Cruz, Tiago Luís Bastos da

Geração e processamento de áudio em tempo real : investigação e aplicação prática

Metadados

Data de Publicação 2020

Resumo Este projecto debruça--#se sobre sistemas de composição, geração e processamento de som em tempo real e interactividade entre humano e máquina e tem como objectivo o desenvolvimento de um software de geração e processamento áudio aplicado neste caso à guitarra. A primeira parte do trabalho(teórica) faz uma análise do panorama da interactividade e composição algorítmica de forma a contextualizar o projecto e a torna--#lo mais coeren...

Editor IPCB. ESART

Tipo report

Revisão de Pares Não

Coleções ESART - Música - Variante de Música Eletrónica e Produção Musical

Licenciatura em Música – Variante de Música Electrónica e Produção Musical

Projecto Individual II

Geração e Processamento de Áudio em Tempo Real - Investigação e Aplicação Prática

Tiago Luís Bastos da Cruz

Orientadores

Prof. Rui Dias

Prof. Gilberto Bernardes

Julho de 2015

Índice Geral

Agradecimentos

Em primeiro lugar quero agradecer à minha família, pela paciência, apoio e por acreditarem naquilo que faço. Ao meu irmão Filipe, à minha Mãe e ao meu Pai.

Um grande "obrigado":

Ao professor António Sousa Dias pela sua luz e clareza nos ensinamentos transmitidos.

Ao professor Gilberto Bernardes pela sua visão metódica e a sua ajuda proactiva.

Ao meu professor e coordenador Rui Dias por todo o conhecimento transmitido ao longo dos anos e pela sua motivação no curso.

Ao Luís Marques pelos "*brainstormings*" e dicas.

Ao Bruno por ser meu companheiro de casa, amigo e pelas noites que estudamos e trabalhamos.

À Haluz, pela inspiração, por acreditar em mim e estar sempre do meu lado, em qualquer parte do mundo.

A todos os outros que de qualquer forma contribuíram para que isto fosse possível:

Hélder silveira | Ricardo Nunes | Filipe Pereira | Àlvaro | Luca | João Neves | João Borges | Eduardo Vaz | João Nipo | Nando | Jéssica Nunes |

Resumo

Este projecto debruça-se sobre sistemas de composição, geração e processamento de som em tempo real e interactividade entre humano e máquina e tem como objectivo o desenvolvimento de um software de geração e processamento áudio aplicado neste caso à guitarra.

A primeira parte do trabalho(teórica) faz uma análise do panorama da interactividade e composição algorítmica de forma a contextualizar o projecto e a torna-lo mais coerente. Abordo também algumas questões relacionadas com composição algorítmica e processamento de áudio em tempo real que me parecem pertinentes para a realização do projecto.

A segunda parte do trabalho, descreve o trabalho prático realizado durante o primeiro e segundo semestre desde o conceito, desenvolvimento e problemas/soluções encontrados e reformulações.

A meta deste projecto será então construir um gerador/processador de áudio para guitarra eléctrica que interaja com o instrumentista consoante as suas acções.

Introdução

Nos últimos anos tem-se assistido a um crescimento explosivo no que toca à tecnologia. Esta tem vindo cada a ficar cada vez mais "inteligente" e intuitiva fazendo com que se abram novos caminhos na interacção homem-máquina e consequentemente, na música.

Na arte, nomeadamente na música, surgiram novos conceitos, conceitos esses que parecem ser cada vez mais inesgotáveis visto que a tecnologia está sempre a evoluir e como tal, a música, e mais concretamente a música electrónica, ganhou novos mundos chegando facilmente a um número maior de pessoas abrindo portas à criatividade.

No ultimo século assistimos a uma grande mudança na forma de expressão artística pois foi também um período de grandes descobertas. Este período, contemporâneo, contrasta inevitavelmente com o panorama musical dito erudito. Surgem agora novas técnicas de performance, de escrita musical, novos sons, e a tecnologia passa então a fazer parte do mundo musical, que já era vasto até então.

No que toca à música feita/assistida por computador surgiu um mundo "infinito" onde o utilizador pode programar a máquina com instruções precisas para satisfazer as suas necessidades composicionais, artísticas ou performativas. A máquina pode também assimilar ordens, "aprender" e gerar um output que vai condicionar o utilizador. Relativamente a este ultimo factor, começa-se a entrar no domínio da interactividade.

Tudo isto levou a que vários campos da arte se fundissem, se reinventassem.

Vivemos numa era em que a interactividade se assumiu como uma forma de relacionar o homem e a máquina, onde, na maior parte das vezes, este ultimo é influenciado pelo primeiro e vice-versa.

Com este projecto proponho me a criar exactamente isso. Um sistema para performance em que homem e máquina se consigam fundir criando uma obra só, obra essa que só faz sentido com a intervenção destes dois organismos.

A máquina servirá de output, output este que será programado de forma a responder a variações de amplitude, frequência, gestos entre outros. O utilizador será

o input respectivamente.

Capítulo I- Estado da Arte

O conceito de processamento em espectral é relativamente recente pois só há cerca de 3 décadas é que esta vertente conseguiu chegar aos utilizadores comuns de computadores. Como tal, é ainda um campo em crescimento que vale a pena explorar.

Há alguns programas de processamento espectral a partir de interface visual que merecem ser referenciados.

1- Metasynth (Wenger e Spiegel 2005)

Neste software é possível aplicar efeitos gráficos às representações FFT do som antes da ressíntese. Várias máscaras pode ser introduzidas no sonograma de forma a modificar o espectro.

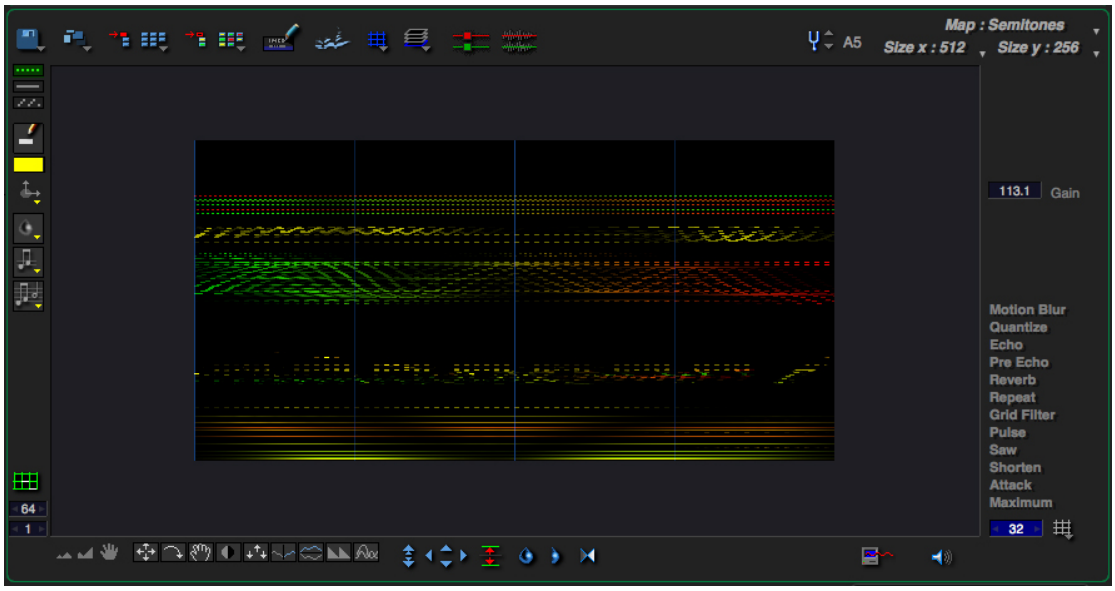


Figura 1- Ambiente do software Metasynth

Spear (Klingbeil 2005)

Este software oferece um grande controlo na análise edição e síntese dos parciais harmónicos do áudio.
É possível desenhar a quantidade de harmónicos que queremos gerar no som ou modular os harmónicos já existentes em tempo real fazendo com que o som não perca a sua essência original.

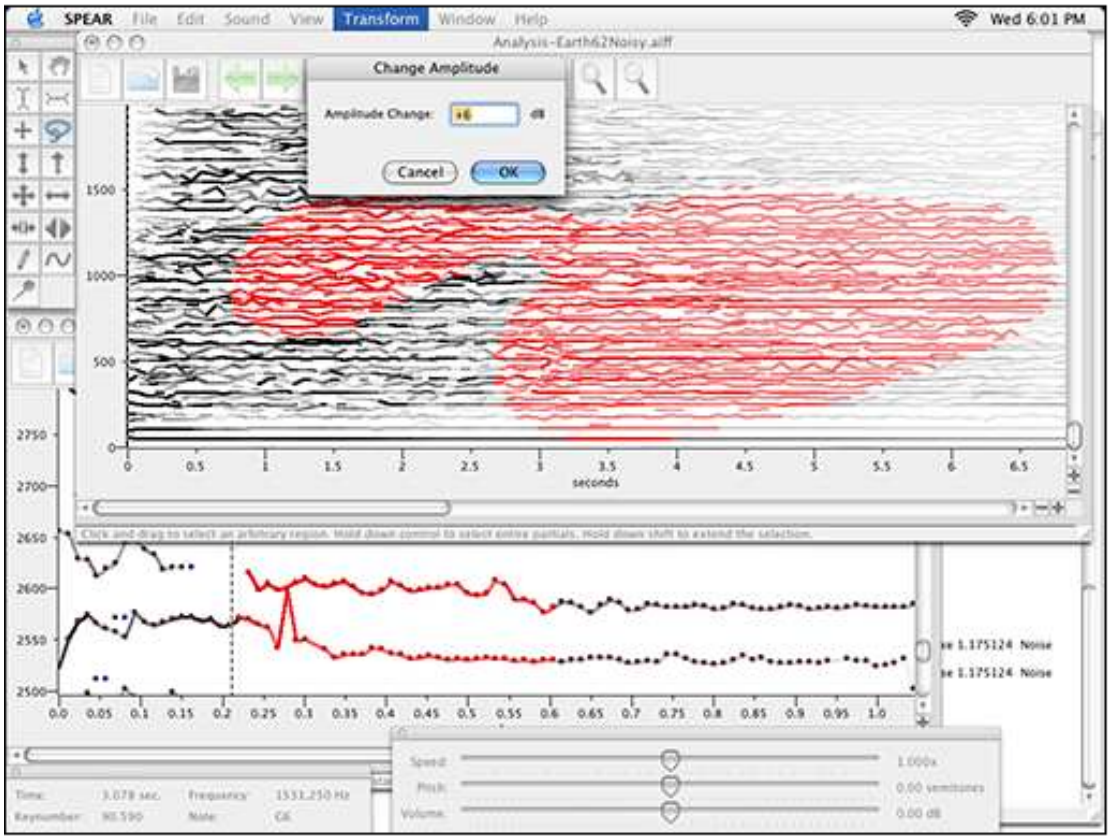


Figura 2 – Ambiente do software Klingbeil

Audiosculpt (IRCAM)

Audiosculpt é um programa desenvolvido pelo IRCAM e é conhecido pela alta precisão de análise de parâmetros FFT. Por exemplo, o utilizador pode ajustar a o tamanho da janela de análise para um valor diferente da janela de FFT.
Este programa possui algoritmos avançados para detecção da frequência fundamental, transposição, cálculo de envolventes espectrais, reconhecimento de parciais harmónicos, *time-stretching* com preservação de transientes, redução de ruído, convolução de áudio entre outros.

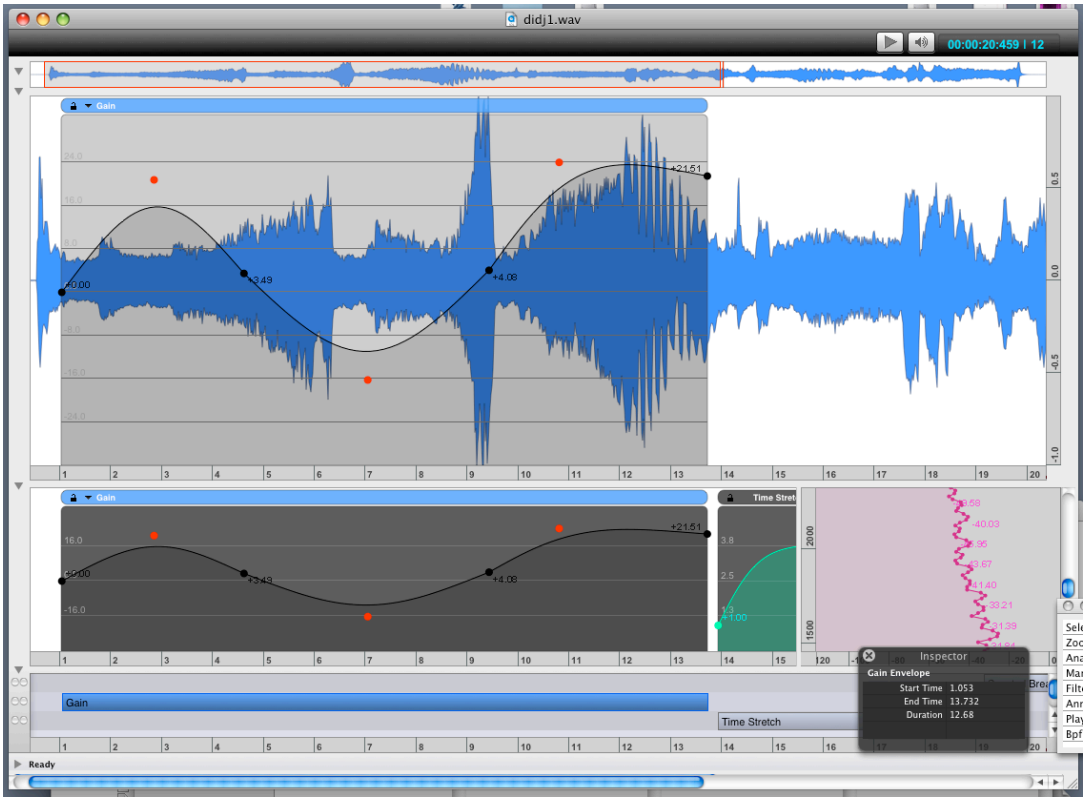


Figura 3 - Ambiente do software audiosculpt

Capítulo II- Interactividade

Estética dos sistemas musicais interactivos

Com o advento dos computadores tornou-se possível a descoberta de novos tipos de composição. Este aparecimento gerou também um novo desenvolvimento social e cultural influenciando por sua vez a prática de desenvolver e tocar música.

A música interactiva toma um especial lugar neste campo pois é agora possível explorar técnicas de composição e notação muito especificas ao mesmo tempo que se cria um enredo entre o utilizador e a máquina.

1- Interacção Homem/Computador

Por serem relativamente fáceis de usar, baratos e versáteis, os computadores tornaram-se numa ferramenta bastante atractiva para realizar diversas tarefas que anteriormente eram executadas por humanos. É agora possível, quebrar barreiras que até antes eram impossíveis, por exemplo criar uma peça para piano onde só um instrumentista não teria dedos suficientes para a tocar.

O pragmatismo do computador fez com que se tornasse uma ferramenta intuitiva e rápida. Não podemos comparar entre uma orquestra real e uma digital mas pelo menos, é possível compôr em computador e ter um feedback do resultado final relativamente rápido sem ter que contratar dezenas de músicos.

Há muitas outras razões para utilizar um computador na música, essas podem ter que ver com o desenvolvimento ou finalidade da obra. O computador revelou-se ser uma ajuda e por ter sido criado para outras finalidades é uma máquina bastante "aberta" à programação, podendo adaptar-se às necessidades de qualquer um.

No meu caso, o computador aliado à música pretende criar mais um caminho criativo, caminho esse que se pode juntar (ou não) com outras técnicas e práticas já existentes. Agrada-me o facto de praticamente depender de mim fazer com que algo aconteça, porque o computador dá me a possibilidade de programar, errar, corrigir os erros e ter um output mais rápido do que se estivesse a depender de várias pessoas ou mecanismos. O computador é cada vez mais o "All-in-one" da tecnologia.

Outro aspecto bastante importante dos sistemas musicais interactivos é o facto de, agora mais do que nunca, o "público" poder participar nas obras do artista e como tal, de certa forma, o público faz parte da obra. Este tipo de arte "nova" abriu portas à criação e juntou vários meios de estudo como por exemplo psicologia, arte, música, programação.

É um modelo artístico bastante completo e, na minha opinião, foi uma lufada de ar fresco no panorama cultural contemporâneo.

2- Composição algorítmica em tempo real

O dicionário de *Webster* define "algoritmo" como "um número predeterminado de passos ou instruções de maneira a resolver um problema específico dentro de um limite de acções".

Uma das minhas inspirações neste campo é o músico e compositor austríaco Karleinz Essl que nas últimas décadas desenvolveu um longo trabalho na área da composição assistida por computador. Um dos seus trabalhos mais interessantes a meu ver são as bibliotecas de composição em tempo real para Max msp (RTC-LIB) que contêm inúmeras abstracções para música generativa desde algoritmos genéticos a cadeias de Markov. Estes algoritmos são aplicados tanto a notas, como a dinâmicas ou fraseado, o que torna o output sonoro bastante credível, coerente e de fácil agrado ao ouvido.

A composição algorítmica está muitas vezes ligada à interactividade pois o sistema interactivo pode comportar-se sobre um âmbito de acções que este pode ter. Refiro a palavra "âmbito" pois o próprio utilizador tem de perceber que o computador não irá reagir a todas as acções que o humano faça. É possível fazer uma analogia com as regras de um jogo. O jogo só funcionará na perfeição se for jogado dentro de certos parâmetros.
Fala-se em composição algorítmica quando o computador pode fazer escolhas e até mesmo actuar sem intervenção humana.

Por se tratar de um programa interactivo é indispensável haver algoritmos que façam a análise e consequente disparo do som.

É importante que o computador tenha algum poder de decisão e para isso foram implementados métodos estocásticos como sistemas Lindenmayer, cadeias Markov ou até decisões aleatórias de forma a conseguir obter resultados sempre diferentes no output do software.

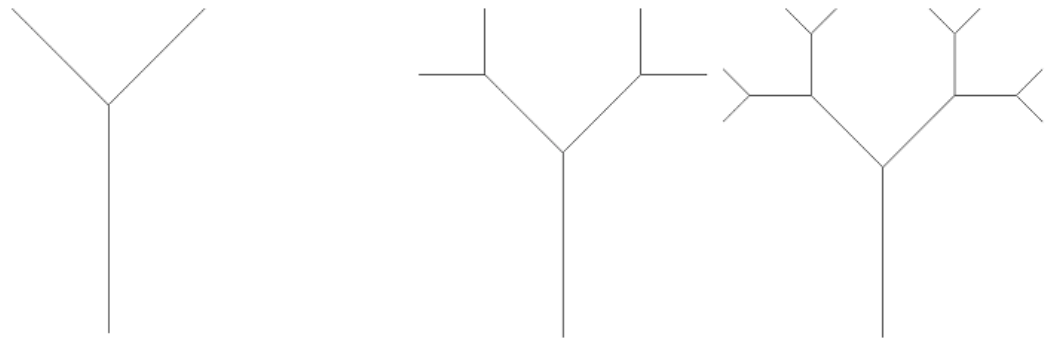


Figura 4 - Exemplo de sistema Lindenmayer

Cadeias de Markov

Para gerar algum tipo de comportamento é necessário detectar e organizar padrões. Estes padrões estão presentes na natureza bem como podem ser criados pelo homem.
A cadeia de markov analisa uma sequência de eventos, distribuida no tempo. Estes eventos são chamados estados. Quando o estado muda dá se o nome de transição.

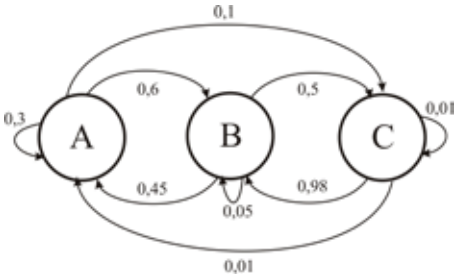
A estas transições está associada uma probabilidade que foi previamente analisada e é então, simulada gerando eventos parecidos.

A cadeia de markov de 1º grau tem em conta apenas o evento anterior, entrando para a estatística apenas o estado anterior.

Pode ser representada pela seguinte equação

$$\Pr(X_{n+1} = x \mid X_n = x_n, X_{n-1} = x_{n-1}, \ldots, X_1 = x_1) = \Pr(X_{n+1} = x \mid X_n = x_n)$$

, (1)



$$P = \begin{bmatrix} 0.3 & 0.6 & 0.1 \\ 0.45 & 0.05 & 0.5 \\ 0.01 & 0.98 & 0.01 \end{bmatrix}$$

Figura 5 – Cadeia de Markov de 3 estados (A, B, C) e respectiva matriz de transição.

Os estados de uma cadeia de Markov são sempre discretos e podem-se traduzir, na música, por exemplo em pausas, notas, intensidades , tudo o que tenha um valor associado.

Capítulo III- Processamento de áudio

Processamento de áudio em tempo real

O processamento de áudio em tempo real é a análise, processamento e consequente ressíntese de um conjunto de amostras espaçadas no tempo. É considerado em tempo real quando a resolução destes processos não demora mais do que dois segundos.

A manipulação do áudio digital pode revelar-se útil se pretendermos criar um novo sinal com características alteradas em relação às do sinal de entrada, por exemplo, uma nova altura, timbre, textura, espectro.

Pode-se então definir o ciclo dsp como uma sequência de 3 processos:

- São recebidas as amostras do sinal de entrada
-
- Dá se o processamento destas amostras e geração de um novo sinal
-
- Output do sinal gerado

Por vezes pode dar-se um atraso de processamento – latência – que pode dever-se ao facto de o computador não ter processamento suficiente para ser rápido a resolver os processos.

1- Processamento Espectral

Por ser um dos meus objectivos a modificação do áudio em tempo real decidi investigar algumas técnicas para esse efeito. Uma das técnicas mais usadas é a transformada de Fourrier.

1.1 – Transformada de Fourier

A análise de Fourier é uma técnica que permite transformar a representação de um sinal do domínio do tempo para o domínio da frequência, e vice-versa.

É utilizada em diversas aplicações no domínio das telecomunicações, como a análise de circuitos, a propagação de ondas rádio, ou o estudo de radiação das antenas.

A ideia base foi popularizada em 1965 mas várias investigações apontam para 1805, ano em que Carl Friedrich Gauss descobriu este tipo de análise.

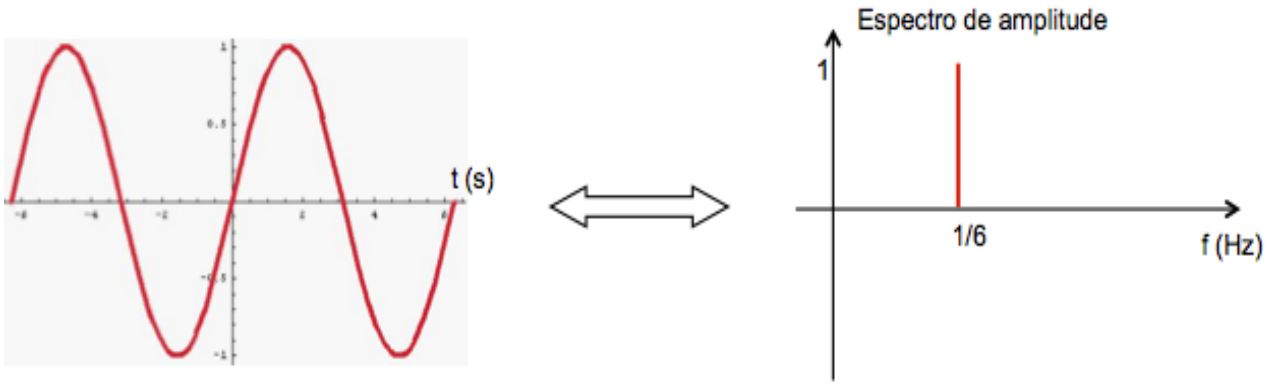


Figura 6- Conversão do domínio da frequência para amplitude

Na análise de Fourier, os sinais dividem-se em dois grupos:

-Sinais periódicos: A técnica utilizada para calcular o espectro do sinal é a Série de Fourier.

-Sinais não periódicos: a técnica utilizada para calcular o espectro do sinal é a Transformada de Fourier.

Para a maioria dos sinais existentes, o espectro é obtido consultando tabelas, que relacionam as características temporais e espectrais.

1.2 - FFT no Max

O programa Max Msp permite aplicar esta técnica e em paralelo com o ambiente Jitter é possível ter bastante controlo sobre os parâmetros do espectro pois podemos enviar o som para duas ou mais matrizes dentro do objecto *Pffft*. Isto pode ser útil se precisarmos de fazer convolução de sons ou outro tipo de processamento por exemplo desenhar uma máscara numa matriz para a quantidade de processamento desejado.

1.2.1 FFT e Matrizes em Max msp jitter

A transformada de Fourrier possui valores cartesianos, contudo é necessário trnsformá-los para valores polares para obter o mais perceptível detalhe antes da ressíntese. Desta forma obtemos a amplitude e a diferença de fase entre os bins.

Esta figura mostra como armazenar informação FFT numa matriz Jitter. Usaremos 32bitfloat. A altura da grelha representa o número de bins de frequência , que deve ser metade do tamanho fft, e a largura representa o número de janelas de análise. Usa-se normalmente uma matriz de dois planos, para a amplitude e fase respectivamente.

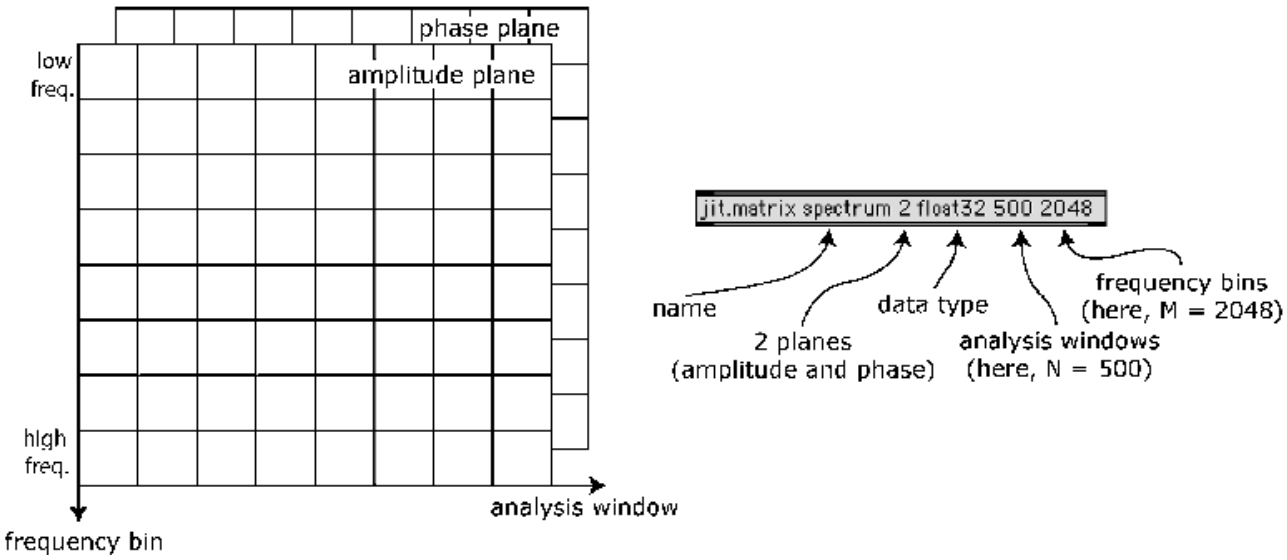


Figura 7 – Matriz de 2 planos em Jitter

Capítulo IV- Projecto individual – parte prática

Inspiração

Há alguns músicos e projectos que me inspiraram na concecção e desenvolvimento deste trabalho.

Corail

Corail é um ambiente musical interactivo para saxofone tenor ou soprano que explora algumas possibilidades no que toca ao processamento de audio em tempo real como por exemplo:

- pitch shifting
- identificação de gestos
- mapeamento de envolvente
- efeitos reverb
- espacialização

Todos estes efeitos são gerados pelo computador através do software Max Msp.

Esta peça não possui partitura. O instrumentista tem â sua disponibilidade uma série de eventos que podem ser "chamados" pelo saxofonista a qualquer momento, consoante as suas ordens performativas.

por exemplo:

- persistência em certas notas (faz com que o programa comece a gerar respostas por si).

Cada performance pode ser diferente mas a estrutura meta composicional mantêm-se.

Para o sucesso desta peça o instrumentista deve memorizar através da prática todas as funções internas do programa.

Robert Rowe

Robert Rowe é uma referência no panorama da música electroacústica e investigação musical tendo já trabalhado em parceria com o instituto de sonologia em Utrecht e Ircam em Paris.

Foi uma inspiração enorme não só pelo seu livro "machine musicianship" mas também pelas suas composições para instrumento e sistema interactivo como por exemplo a obra "Submarine" (1996), e "Flutter" (2002). Este tipo de peças, elevam, a meu ver, a composição a um outro nível pois o autor compõe tendo em conta diálogos e ambientes que possam surgir do próprio instrumento e da interacção/processamento.

Homem e máquina fundem-se numa obra só, sendo inseparáveis e indispensáveis para o sucesso da obra.

O Uivo – Rui penha

Rui penha é um compositor e performer português que tem desenvolvido várias obras e software para musica electroacústica e por computador.

O Uivo é um improvisador automático feito para o guitarrista Paulo Vaz de Carvalho. O software escuta o músico e improvisa de volta usando a sua própria linguagem musical e uma manipulação dos seus próprios sons .

Figura 8 - Software desenvolvido pelo compositor Rui Penha para a obra "o Uivo"

Utilidade

O desenvolvimento deste gerador e processador automático tem como objectivo a exploração de várias possibilidades de geração e processamento e áudio em tempo real e em simultâneo a interacção directa com a máquina que fará o papel de meta-instrumentista.
Este processador será útil num contexto de *live* performance onde o instrumentista deverá previamente ter em mente a programação aplicada e de que forma a máquina pode reagir, para com ela poder interagir correctamente.

Porquê um gerador/processador "automático"?

O processamento de áudio em tempo real já existe há várias décadas. O exemplo mais comum que podemos encontrar é por, exemplo, a cadeia de efeitos usada por um instrumentista (pedais e outros processadores) ou até mesmo a estação de tratamento de palco utilizada por um técnico de som (mesa de mistura e dsp).

Estes utensílios são bastante funcionais e conseguem responder às necessidades da performance com muita fidelidade. Contudo, o funcionamento destes processadores está quase sempre dependente de uma pessoa que terá que activar ou desligar os efeitos.

Com este projecto pretendo fazer um programa de processamento de áudio dinâmico, ou seja que se altere e se adapte ao input do músico mas que ainda assim consiga tomar decisões gerando notas.

Interessa-me particularmente o processo de programação algorítmica para chegar a este tipo de resultados que simulam percepções musicais humanas.

Hardware

Como interface de comunicação com o computador utilizei uma mesa de mistura Yamaha mx10 que serve como placa de som pois tem entrada usb.

Figura 8 – Interface utilizado (Yamaha mx10c)

Foram também feitas experiências com uma DI. Mais informações na secção "Problemas e Soluções".

Software

O software desenvolvido para projecto foi implementado em Max Msp e Jitter e em paralelo será também utilizado o sequenciador Logic pro X.

Max msp

Tem como função gerar notas e sinal áudio através dos diferentes módulos de processamento. Estes módulos podem ter diferentes combinações ou ser activados com diferentes gestos que o programa vai receber e interpretar através de processos estocásticos.

Logic Pro X

Este sequenciador foi usado de forma a receber a informação midi que é gerada no Max Msp.

É possível que o utilizador escolha qualquer plugin para reproduzir a informação midi, fazendo com que o computador tenha um timbre mais real.

Protótipo técnico

Quando a ideia deste software surgiu criei inicialmente um protótipo de forma a conseguir ter sinal de entrada e algum controlo do aúdio no max. Para isso implementei um compressor e um overdrive na entrada do sinal e um limiter na saída. Estes módulos servirão apenas para fazer alguns ajustes quando necessário.

Na imagem em baixo vemos o layout do programa. É possível calibrar o software a partir de alguns parâmetros como o threshold - para a detecção de ataque - , número de parciais a serem analisados – útil para uma melhor detecção de frequencia, espaço de tempo entre cada detecção de ataque – reattack, e margem para desvio de detecção de frequência – vibrato

É possível também saber quais as notas que o programa está a gerar bem como em que modo de interacção o programa se encontra. .

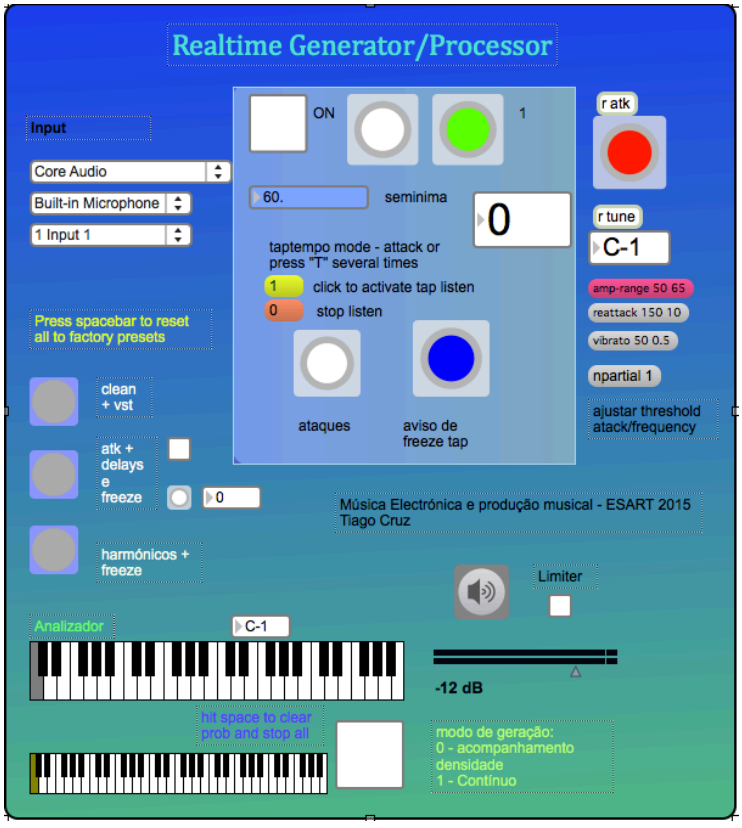


Figura 9 – Protótipo técnico

Modos de interacção

Os modos de interacção são os diferentes efeitos que o computador pode fazer.

O primeiro é o clean + vst que detecta notas e ataques produzindo também notas e ritmos parecidos. Foi utilizada uma cadeia Markov de 1º grau para o efeito. Quando o computador deixa de receber ataques durante um determinado período de tempo este avança para o segundo modo.

O segundo modo de interação é o "atk + delays e freeze" que produz atrasos no som – também consoante o ataque do input – e congela 8 frames de áudio fazendo com que as frequências fiquem a soar continuamente. Neste caso, quando se dá o 2º freeze o programa avança para o ultimo modo.

O terceiro modo faz a ressíntese de certos harmónicos que são disparados com uma envolvente pré-definida.

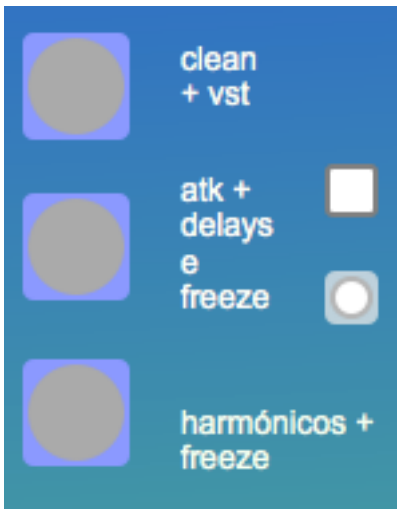


Figura 10 – Modos de interacção

Conecção Max msp – Logic Pro x

Para enviar informação midi para o Logic é possível selecionar a opcção “to max 1” no objecto “noteout” e criar no sequenciador um instrumento externo enviando-o para um bus que tenha o plugin.

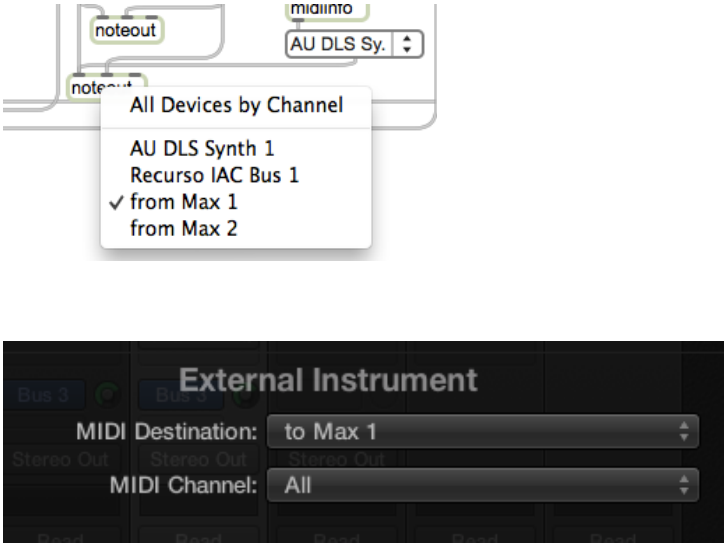


Figura 11 – Conecção Max – Logic pro x

Metrónomo

Foi necessário criar um metrónomo de maneira a sincronizar a informação midi gerada ou até outros efeitos.

Os valores tidos em conta no metrónomo são bpm's e também milissegundos. Estes dois valores vão ser posteriormente usados para disparo e quantização de durações.

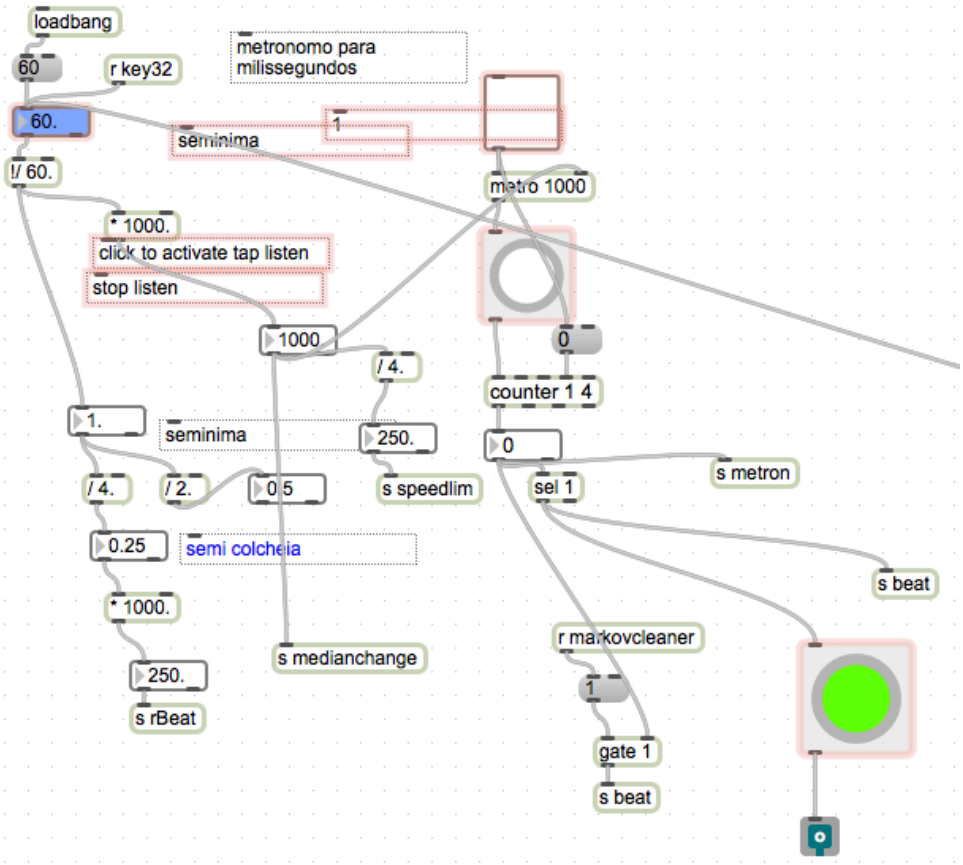


Figura 12 - Metrónomo

Este metrónomo conta também com um tap tempo que pode ser activado através de ataques que o músico faça ou pressionando a tecla “t” algumas vezes. Este módulo faz a média da duração entre toques consecutivos.

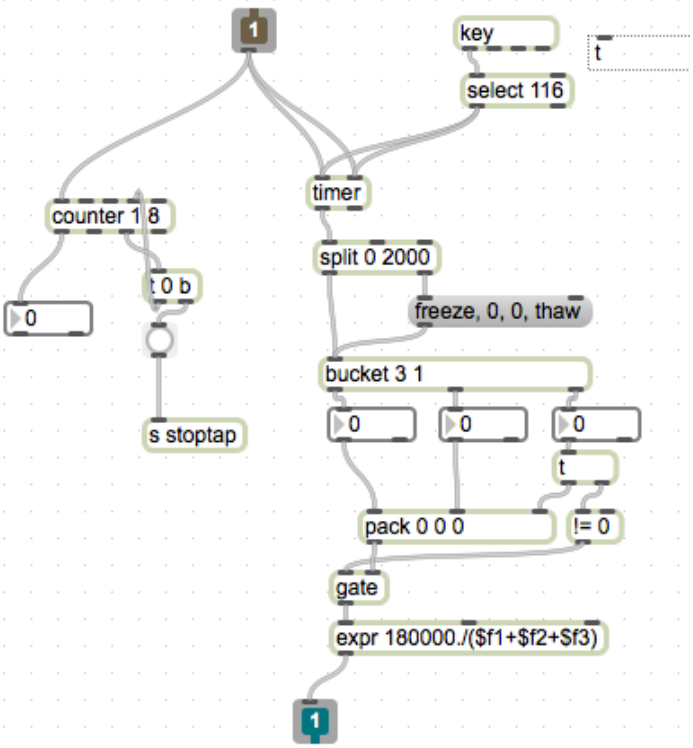


Figura 13 - Tap tempo

Definiçao de inputs de interacção (detectores)

Como detecção de inputs cheguei há conclusão que era importante ter três módulos de detecção (amplitude, frequência e repetição). No que toca à amplitude foi importante definir vários thresholds de modo a criar âmbitos de valores para que fosse disparada a acção seguinte desejada.

1-Amplitude

A amplitude é reconhecida através de um objecto que detecta a mudança, ou seja, quando houver uma mudança de amplitude (por exemplo um decrescer) o objecto [change] detecta um pico e retorna um [1]. De seguida o valor da amplitude é armazenado num float e disparado quando o objecto [change] detecta a mudança.

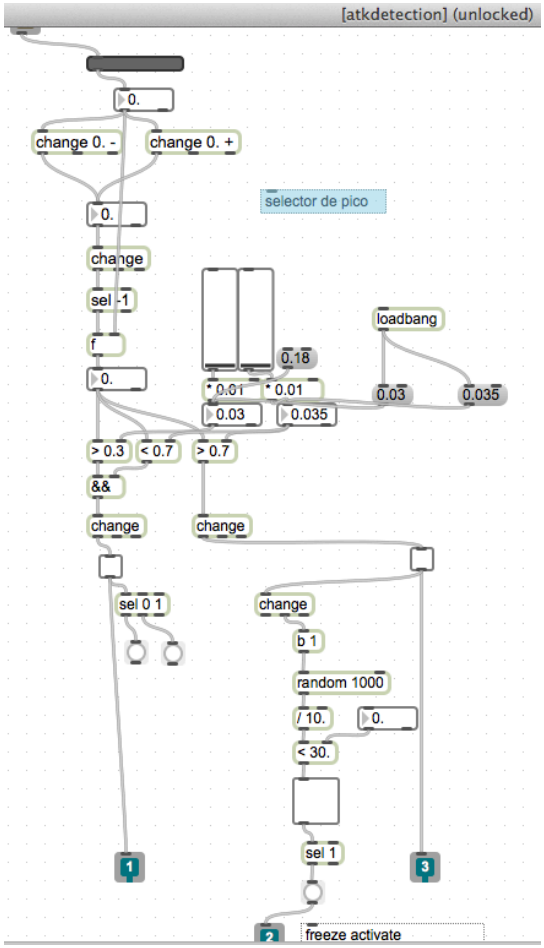
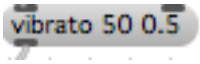


Figura 14 – Detector de ataque

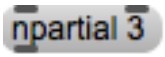
2- Frequência

No que toca à frequência, o sinal está a ser analisado através do objecto [pitch ~] . Este objecto tem algumas mensagens que podem ser cruciais na análise do áudio obtendo assim uma detecção mais fidedigna da altura da nota. De referir : "vibrato" e "npartial".

- Vibrato – impede desvios na detecção da frequência em "x" milissegundos e num âmbito de "y" meios tons/ cents.



- nPartial – estes são os parciais a ter em conta na detecção da frequência. No caso abaixo o parcial 3 ira ter a metade do peso da frequência fundamental, no que toca à detecção.



O pitch ~ devolve então os valores de frequência, midi, e componentes sinusoidais.

Optei que a frequência fosse detectada e traduzida imediatamente para midi porque não há tanta variação nos valores fazendo com que a detecção de altura seja mais estável.

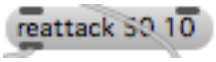
Para evitar também saltos na nota desnecessários todas as notas detectadas apenas são oficializadas e enviadas para fora depois de receber um bang (ataque). Esta operação é executada pelo objecto [i]. A mensagem [in1] que está ligada ao [i] foi crucial pois funciona como uma espécie de cleaner, evitando que este último objecto fique com valores guardados mesmo depois do reset to programa.

Foram também realizados testes com o objecto [speedlim] para reduzir o tempo de output mas este não se revelou eficaz.

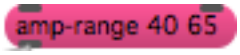
A detecção de ataque no objecto [picth ~] tem também algumas mensagens que auxiliam na análise da amplitude, de referir: “re attack” e “amp range”.

-Re attack – é o tempo, em milissegundos, em que é detectado um re attack se a amplitude subir mais de “x” db.

-



- Amp-range – threshold mínimo e máximo de amplitude em que é enviada uma nota.



Usei o patcher [p osc-bank] que serve para ressíntetizar alguns harmónicos. Este será tratado mais à frente.

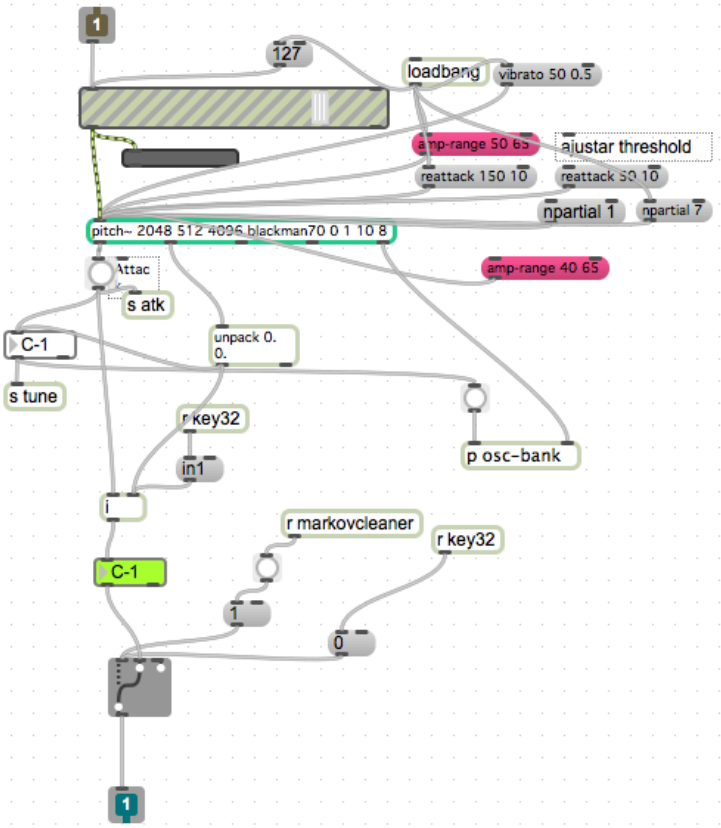


Figura 15 – Detector de frequência

3- Repetição

A detecção de repetição está combinada com a detecção de ataque e pode ser é activada se receber 2 ataques (bangs) num espaço de tempo que a ser definido.

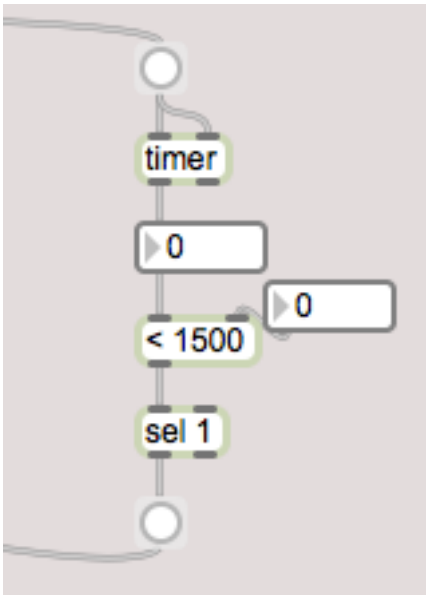


Figura 16 – Detector de repetição

A detecção de repetição está combinada com a detecção de ataque e pode ser é activada se receber 2 ataques (bangs) num espaço de tempo a ser definido.

4- Densidade

Este módulo permite ao computador saber a quantidade de notas recebidas num espaço de tempo fazendo-o gerar mais ou menos notas consoante o input do músico seja maior ou menor.

Neste modo são assumidos três estados:

- estado 0 – o computador não envia notas para a cadeia Markov fazendo com que esta seja inutilizada.
-
- estado 1 – o computador envia notas para a cadeia Markov a cada primeiro tempo.
-
- estado 2 – o computador envia notas para a cadeia Markov a cada primeiro e terceiro tempo.

Estes estados vão mudando consoante a utilização do utilizador.

-

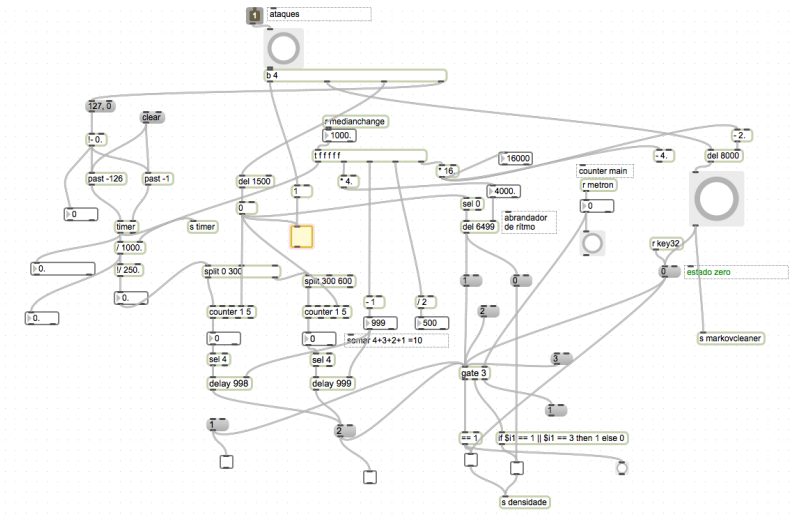


Figura 17 – Detector de densidade

5- Detector de início de frase

Este detector percebe a primeira nota recebida e envia-a para a geração markov fazendo com que haja sempre uma infinidade de opções. Se o módulo markov esgotar todas as possibilidades de transições, volta à primeira nota começando de novo com geração "aleatória".

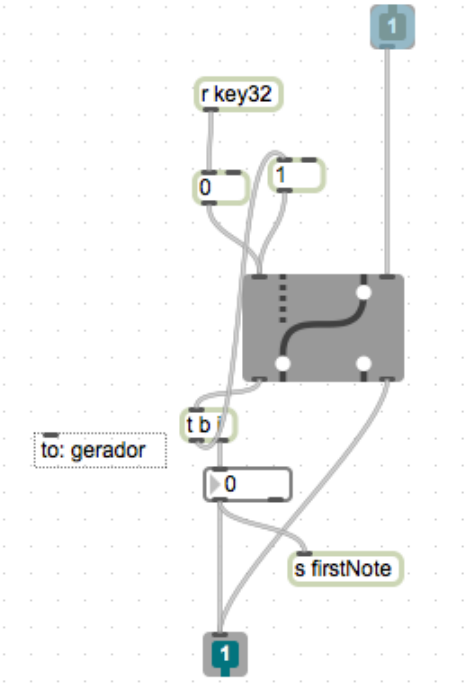


Figura 18 - Detector primeira nota

Estes cinco detectores servem então para activar os módulos de geração/processamento consoante parâmetros estabelecidos.

Módulo de geração Markov

É neste módulo que se dá a geração de notas e ritmos onde é utilizada uma cadeia markov de 1ª ordem.

Este módulo tem dois modos de acção: modo acompanhamento e modo contínuo. O modo acompanhamento é o mais reactivo pois é influenciado pela densidade das notas do utilizador fazendo com que não esteja sempre a ser gerado áudio.

Já o modo contínuo está sempre a gerar som independentemente do que o utilizador faça.

As notas são recebidas pelo objecto [anal] que guarda números de dois em dois. De seguida, o objecto [prob] cria uma tabela de transições que apenas é activada (no caso do "modo acompanhamento") quando recebe a média da densidade. No caso de serem esgotadas todas as possibilidades de transições, este objecto envia um bang com a primeira nota recebida.

Para os ritmos é também usada uma cadeia de markov, fazendo com que haja geração de ritmos parecidos (mais eficaz no modo contínuo).

Por ser em tempo real é necessária uma quantização dos ritmos recebidos de forma a que na geração haja um disparo das notas no tempo correcto. Para isso foi desenvolvido um quantizador que recebe o valor mínimo possível ,através do metrónomo, em milissegundos (por exemplo uma semi-colcheia). Este valor é dividido pelo valor que entra no timer (espaçamento entre duas notas consecutivas). O resultado desta operação é então arredondado para o valor inteiro mais proximo.

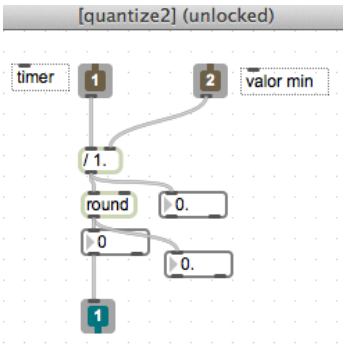


Figura 19 - quantizador de ritmo

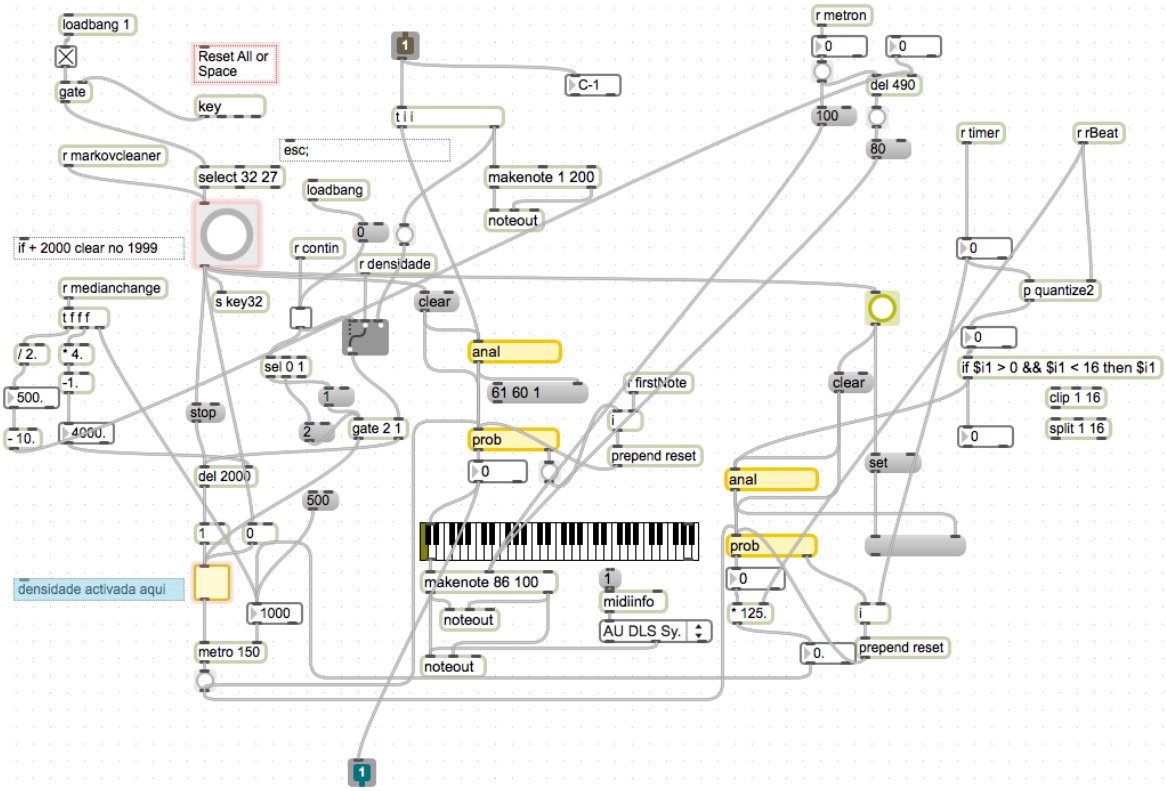


Figura 20 – Módulo Cadeias de Markov

Módulos de processamento

1-Sintetizador de harmónicos

O sintetizador de harmónicos consiste, tal como está, na ressíntese de 8 harmónicos do áudio. Estes harmónicos podem ser uma transposição real da frequência ou ser notas compactadas na oitava. No segundo caso, é como se houvesse uma aproximação do tom.

Estes harmónicos são então produzidos num [poly~] onde são enviados para o objecto [cverb~] de maneira a esbater um pouco o som.

É possível desenhar o envolvente destes harmónicos.

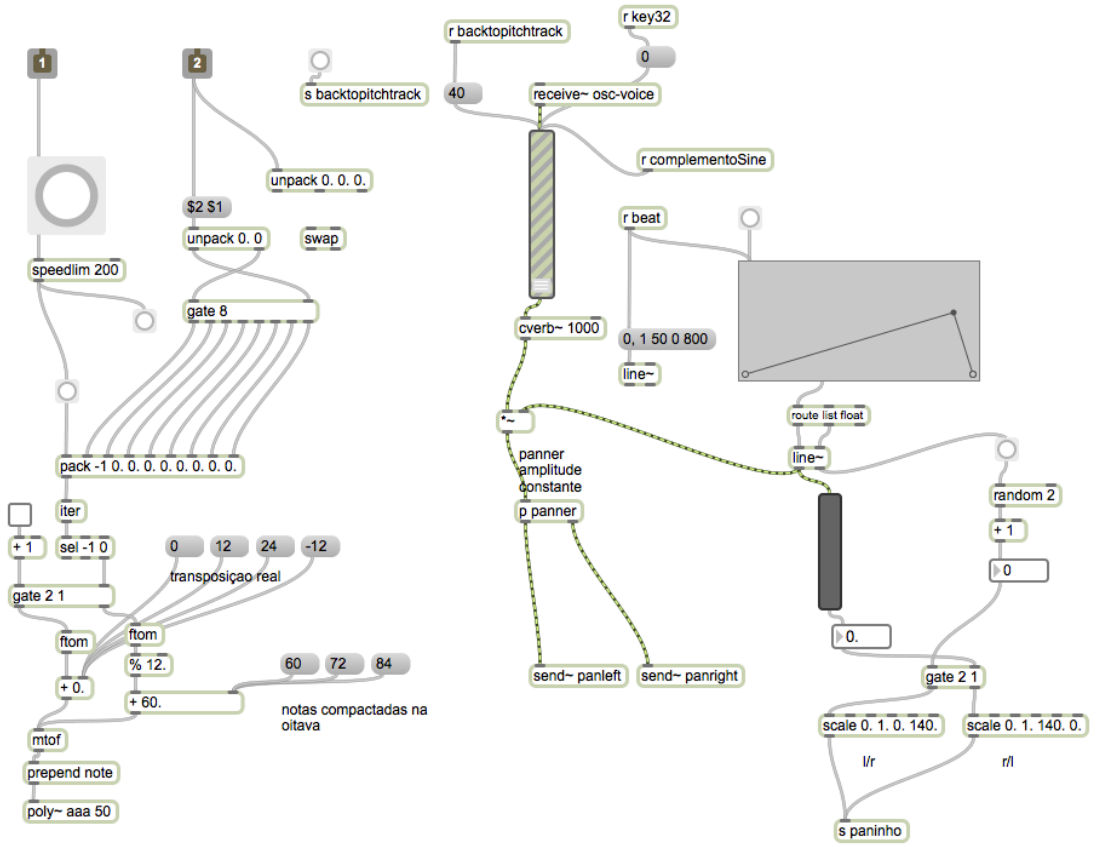


Figura 21 – sintetizador de harmónicos

2-Panner amplitude constante

A instensidade do som é proporcional à raiz quadrada da amplitude. Logo, se queremos ter uma mudança linear na intensidade ao ir de 0 a 1 e de 1 a 0, precisamos de calcular a raiz quadrada dessa mudança linear para calcular a amplitude.

Com este método, quando o som se encontra no centro, entre duas colunas, em vez de a amplitude de cada coluna ser 0.5, será a raíz quadrada de 0.5, que é 0.707 (que é um aumento de 3 db comparando com 0.5).

Isto reduz o "buraco no meio" existente no panner de amplitude linear.

O computador decide qual a direcção a tomar.

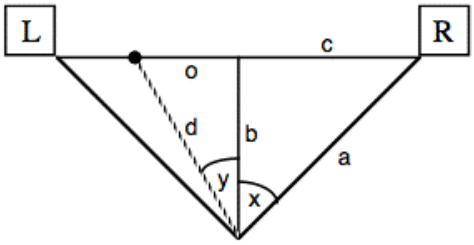


Figura 22 - trigonometria aplicada ao som

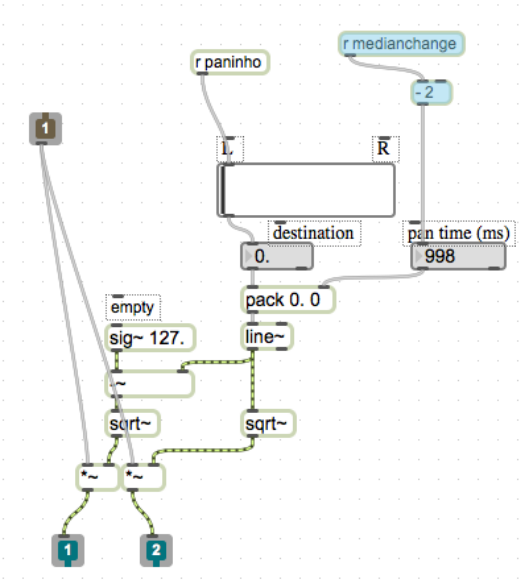


Figura 23 - Panner amplitude constante

3-Mapeador de envolvente

O módulo de corte de ataque consiste e na detecção de um impulso mediante um threshold . Quando este impulso é detectado é criada uma rampa de volume fazendo com que cada nota dada soe transformada. Quando este módulo encontra uma nota nova faz imediatamente reset ao volume crescendo então a partir daí.

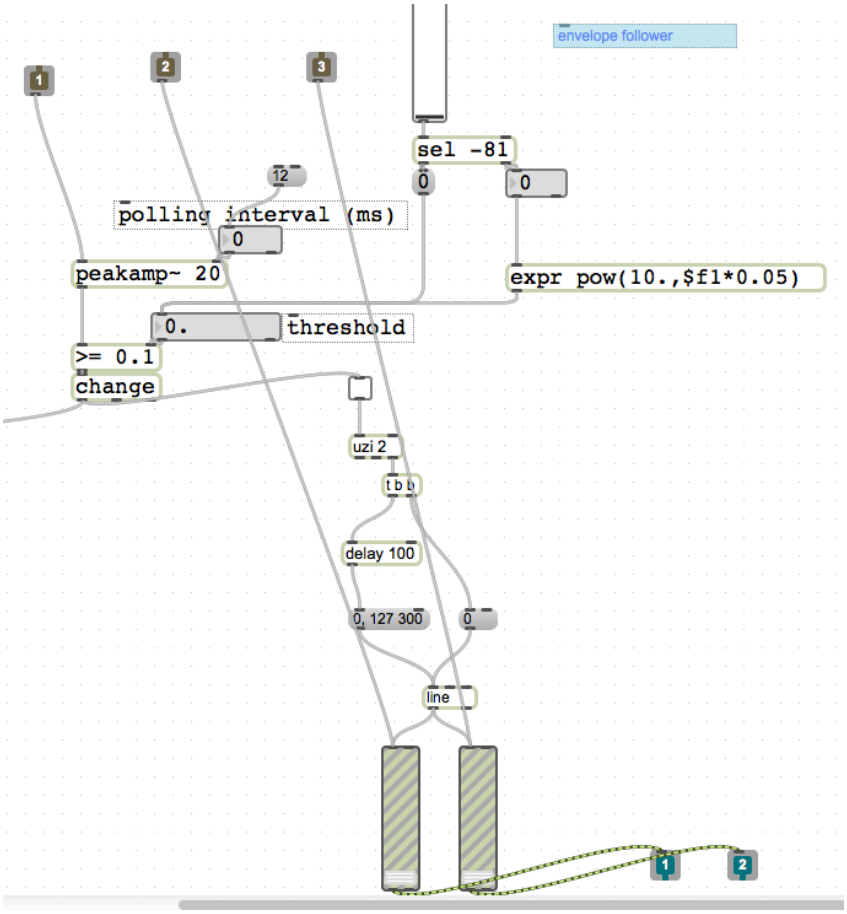


Figura 24 - Mapeador de envolvente (modulo “atkcutter”)

4- Módulo freeze

O módulo de freeze consiste no congelamento de um frame aúdio que é guardado numa matriz de oito frames e consequentemente reproduzido "em loop". Este som gravado pode ser reproduzido imediatamente ou em crescendo, dependendo dos valores do slider_up. É também possível definir o decay do som gravado através do slide_down. Combinando estes dois sliders é possível criar um efeito de crossfader entre os sons reproduzidos.

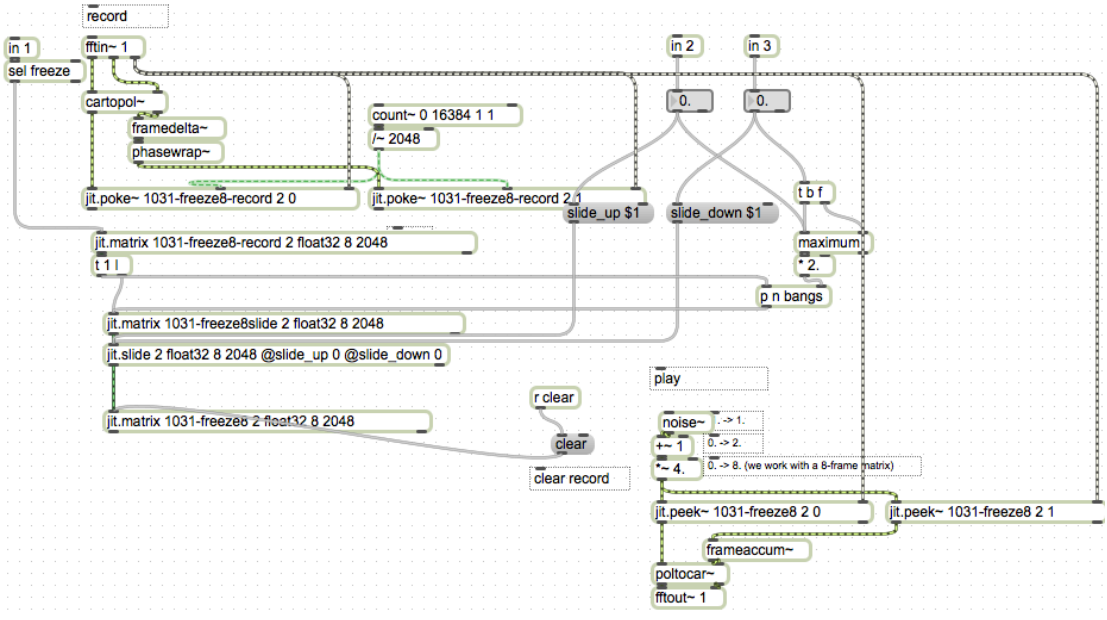


Figura 25 - Módulo Freeze

*baseado em Jean-François-Charles

5- Módulo filtro

O módulo de filtro fft serve neste caso para filtrar os “bins” desejados. Pode ser controlado manualmente mas neste caso está a ser activado por um “line”.

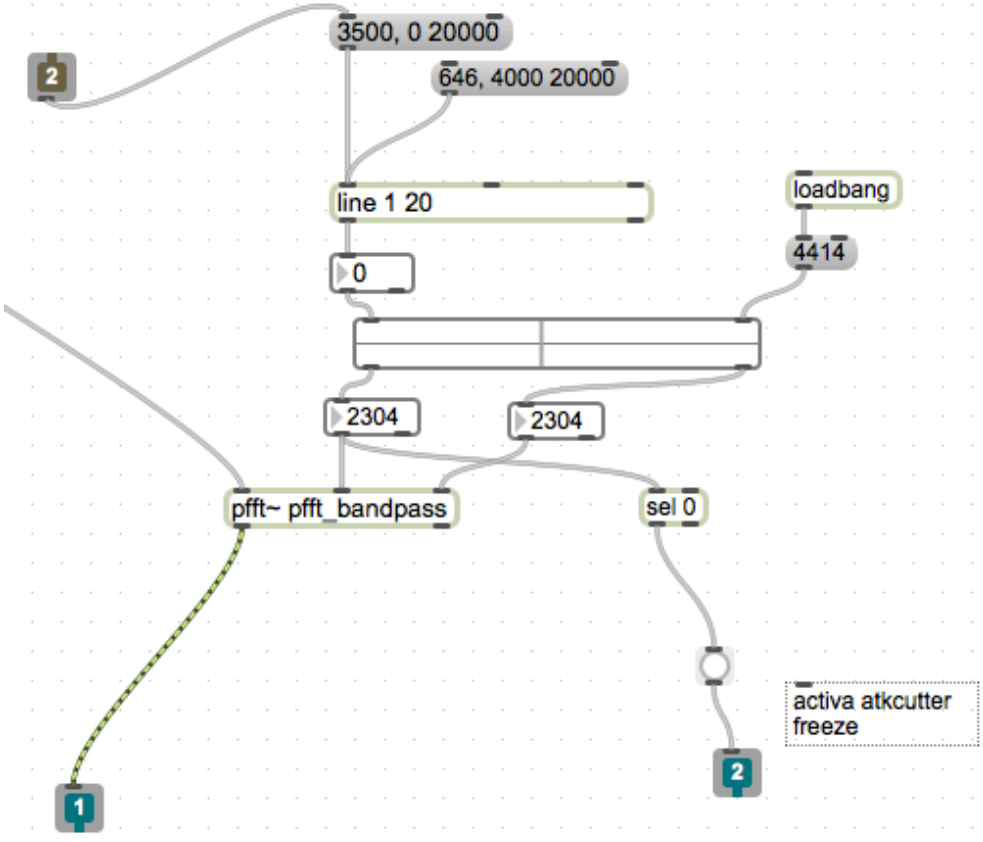


Figura 26 - Módulo filtro fft

6- Módulo Delay

Este módulo gera vários delays com tempos diferentes que vão ser activados consoante a amplitude recebida. Para fazer com que os delays se vão ligando e desligando utilizei o objecto "line".

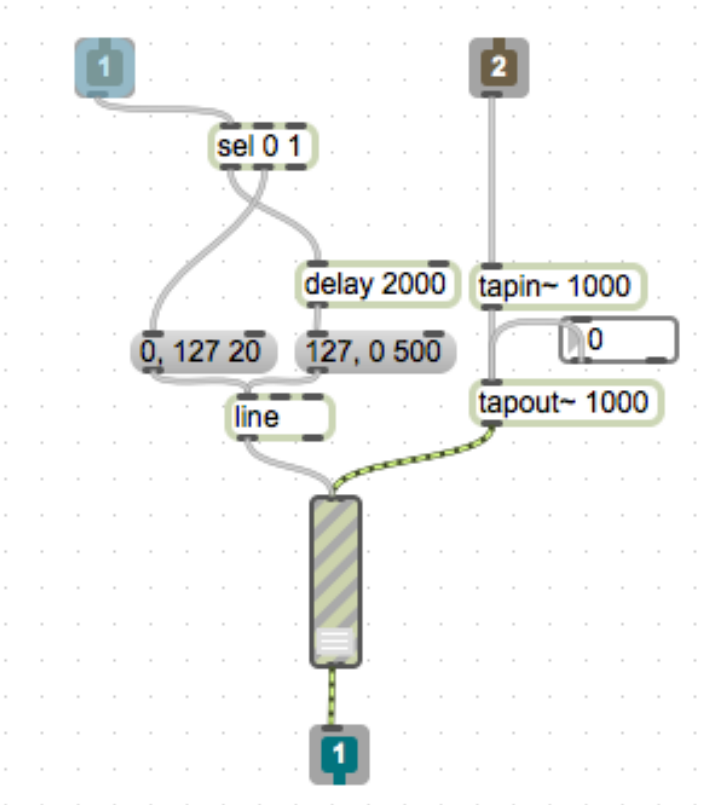


Figura 27 - Módulo Delay

Problemas e soluções

1- Graphic User interface

Um dos aspectos a ter em conta quando se desenvolve um programa é o consumo do cpu. Este deve ser reduzido ao máximo de forma a que o software corra o mais fluente possível.

Os objectos gráficos e objectos de sinal (p. Ex. “number~”) podem revelar-se ineficientes no que toca à gestão do cpu. Como tal, durante a realização do projecto, tentei reduzir ao máximo estes factores utilizando apenas o necessário para a medição de valores.

2- D.I.

Foram feitas algumas experiências com o intuito de reduzir o ruído próprio do hardware e para isso usei uma DI conectada entre a guitarra e a mixer. Após várias medições e comparações de sinal (com ou sem DI) cheguei há conclusão que, neste caso, era dispensável o uso deste aparelho pois não havia nenhuma melhoria perceptível.

Seleccionando 24db na DI, o ruído de facto baixava, mas baixava também o áudio fazendo com que tivesse que compensar na mixer acabando por ainda assim não obter o o sinal de entrada desejado.

Conclusão

A realização deste trabalho foi bastante frutífera tanto a nível investigacional como prático/criativo pois comecei a perceber a verdadeira potencialidade da geração e processamento de som num contexto de *live performing*.

Por ser um campo bastante vasto não consegui investigar e experimentar tudo o que queria mas sinto-me satisfeito com o produto final pois sinto que estou a tocar com outro músico, por muito rudimentar que seja, tal como eu.

Apesar de tudo penso que o meu objectivo está cumprido pois pretendia aprender bastante e conseguir um produto final que funcionasse.

Esta é sem dúvida uma das áreas com mais interesse para mim pois considero que faça parte da "vanguarda" da música e pretendo continuar a explorá-la daqui em diante.

Referências Bibliográficas

Audiosculpt:http://anasynth.ircam.fr/home/english/software/audiosculpt

Composição algorítmica: http://algocomp.blogspot.pt

http://www.essl.at/software.html

http://www.cgjennings.ca/toybox/lsystems/

http://guizzetti.ca/blog/2012/04/markov-models-or-the-future-is-now/

Estado da arte:

Audiosculpt:http://anasynth.ircam.fr/home/english/software/audiosculpt

Metasynth:http://www.uisoftware.com/MetaSynth/index.php

Spear: http://www.klingbeil.com/spear/

Estética dos sistemas musicais interactivos: Guy E. Garnett – "The aesthetics of interactive music systems"; Robert Rowe- "The aesthetics of interactive music systems"

http://www.music.buffalo.edu/lippe/pdfs/hongkong.pdf

Inspiração: Corail: http://www.edmundcampion.com/project_corail/index.html

Robert Rowe: https://files.nyu.edu/rr6/public/

Pedro Rebelo: https://www.youtube.com/watch?v=ETz6-UORsFc&spfreload=10

Processamento de áudio:

http://www.music.buffalo.edu/lippe/pdfs/hongkong.pdf

Jean François Charles- " A tutorial on Spectral Sound processing using Max Msp and Jitter"

https://cycling74.com/docs/max5/tutorials/msp-tut/mspchapter25.html

http://en.wikipedia.org/wiki/Audio_signal_processing

Programação Max Msp: cycling74.com/forums