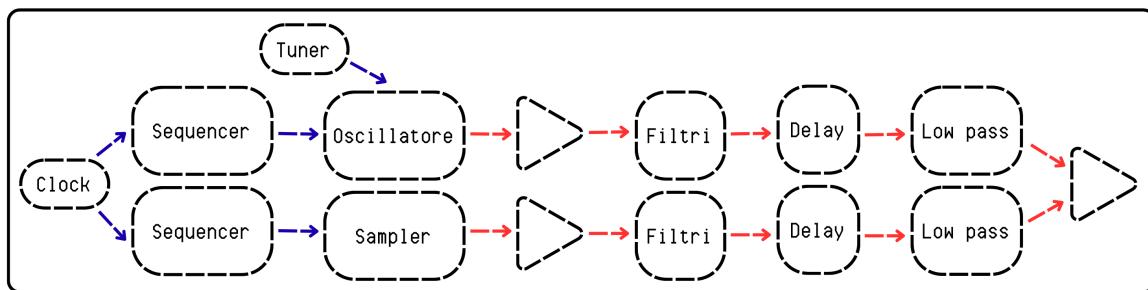
Andrea Gaio

Sommario

Questo progetto è una patch di Pure Data che comprende un Synth e un Sampler compatibile con AKAI MIDImix, un mixer MIDI riadattato allo scopo. Gli 8 fader presenti sul controller permettono di controllare un sequencer stocastico programmando la probabilità di ogni singola nota. La quantità di controlli sul MIDImix ha permesso ad ogni parametro di essere associato a due potenziometri diversi, uno principale ed uno di randomicità che muove il parametro principale con diversi algoritmi stocastici. Il progetto è un esperimento per analizzare come i comportamenti randomici applicati ad ogni parametro influenzino il timbro finale della macchina.

1 Introduzione

1.1 Schema a blocchi



Le frecce indicate con il blu sono controlli, quelle con il rosso sono segnali. È possibile interagire con tutti i parametri dei blocchi da GUI su Pure Data. Su controller MIDI invece, bisogna scegliere tramite un bottone fisico se si stanno variando i parametri del flusso superiore o di quello inferiore (sulla patch è un toggle indicato come "modalità edit"). Sul controller è anche presente un pulsante "Send All" nativo del MIDIMix che invia alla patch di pure data la posizione fisica attuale di tutti i controlli.

1.2 Algoritmi stocastici

Gli algoritmi implementati, grazie all'utilizzo della libreria Else di Pure Data, sono tre. Il primo è quello usato nel sequencer, ossia una scelta della nota da suonare in base ad una distribuzione di probabilità inserita dall'utente. Il secondo è quello di una scelta randomica di un valore all'interno di un intervallo. Questa scelta avviene ad ogni colpo di clock e viene usata ad esempio nell'oscillatore per la determinazione di attacco e rilascio della nota. Con il parametro principale viene controllato il centro dell'intervallo, con il parametro di randomicità la grandezza dell'intervallo. Il terzo algoritmo è l'unico che non lavora a colpi di clock ma è costantemente attivo per modulare in tempo reale il parametro (come se fosse un LFO). L'algoritmo scelto per questa operazione è quello del moto browniano. La quantità di randomicità indica quanto ampio è questo moto attorno al valore indicato. Questo algoritmo viene usato dai filtri, dal delay (è possibile visualizzare questi moti sui fader presenti nella patch), ma anche dal clock per la determinazione del BPM.

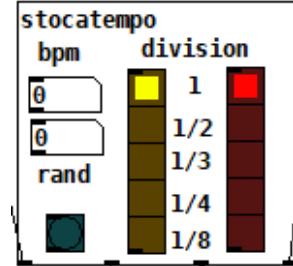


Figura 1: Il controller MIDIMix con la mappatura di .SSS

2 Componenti di controllo

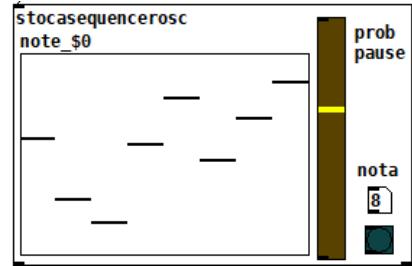
2.1 Clock

Il clock sceglie il bpm del progetto e la divisione ritmica dei due sequencer diversi. Il bpm ha un controllo di randomicità basato su moto browniano. Su controller è anche presente un pulsante di Tap tempo. La luce del Tap tempo sul controller si accende ad ogni colpo di clock base, senza quindi seguire le divisioni inviate ai due sequencer.



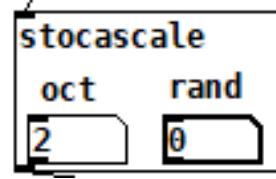
2.2 Sequencer

I sequencer scelgono che nota inviare ai due rispettivi generatori di segnali tramite una distribuzione di probabilità inserita dall'utente negli array dedicati. Gli indici degli array vanno da 0 a 7 e la distribuzione segue gli otto fader presenti sul controller MIDI. La nota (tra 0 e 7) scelta dal sequencer ad ogni colpo di clock è visibile sulla GUI. C'è anche un fader (fisicamente è un knob) per scegliere la probabilità che il sequencer suoni una pausa, indicata sulla GUI con la nota numero 8. Questa patch si occupa anche di accendere il led corrispondente della nota suonata sul controller, gialla per il synth e rossa per il sampler.



2.3 Tuner

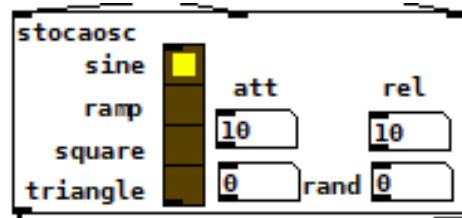
Nell'attuale implementazione il Tuner sceglie solo l'ottava di utilizzo del synth. Ha un parametro di ottava (di default 4) e uno di randomicità con l'algoritmo di scelta in un intervallo. Deve ancora essere aggiunto il controllo per la tonalità e per il modo del synth, in modo da poter seguire dal vivo una struttura armonica.



3 Generatori e modificatori di segnali

3.1 Oscillatore

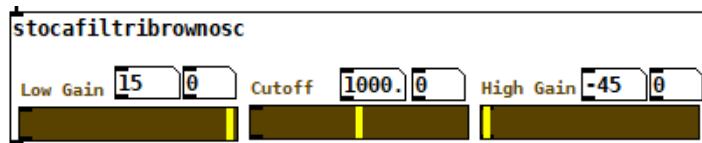
L'oscillatore del synth ha 4 possibili forme d'onda: sinusoide, rampa, quadra e triangolare. L'inviluppo della nota è dato da un Attack/Release con i rispettivi controlli di randomicità, che determinano l'ampiezza degli intervalli nei quali avvengono le scelte del parametro.



3.2 Sampler

Il lettore di campioni legge uno degli otto file contenuti nella cartella samples del progetto. Il tempo della lettura, che influenza anche il pitch del sample, è calcolato sul tempo che intercorre tra un battito di clock ed un altro. Nella patch originale sono caricati dei sample di una batteria stile 909.

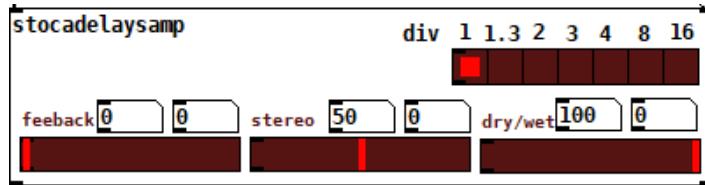
3.3 Filtri



I filtri implementati nella patch sono filtri biquadratici semiparametrici basati sulle equazioni dell'EQ Cookbook di Robert Bristow-Johnson ¹. I filtri sono un Low-Shelving e un High-Shelving aventi la stessa frequenza di taglio e la stessa ripidità. I parametri di Boost/Cut delle due bande vanno da -45dB a +15dB. Questi due parametri, uniti al parametro della frequenza di taglio, possono essere messi in movimento dalla quantità di moto browniano che si vuole dare. I parametri di randomicità del Low/High Gain sono stati rimossi dal controller fisico per dare spazio ad altre funzionalità. I fader visibili sulla GUI sono solamente indicazione visiva sul movimento in corso del parametro, per controllare i parametri vanno usati i number box in alto (così come per i parametri del delay).

¹<https://shepazu.github.io/Audio-EQ-Cookbook/audio-eq-cookbook.html>

3.4 Delay



Il delay è un ping-pong delay, con parametri per determinarne la divisione ritmica in maniera deterministica, e il feedback, l'apertura stereofonica e il mix in maniera stocastica tramite il moto browniano. Essendo il delay basato semplicemente su due linee di ritardo (una per il canale sinistro e l'altra per il destro), non essendo stati implementati algoritmi di warp, ed essendo chiaramente la divisione ritmica legata al bpm attuale, qualsiasi cambio di divisione del delay o cambio di bpm implica un detune momentaneo dato dal cambio di velocità di lettura della linea di ritardo.