

Felippe Brandão Barros

# **A improvisação de máquina como suporte pedagógico à prática da improvisação livre**

Belo Horizonte - Brasil

2021



Felippe Brandão Barros

## **A improvisação de máquina como suporte pedagógico à prática da improvisação livre**

Trabalho de Conclusão de Curso de Graduação em Música da Universidade Federal de Minas Gerais, como requisito parcial para obtenção do título de Licenciado em Música.

Universidade Federal de Minas Gerais - UFMG

Escola de Música

Trabalho de Conclusão de Curso

Orientador: Sérgio Freire Garcia

Belo Horizonte - Brasil

2021

Felippe Brandão Barros

A improvisação de máquina como suporte pedagógico à prática da improvisação livre/ Felipe Brandão Barros. – Belo Horizonte - Brasil, 2021-  
73p. : il. (algumas color.) ; 30 cm.

Orientador: Sérgio Freire Garcia

Trabalho de Conclusão de Curso  
Universidade Federal de Minas Gerais - UFMG  
Escola de Música  
Trabalho de Conclusão de Curso, 2021.

1. Improvisação de máquina. 2. Interação humano-computador. 3. Improvisação musical. I. Sérgio Freire Garcia. II. Universidade Federal de Minas Gerais. III. Faculdade de Música. IV. A improvisação de máquina como suporte pedagógica à prática da improvisação livre.

Felippe Brandão Barros

## **A improvisação de máquina como suporte pedagógico à prática da improvisação livre**

Trabalho de Conclusão de Curso de Graduação em Música da Universidade Federal de Minas Gerais, como requisito parcial para obtenção do título de Licenciado em Música.

Belo Horizonte - Brasil, 08 de setembro de 2021:

---

**Sérgio Freire Garcia**  
Orientador

---

**Jalver Machado Bethônico**  
Convidado 1

---

**José Henrique Padovani**  
Convidado 2

Belo Horizonte - Brasil  
2021



*A meus pais*





# Agradecimentos

Agradeço a Sérgio Freire pela orientação atenciosa, não somente neste trabalho, mas como por vários momentos do meu percurso de graduação, me ajudando a poupar diversos aparelhos de som em minhas experiências sonoras.

Aos professores José Padovani e Maurício Loureiro, com os quais percebi como computadores podem expandir nosso entendimento da expressividade musical, e como o som é tão essencial para nossa percepção do mundo.

Aos colegas do CEGeME, por me convencerem a mergulhar de cabeça no universo do software livre, e por me darem a confiança necessária para trilhar meu próprio caminho acadêmico.

Agradeço a tutoria de Fernando Rocha, Marco Scarassatti, Matthias Koole, e Nathália Fragoso, em suas provocações fundamentais em meu desenvolvimento como improvisador. Da mesma forma, aos amigos Caio Campos, Daniel Tamiatti, João Viana e Marc Wallach, com quem compartilhei tantos bons momentos improvisando e conversando no banquinho. E em especial, agradeço a Marcos Alves, Pedro Gilberto e Vanessa Bubgniak, pela amizade e receptividade em participar deste trabalho.



*“There’s no such thing as a wrong note.”*  
*(Art Tatum)*



# Lista de abreviaturas e siglas

MIDI	Musical Instrument Digital interface
IA	Inteligência Artificial
IRCAM	Institut de Recherche et Coordination Acoustique/Musique
PD	Pure Data
RMS	Root Mean Square
IOI	Inter Onset Interval
FM	Frequency Modulation
HCI	Human-Computer Interaction



# Sumário

	<b>Resumo</b> . . . . .	<b>15</b>
	<b>Abstract</b> . . . . .	<b>17</b>
<b>1</b>	<b>INTRODUÇÃO</b> . . . . .	<b>19</b>
<b>2</b>	<b>POR QUE IMPROVISAR COM UMA MÁQUINA?</b> . . . . .	<b>21</b>
<b>3</b>	<b>OBJETIVOS</b> . . . . .	<b>25</b>
3.1	Objetivo geral . . . . .	25
3.2	Objetivos específicos . . . . .	25
<b>4</b>	<b>FUNDAMENTAÇÃO TEÓRICA</b> . . . . .	<b>27</b>
4.1	M & Jam Factory (ZICARELLI, 1987) . . . . .	28
4.2	Robert Rowe - Machine Listening and Composing with Cypher (ROWE, 1992b) . . . . .	29
4.3	Improvised Music after 1950: Afrological and Eurological Perspectives (LEWIS, 1996) . . . . .	30
4.4	Interacting with Latter-day Musical Automata (LEWIS, 1999) . . . .	31
4.5	Rameses (COLEMAN, 1999) . . . . .	33
4.6	Machine Musicianship (ROWE, 2001) . . . . .	34
4.6.1	Edmund Campion - Natural Selection (1996) . . . . .	35
4.6.2	Amnon Wolman - New York (1998) . . . . .	35
4.6.3	Mari Kimura - Izquierda e Derecha (1998) . . . . .	36
4.7	OMaxist Dialectics: Capturing, Visualizing and Expanding Improvi- sations (LÉVY; BLOCH; ASSAYAG, 2012) . . . . .	36
4.8	Marcos Campello - Michel Temer Trio . . . . .	37
4.9	Ideias aproveitadas . . . . .	38
<b>5</b>	<b>DESENVOLVIMENTO DO PROTÓTIPO</b> . . . . .	<b>41</b>
5.1	Pure Data e bibliotecas . . . . .	41
5.2	Organização do patch . . . . .	42
5.3	Interface . . . . .	43
5.4	Escuta . . . . .	43
5.4.1	Ataque . . . . .	45
5.4.2	Altura . . . . .	46
5.4.3	Ritmo . . . . .	47

5.4.4	Timbre . . . . .	48
5.4.5	Dinâmica . . . . .	49
5.4.6	Legato . . . . .	51
<b>5.5</b>	<b>Pensamento . . . . .</b>	<b>51</b>
5.5.1	Vozes . . . . .	52
5.5.2	Homorritmo . . . . .	53
5.5.3	Contraponto . . . . .	55
5.5.4	Orquestração . . . . .	58
<b>5.6</b>	<b>Toque . . . . .</b>	<b>59</b>
5.6.1	Síntese . . . . .	59
5.6.2	Espacialização . . . . .	60
5.6.3	Reverberação . . . . .	61
5.6.4	MIDI . . . . .	61
<b>6</b>	<b>PRIMEIROS TESTES E IMPRESSÕES . . . . .</b>	<b>63</b>
<b>6.1</b>	<b>Testagem inicial . . . . .</b>	<b>63</b>
<b>6.2</b>	<b>Primeiras impressões . . . . .</b>	<b>64</b>
6.2.1	Comentários do autor . . . . .	65
<b>7</b>	<b>CONSIDERAÇÕES FINAIS . . . . .</b>	<b>67</b>
<b>7.1</b>	<b>Perspectivas futuras . . . . .</b>	<b>68</b>
	<b>Referências . . . . .</b>	<b>71</b>



# Resumo

Seria a improvisação musical uma atividade exclusivamente humana? Musicistas e computadores podem interagir criativamente? De que forma uma máquina pode auxiliar instrumentistas na prática da improvisação? O presente Trabalho de Conclusão de Curso pretende investigar diferentes metodologias, nos âmbitos criativo e técnico, de implementações computacionais capazes de criar autonomamente uma peça musical ao interagir com músicos em tempo real, por meio da elaboração de um protótipo e de uma avaliação preliminar de seu uso na prática da improvisação livre não-idiomática.

**Palavras chave:** improvisação de máquina, interação humano-computador, improvisação musical, pedagogia musical, improvisação livre.



# Abstract

Should improvisation be an exclusively human skill? Can musicians and computers interact creatively? How could a machine help instrumentalists in the practice of improvisation? This undergraduate thesis aims to investigate different methodologies, in the creative and technical spheres, of computational implementations capable of autonomously perform a musical piece in interaction with a musician in real time, through the development of a prototype and preliminary evaluation of its use in free improvisation practice.

**Keywords:** machine improvisation, human-computer interaction, musical improvisation, music pedagogy, free improvisation.



# 1 Introdução

A improvisação representa uma das atividades mais disseminadas no mundo musical, não se reduzindo a um gênero, um estilo, uma cultura ou um conjunto de técnicas, sendo difícil definir de forma unívoca uma atividade tão vasta. Dessa forma, “estrutura, sentido e contexto na improvisação musical surgem da análise, geração e manipulação, e transformação de símbolos sonoros de domínios específicos” (LEWIS, 1996, p. 218).

Portanto, em cada estilo musical improvisativo, seu entendimento difere a partir de suas estruturas internas, estando diretamente relacionado com o contexto cultural do qual faz parte. Ainda assim, um elemento em comum a todos esses estilos é a noção de que a improvisação é uma atividade criativa espontânea. Uma definição simples é a de que a “improvisação musical pode ser tomada como a concepção e performance simultânea da música” (DEAN, 1992, p. X).

Nem todos os gêneros musicais tratam a improvisação da mesma forma. Por vezes ela é regida por rígidos sistemas de organização musical ou cultural (bem exemplificados nas músicas indianas Hindustani e Carnática (BAILEY, 1993, p. 1–11)). Por tal razão, este trabalho estuda de modelos de máquinas criativas em somente em dois contextos específicos: a criação de música algorítmica realizada em tempo real e o da *improvisação livre* (aqui definida como práticas de improvisação não-idiomáticas). Essa escolha se deve ao fato de que máquinas improvisadoras já foram desenvolvidas em ambos contextos, além de serem culturalmente e musicalmente abertos a esse tipo de exploração.

Uma máquina improvisadora pode ser definida como um conjunto de algoritmos capazes de criar uma peça musical em tempo real. Essa criação pode ser gerada a partir de métodos de *aprendizado de máquina*, *reconhecimento de padrões* ou *composição algorítmica*. O resultado desses processos é o desenvolvimento do que chamamos de *autômato musical*, uma máquina capaz de realizar música por conta própria, a partir de sua organização interna, compreendendo uma “presença musical com personalidade e comportamento próprios, podendo apresentar diferentes níveis de dependência e interação com musicistas” (ROWE, 1992a, p. 8).

Uma vez que a performance da livre improvisação vem habitualmente sendo desenvolvida em práticas de conjunto, o meio carece de ferramentas para que sua prática possa ser realizada de forma desacompanhada. Portanto, o presente trabalho pretende uma revisão bibliográfica de diferentes modelos de máquinas improvisadoras no campo da música, com a elaboração de um protótipo, batizado *engenhoca* a ser analisado no contexto da prática individual da improvisação. Instrumentistas foram convidados a interagir com o modelo, e suas impressões foram avaliadas sob uma perspectiva pedagógica.

O trabalho de programação do modelo foi realizado na linguagem de programação visual Pure Data (PUCKETTE, 1997), distribuída gratuitamente sob a licença BSD(BSD..., 2004). Uma forte motivação do trabalho está em fornecer um exemplo de implementação de uma máquina improvisadora em uma linguagem de programação de fácil acesso, largamente utilizada por músicos, a partir de uma distribuição de código aberto, permitindo que este seja modificado, redistribuído e estendido de acordo com os interesses e necessidades de qualquer usuário do projeto.

## 2 Por que improvisar com uma máquina?

O interesse pela criação de máquinas capazes de improvisar música pode surgir de diferentes motivações ou necessidades. Além da vontade de expandir as possíveis interações entre instrumentistas e cantores para diálogos que incluem agentes não humanos, uma forte motivação está no interesse pela investigação de como um material improvisado surge e se sustenta em um determinado contexto musical, utilizando-se de uma máquina como meio para modelar este tipo de organização.

O desenvolvimento de programas de computador capazes de reconhecer e raciocinar sobre conceitos musicais humanos possibilita a criação de aplicações para a performance, a educação e a produção que ressoam e reforçam a natureza básica da musicalidade humana” (ROWE, 2001, p. 3).

Nesse sentido, a implementação de máquinas improvisadoras pode ser um meio de entender quais são os tipos de escolhas realizadas em tempo real no momento de uma improvisação, e como essas escolhas são combinadas ou contrastadas com as demais escolhas realizadas por outros participantes em um improviso. A atividade também nos permite investigar quais concepções podem vir a não se encaixar em nossa percepção de uma improvisação satisfatória.

Portanto, quando nossas máquinas improvisam musicalmente, elas nos permitem explorar como o sentido é transferido por meio do som. Improvisar é encontrar pontos de vista alternativos e aprender com o outro; improvisar com computadores nos permite uma forma de observar dentro desses e de outros processos fundamentais da interação. (LEWIS, 2018, p. 5)

Além disso, a criação de máquinas improvisadoras se justifica pela infinidade de uso dos resultados encontrados em contextos diversos dentro do campo musical: desde a pedagogia de um instrumento, de forma que estudantes poderiam desenvolver suas habilidades com a ajuda de musicistas virtuais, a até mesmo o uso de soluções encontradas pelas máquinas por parte de compositores, uma vez que o “pensamento” das máquinas ou dos improvisadores não necessariamente percorre os mesmos caminhos trilhados pelos humanos nos processos de composição musical em tempo diferido.

Portanto, este trabalho deseja vasculhar por respostas a questionamentos fundamentais no mundo da improvisação, observando diferentes técnicas de improvisação de máquina já propostas na literatura acadêmica para o desenvolvimento de um protótipo. O recorte aqui apresentado se foca no potencial pedagógico deste tipo de trabalho, acreditando que

tais modelos possam figurar como ferramentas de interesse para o amadurecimento de processos criativos de musicistas em qualquer nível técnico.

É importante salientar que o contexto musical específico de aplicação de uma máquina improvisadora deve ser levantado para a discussão, como nota Rowe em um enunciado que se mantém atual:

Não existe ainda uma máquina improvisadora generalista. Ainda que sistemas improvisadores não saibam o que ocorrerá em uma performance, eles geralmente são programados para participar na música de um estilo particular. Implementações de improvisadores de *bebop* soariam fora de contexto em uma performance de *free jazz*, e improvisadores da *nova-música* não podem tocar o *blues*. Em última análise, a pesquisa em progresso em análise/síntese de estilos talvez faça possível escrever uma máquina improvisadora capaz de reconhecer características estilísticas de uma música sendo tocada e adaptar sua contribuição de forma adequada. Ainda assim, improvisadores de um estilo específico já comprovaram seu mérito artístico, e ainda se beneficiam de uma análise de um contexto musical mesmo quando esse contexto é essencialmente restrito a priori. (ROWE, 2001, p. 278)

Portanto, este trabalho anseia levantar a discussão sobre uma possível aplicação de máquinas no contexto da *improvisação livre*, abordando uma motivação específica de tal cenário: uma vez que o contexto da prática da improvisação livre está comumente associada a uma atividade de conjunto (BAILEY, 1993), o campo carece de ferramentas para que sua prática seja realizada em contexto solo.

Improvisadores são, como regra, musicalmente gregários, preferindo trabalhar com outros músicos em qualquer combinação desde duplas até conjuntos inexplicavelmente grandes. Para a maioria das pessoas, improvisação, ainda que um veículo para expressão pessoal, é sobre tocar com outras pessoas e algumas das melhores oportunidades promovidas pela improvisação livre está na exploração de relações entre instrumentistas. Neste respeito improvisação solo não faz sentido algum. Apesar disso, em algum momento ou outro, a maioria dos improvisadores investiga as possibilidades de se tocar sozinho. (BAILEY, 1993, p. 105)

John Corbett demonstra que apesar de em um primeiro momento a ideia de improvisadores solistas poder ser questionada inclusive como improvisação, uma análise mais profunda da atividade pode revelar processos de interação que também estão presentes em improvisações em grupo, ainda que menos aparentes:

Dúvidas podem ser levantadas sobre se improvisação é realmente possível para um musicista solitário. Seria a interação uma necessidade absoluta? Seria tocar sozinho na verdade um tipo de composição ao invés de improvisação? Tocar sozinho pode ser uma atividade fascinante, rica, completa em si mesma. (...) Mas nela você não tem o prazer de escutar outras pessoas trabalhando os sons em conjunto, o que para mim é metade da diversão.



---

Uma maneira de pensar sobre solos como sendo genuinamente improvisados é tentar imaginar que o tipo de interação que se procura entre musicistas encontra-se confinada à relação entre o solista e o instrumento ou o contexto da performance ou a audiência. (CORBETT, 2016, p. 119)

Ainda de acordo com Bailey, “o nascimento da improvisação livre como um movimento musical coeso no início dos anos sessenta e sua prática subsequente” está fortemente relacionado ao “ímpeto pelo questionamento da linguagem musical e suas “regras” governantes” (BAILEY, 1993, p. 84). Dessa forma, é notável uma demanda recorrente pela dilatação das barreiras do estilo, não sendo surpresa, portanto, que investigações no mundo da improvisação “atingiriam um estágio em que um vasto alcance de técnicas encontravam-se em uso” já ao início dos anos 90 (DEAN, 1992, p. 187). Entretanto, ainda poderiam ser esperados avanços em técnicas de interação e suas aplicações em tempo real, a partir do uso de novas ferramentas tecnológicas (DEAN, 1992, p. 187).

Do ponto de vista pedagógico, o desenvolvimento da música improvisada, fora dos limites praticados em salas de concerto na primeira metade do século XX, também implicou a organização natural de novas metodologias de aprendizado, largamente caracterizadas pelo autodidatismo, como apresenta George Lewis:

À medida em que a confiança nos estilos recebidos e na codificação das “melhores práticas” musicais parecia diminuir, os papéis de professores e alunos foram radicalmente redefinidos, a ponto de não haver uma linha clara de demarcação entre eles. Os músicos ensinaram a si mesmos e uns aos outros não apenas como tocar, mas como ensinar. Além disso, essas comunidades autodidatas mais recentes desenvolveram variantes locais de uma prática de improvisação “aberta” ou “livre”, que turvou as fronteiras entre improvisação como performance, como investigação musical crítica, e como ativismo político e social, tudo no curso da pesquisa de novos sons e modos de comunicação. Tanto na América do Norte quanto na Europa, essas comunidades redefiniram o papel da pedagogia em si para incluir a crítica cultural e social. (LEWIS, 2007, p. 1)

A separação entre a pedagogia musical tradicional e novas metodologias de aprendizado aplicadas pelos improvisadores pode estar diretamente relacionada à dificuldade de se conceber como musicistas são capazes de criar ideias musicais em tempo real, sem a necessidade de se ater à memória ou a sistemas de composição (JOHNSON-LAIRD, 1991, p. 292–293). Diante de um cenário tão desafiador, a didática tradicional pode se colocar como uma barreira à prática de uma atividade criativa, indicando que as melhores soluções podem estar na busca pessoal por uma forma individual de expressividade, como coloca John Stevens, apud (BAILEY, 1993, p. 98):

“Improvisação é a base para aprender a tocar um instrumento musical. Mas o que ocorre usualmente? Você decide que quer um certo instrumento. Você compra o instrumento e então pensa, ‘Eu vou encontrar um professor, e quem sabe, em sete ou oito anos eu talvez seja capaz de tocar isso’. E

dessa forma você perde um monte de experiências musicais importantes. Estudar formalmente com um professor pode ser a maneira correta de alcançar certos objetivos específicos, mas fazer somente isso é uma forma muito distorcida de abordar um instrumento musical. Há de se perceber que a investigação pessoal de um instrumento por uma pessoa - sua exploração - é totalmente válida.”

Dada a grande importância que o contato individual com um instrumento possui no desenvolvimento de um músico, o presente projeto aborda o interesse em máquinas improvisadoras sob a perspectiva pedagógica, apontando para a possibilidade de que modelos sejam pensados como uma maneira de praticar a improvisação livre de maneira individual, fornecendo um contraponto a uma performance em tempo real, e garantindo a liberdade para que o músico tome suas próprias decisões de como reagir à improvisação da máquina.

Uma escolha técnica do projeto foi a implementação de um modelo de máquina improvisadora *transformativa*, tratando-se de um protótipo que “utiliza material musical já existente, aplicando-se métodos de transformação para produção de variações”, que “podem ou não ser reconhecidas como relacionadas ao material original” (ROWE, 1992a). O modelo projetado realiza tais transformações em tempo real, à medida que o material musical é tocado. Essa escolha se deve à simplicidade de implementação, bem como limitado uso de recursos computacionais: não é necessário armazenar arquivos de áudio ou dados, nem mesmo treinar o modelo, viabilizando que sua distribuição e execução seja compatível com a maior parte dos sistemas. Por fim, o método também se faz coerente para o contexto da improvisação livre, ao evitar assumir escolhas pré-determinadas para conteúdo musical.

## 3 Objetivos

### 3.1 Objetivo geral

Investigar diferentes metodologias, nos âmbitos criativo e técnico, de modelos de máquinas improvisadoras no campo da música, por meio da elaboração de um protótipo e de uma avaliação preliminar de seu uso como ferramenta pedagógica na prática individual da improvisação livre não-idiomática.

### 3.2 Objetivos específicos

1. Realizar uma revisão bibliográfica de modelos de *improvisação de máquina* existentes, de forma a compreender diferentes metodologias computacionais de cada método estudado, refletindo sobre suas potencialidades em diferentes contextos musicais.
2. Desenvolver um sistema de máquina improvisadora própria, dentro do contexto musical da improvisação livre.
3. Analisar potencialidades de uso do protótipo desenvolvido como ferramenta pedagógica na prática da improvisação livre.



## 4 Fundamentação teórica

A elaboração de modelos de máquinas criativas já se demonstra um campo maduro dentro da pesquisa musical. Ainda assim, muitos trabalhos concentram-se em formas de improvisação *idiomáticas*, como em (GILLICK; TANG; KELLER, 2010) e (CHORDIA; RAE, 2010) bem como na composição dentro de estilos específicos (COPE, 1992). Entretanto, para o desenvolvimento de uma máquina dentro do contexto da improvisação livre em tempo real, outros desafios são impostos, onde técnicas baseadas em *gramáticas* ou *aprendizado de máquina* podem apresentar resultados insatisfatórios, e possivelmente enviesados, devido à natureza desregrada e extremamente variada do estilo. Portanto, a realização do trabalho partiu de uma revisão bibliográfica que buscasse atender a pluraridade da improvisação livre.

Trabalhos de modelagem computacional da criatividade costumam enxergar essa capacidade humana como equivalente a qualquer outra, sem entretanto acreditar que isso seja depreciativo à valorização do trabalho de artistas. Pelo contrário, tal prática na verdade poderia representar um reconhecimento do fascínio por essa capacidade humana, que merece ser melhor estudada.

Criatividade não é mágica. É um aspecto de uma inteligência humana normal, não é uma faculdade especial garantida a uma pequena elite. Existem três formas: combinatorial, exploratória, e transformativa. Todas as três podem ser modeladas por uma IA - em alguns casos, com resultados impressionantes. Técnicas de IA sustentam vários tipos de arte computacional. Se computadores podem “realmente” ser criativos não é uma questão científica, mas filosófica, para a qual não há uma resposta clara. Mas nós temos sim o início de um entendimento científico da criatividade. (BODEN, 2009)

Nesse sentido, é importante ter cuidado ao admitir que um computador seja capaz de uma criatividade genuína, bem como de verdadeira autonomia em suas decisões, como apontam Majid al-Rifaie e Bishop:

...sugerimos que atos criativos totalmente intencionais, engajados no processo de genuinamente compreender o mundo, requerem tanto um cérebro totalmente funcional quanto um corpo totalmente funcional; deslocando então a questão dos requerimentos de uma “criatividade cognitiva genuína” para a definição de funcionalidades dos processos por meio dos quais o cérebro e o corpo interagem com o mundo e sociedade. (AL-RIFAIE; BISHOP, 2015)

Portanto, ao se tratar de definições de criatividade expostas nesse capítulo, as ideias aqui expressas devem ser levadas em consideração, evitando descontentamentos em relação à humanização de funcionalidades computacionais.

A maior parte dos trabalhos estudados são de implementações, na prática, de modelos de máquinas improvisadoras no campo da música. Os resumos a seguir foram realizados a partir de tradução própria, bem como inserção de opiniões do autor. Os trabalhos incluídos serão listados a seguir:

1. David Zicarelli - M and Jam Factory (1987)
2. Robert Rowe - Cypher (1992)
3. George E. Lewis - Improvised Music after 1950: Afrological and Eurological Perspectives (1996)
4. George E. Lewis - Interacting with Latter-day Musical Automata (1999)
5. Steve Coleman - Rameses (1999)
6. Robert Rowe - Machine Musicianship (2001):
  - 6.1 Edmund Campion - Natural Selection (1996)
  - 6.2 Amnon Wolman - New York (1998)
  - 6.3 Mari Kimura - Izquierda e Derecha (1998)
7. Vários autores - OMaxist Dialectics: Capturing, Visualizing and Expanding Improvisations (2012)
8. Marcos Campello - Michel Temer Trio (2019)

## 4.1 M & Jam Factory (ZICARELLI, 1987)

M and Jam Factory são ambientes de composição e performance que utilizam interfaces gráficas e gestuais para trabalhar com MIDI. Diferentes métodos permitem um grande controle dos parâmetros geradores do fluxo sonoro, que podem ser modificados em tempo real, à medida em que se compõe ou improvisa com o sistema. Dentre ambos, *Jam Factory* se apresenta como o mais voltado para uma interação a partir da improvisação, e portanto lhe será dado maior atenção neste resumo.

*Jam Factory* utiliza *cadeias de Markov* em combinação com diferentes algoritmos capazes de escutar entrada MIDI e improvisar respostas em diferentes níveis em relação ao material original. É possível modificar a resposta ao se tocar mais notas, ou alterando diretamente parâmetros enquanto a resposta ocorre. É perceptível uma preocupação com o controle do musicista sobre o sistema. Apesar disso, é possível interagir sem necessariamente reger todo o controle sobre as respostas.

Para variações no material melódico, o sistema reordena as sequências tocadas, em um processo controlado por probabilidades nas cadeias Markov. Essa escolha garante controle ao musicista, bem como mantém a tonalidade do que é tocado. O autor aponta que a “escolha de notas é mais importante do que sua ordem”, e como indica (YAVELOW, 1987), “mudanças sutis em ordenamentos de alturas, como permitidas por tabelas de transição, é uma boa alternativa ao *loop*”.

O controle rítmico de *Jam Factory* é realizado através de valores de duração quantizados, respeitando um pulso, bem como métricas tradicionais. Portanto, todos os valores rítmicos são frações de uma figura relativa, sendo o valor mínimo sua divisão em 24 partes. Ainda é possível tocar valores específicos através de um teclado MIDI, novamente garantindo um controle composicional do resultado.

A organização “orquestral” de *Jam Factory* consiste em quatro “instrumentistas” possuindo comportamentos próprios, garantidos por suas próprias tabelas de transição. Variações das saídas de cada um também podem ser controladas pelo musicista. Para cada nota tocada no sistema, são escolhidas diferentes tabelas de transição, o que garante um maior nível de variações das saídas.

Considerados os “avôs” das máquinas improvisadoras, é observável que os sistemas apresentam uma grande consideração pelo controle da saída, como uma forma de permitir um controle composicional sobre os resultados improvisados pela máquina.

## 4.2 Robert Rowe - *Machine Listening and Composing with Cypher* (ROWE, 1992b)

*Cypher*, assim como vários outros sistemas interativos na área da música, subdivide-se entre algoritmos de escuta e algoritmos de reprodução. A escuta é realizada somente a partir do fluxo de MIDI, assim como outros programas da mesma época.

O “escutador” classifica funcionalidades da entrada e seu comportamento no tempo, enviando mensagens que comunicam essa análise para o tocador. Funcionalidades classificadas incluem *velocidade*, *densidade*, *dinâmica*, *harmonia*, e *ritmo*. Um usuário do *Cypher* pode configurar o componente “tocador” para reagir a essas mensagens, de forma que a reação se faz como a execução de métodos de composição produzindo música em resposta. (p. 43)

O programa foi elaborado para diversas utilizações distintas, podendo servir desde como um acompanhador em um ambiente com uma partitura definida, bem como companhia para uma improvisação totalmente espontânea. Além disso, é capaz de gerar música sem que haja nenhum influxo de um musicista, a partir tanto de material escutado no passado, como materiais novos, totalmente gerados por seus métodos de composição.

O processo de escuta analisa o material musical tocado e determina sua localização em um espaço de possíveis classificações, agrupando por exemplo “acordes agudos e intensos”, ou “notas melódicas graves e suaves”. Esse espaço é organizado em uma matrix de seis dimensões: “*densidade vertical, velocidade de ataque, intensidade, registro, duração e harmonia*”. Esses classificadores são determinados a partir de limiares arbitrários, já definidos no programa.

O “tocador” possui um número de métodos de resposta, como tocar sequências, iniciar algoritmos de composição, ou transformar blocos de eventos. Coleções específicas de métodos são realizadas de acordo com conexões feitas pelo usuário entre o “escutador” e o “tocador”. Estabelecer tais conexões significa que sempre que uma mensagem particular for enviada, o método em que é conectada deverá ser invocado, e a saída do método é enviada aos sintetizadores (pag. 44).

O musicista possui a liberdade para “configurar as maneiras pelas quais o ‘tocador’ irá responder” às mensagens enviadas pelo componente de escuta. Dessa forma, o sistema não reproduzirá sua saída necessariamente de acordo com as escolhas estéticas de Rowe, podendo ser modificado da forma que outros musicistas preferirem para uma performance.

O “tocador” permite três possibilidades de criação com o material enviado pelo “escutador”: “transformação do material MIDI; algoritmos que produzem material escolhidos dentre uma variedade de estilos texturais e gestuais; performance de uma biblioteca de sequências”, que podem variar desde pequenas frases até partituras completas.

De longe, o método mais usado na seção de composição envolve a transformação de material, realizada pelo encadeamento direto de pequenos módulos, em que cada um realiza modificações simples e consistentes no material que recebe de entrada. A ação desses módulos é cumulativa; se mais de um é utilizado, eles são aplicados em série, com a saída de uma operação sendo passada para a entrada do próximo. Mesmo sendo a ação de cada módulo separadamente simples e fácil de compreender, cadeias mais longas de transformações podem montar materiais complexos, ainda que deterministicamente derivados da entrada (pag. 45-46).

Isso não significa que o programa depende de um musicista para funcionar, e de forma contrária, Rowe afirma que “Cypher é um entusiasmado solista; ele sempre considera a ausência de entrada como um sinal para tocar mais”. Quando toca sozinho, o sistema continua a utilizar seus métodos transformativos para a geração de novos materiais.

### 4.3 Improvised Music after 1950: Afrological and Eurological Perspectives (LEWIS, 1996)

George Lewis realiza um apanhado histórico e crítico das abordagens à improvisação musical que ocorreram na Europa e nos EUA durante o século XX, defendendo que a música



afro-americana, notadamente inspirada por Charlie Parker, se trata uma metodologia *afrológica*, enquanto sistemas de improvisação desenvolvidos por compositores de concerto largamente influenciados por John Cage, possuem uma abordagem *eurológica*.

Observa, assim como Derek Bailey (BAILEY, 1993), que a “música de concerto possuiu um hiato de quase dois séculos com a improvisação”, tendo reduzido, desde o final do século XVIII, a liberdade para intérpretes realizarem investigações expressivas mais amplas, sendo uma notável exceção a escola de órgão improvisado francês. Essa liberdade aparece de forma representativa na música afro-americana, dissociada do ambiente acadêmico da música de concerto no século XX, que passa a influenciar compositores de tradição europeia, que entretanto, apresentam “forte despedimento dos princípios das formas improvisativas afro-americanas”.

O autor nota que a ênfase radical de Cage em espontaneidade e singularidade aparece entre oito a dez anos *após* as inovações do *bebop*. E enquanto a literatura a respeito da “música experimental” do século XX raramente nota a importância de improvisadores como “Parker, Coltrane, Taylor e Coleman”, páginas inteiras são dedicadas a descrições de improvisadores de linha *eurológica*, mesmo quando sua documentação em áudio não é representativa.

Um ponto abordado por Lewis, que deve ser diretamente levado em consideração para o desenvolvimento de uma máquina improvisadora, está na refutação da crença de que a improvisação trabalha necessariamente com a repetição de *frases* e *motivos* que o intérprete já conhece. Como citado no texto de Lewis, Johnson-Laird afirma que alguém teria de ter inventado os motivos tocados pelos demais, e vai além:

A análise do corpo de improvisações de um músico nos permite identificar que muitas frases ocorrem somente uma vez. Um cético poderia dizer que a análise de cada uma das improvisações feitas por um músico falsificaria essa afirmação. Ainda assim, existiriam muitas outras possíveis frases características do estilo de um músico mesmo se elas não ocorreram nesse *corpus*. Terceiro, o trabalho de guardar na memória um número suficiente de motivos para garantir a improvisação de solos inteiros é por si só muito grande para ser prático. (...) qualquer músico de jazz competente te dirá que é bem mais fácil criar novas frases do que tentar aprender um vasto repertório delas para uso nos solos (JOHNSON-LAIRD, 1991, p. 293).

Isso nos leva a questionar se máquinas improvisadoras deveriam possuir frases ou motivos em sua *memória* como método para utilização no improviso.

## 4.4 Interacting with Latter-day Musical Automata (LEWIS, 1999)

Neste segundo artigo, Lewis apresenta seu trabalho na confluência entre computadores e músicos improvisadores, em um sistema que envolve uma interação destes nos

níveis de organização e estruturação de uma performance em tempo real. O texto busca explorar implicações estéticas, filosóficas, culturais e sociais deste tipo de ocupação.

Lewis considera que o papel da improvisação não pode ser “composição em tempo real”, uma vez que isso subjulgaria o ato de improvisar ao de compôr, e da mesma forma, não pode ocupar o mesmo espaço de um intérprete, já que não existe uma “peça” a ser tocada. Reservando à música improvisada um lugar à parte da música tradicionalmente notada ou interpretada, ao acreditar que para ela deve ser destinada uma categoria diferente, e para o improvisador deveria ser destinada uma categoria ao lado do “*compositor, performer, intérprete, psicoacústico*, e do *teórico*”.

Lewis observa que a “música improvisada privilegia o papel do som como meio para história e identidade cultural, retirando a ênfase nas noções clássicas europeias de forma e estrutura em favor da troca de narrativas culturais e sociais”, e portanto, seu trabalho “se atentará a este processo”.

*Voyager* se trata, à época, de sua última peça “desenvolvida para funcionar como um ambiente musical interativo não-hirerárquico que privilegia a música improvisada”:

Um programa de computador analisa aspectos da performance de um improvisador em tempo real, utilizando esta análise para guiar um programa de composição automática que gera respostas complexas para o que é tocado. Em *Voyager*, o sistema computacional não é um instrumento, e portanto não pode ser controlado por um performer (p. 103).

O sistema utiliza combinações de instrumentos para dezesseis “instrumentistas” a partir do *sampler* Proteus/2 desenvolvido pela Emu Systems, elaborando uma orquestra virtual. Especificações de comportamento determinam quais combinações instrumentais ocorrerão, e como serão agrupadas. *Voyager* incorpora a ideia de orquestra inteligente, no sentido de que cada instrumento pode realizar escolhas que modificam o nível global, para tal, importa para o trabalho noções de uma orquestra de gamelão javanês.

Diferentes descritores extraem informação musical do que é tocado pelo músico, como “*volume, velocidade, duração do som, duração entre onsets, uso de oitavas e registro, tamanho dos intervalos, alturas utilizadas, variação do volume e onsets, frequência de silêncio, técnicas de articulação*”, que juntas de outras características são usadas para guiar o comportamento do programa.

Outras escolhas musicais importantes são geradas internamente por números aleatórios. Este processo promove muito da “personalidade” do sistema, incluindo: harmonia e melodia, orquestração, ornamentação, ritmo, andamento, transposição e decisões de comportamento internas. Essas opções podem caracterizar como e se haverá reação da eletrônica, quão rápido os dados de um parâmetro serão alterados ou qual parâmetro será alterado (p. 104).

Mesmo sem entrada, o sistema é capaz de gerar especificações para todos os parâmetros. Portanto, não há hierarquia entre o performer e o sistema. Os valores aleatórios são gerados por filtragens a partir de diferentes porcentagens da saída de geradores de ruído branco. Tal escolha garante independência ao sistema, e separa o trabalho de interações baseadas em processos de transformação, que funcionam a partir de estímulo-resposta.

Por fim, defende a ideia de uma computação musical não-eurocêntrica, que “não pode ser considerada historicamente e culturalmente como uma derivação da música contemporânea europeia”. A improvisação para ele não se trata de uma narrativa sobre “composição”, e está fora dos sistemas teóricos e práticas culturais da música europeia. Nesse sentido, Lewis substitui a noção de *controladores* por *envolvedores* e *possibilitadores*, permitindo uma *agência* da eletrônica. E da mesma forma, acredita em uma metodologia intuitiva para o desenvolvimento de programas interativos.

## 4.5 Rameses (COLEMAN, 1999)

*Rameses* trata-se de um projeto de improvisação com máquinas desenvolvido para o contexto de música de câmara, onde um computador interage com o saxofonista autor do projeto, Steve Coleman e uma banda (Five Elements) de cinco integrantes, com a formação: saxofone tenor, trompete, guitarra, baixo elétrico e bateria. O trabalho foi desenvolvido no centro de informática do IRCAM. O programa é fruto da prática e formação de Steve Coleman como saxofonista e compositor, integrando ideias rítmicas, melódicas e harmônicas desenvolvidas ao longo dos anos em sua prática musical.

Coleman se inspirou no pioneirismo de George Lewis no desenvolvimento de sistemas de improvisação com máquinas. A elaboração do projeto foi gradual, tendo sido portado para diferentes sistemas e linguagens, e se desenvolvido junto dos avanços tecnológicos da época, até que fosse permitido a reprodução em tempo real, tendo passado pelos sistemas *Commodore 64*, *Atari St* e por fim *Macintosh*, portado da linguagem de programação *Forth* para *Max*.

O sistema não possui programação musical pré-determinada, criando tudo em tempo real, improvisando todas as suas ideias, de forma que nunca se repete. Os músicos, conseqüentemente, não sabem o que o sistema irá criar antes do início de uma performance, conhecendo somente as grandes linhas estilísticas que servirão de fundamento à criação.

*Rameses* é capaz de escutar acordes e criar melodias improvisadas a partir deles, levando em consideração sua modalidade, buscando equilíbrio e consonância com o modo tocado. As respostas são geradas por processos transformativos, análises de eixos de simetrias, sequências de Fibonacci, e ideias harmônicas calculadas a partir de decanatos

egípcios e conceitos inspirados pelo compositor Henry Threadgill, também de Chicago, como Coleman e Lewis.

(Rameses) é capaz de elaborar suas próprias progressões harmônicas a partir de um sistema derivado dos decantos cinéticos (uma interpretação de grupos de estrelas no céu do Egito Antigo). O programa então anexa essas células harmônicas simultaneamente às células criadas com a escuta da improvisação dos musicistas. Este método produz na tela pistas distintas que detalham uma informação que se move ao mesmo tempo, mas em andamentos diferentes. Os músicos podem escolher a trilha que desejam seguir, como podem escolher reagir a mais de uma pista ao mesmo tempo. O que significa que a música é realizada em mais de uma dimensão. (...) É possível submeter uma lista de pontos temporais a *Rameses* (datas e localizações geográficas), por meio do qual ele criará uma série rítmica baseada nas fases da lua, na posição do sol, nas posições dos planetas e das estrelas, a posição do sol nascente, o meio do céu, os equinócios de primavera e outono, os solstícios de verão e inverno, levando sempre em consideração os cálculos do dia e da noite.

Os sons tocados pelo sistema são gerados por síntese de formantes (sons de vogais), análises espectrais e amostragens. Também estão presentes uma bateria e um coro de tambores africanos virtuais, o que indica uma forte inserção estética na música afro-americana.

## 4.6 Machine Musicianship (ROWE, 2001)

O sétimo capítulo do livro *Machine Musicianship*, de Robert Rowe, é dedicado à problematização e análise de soluções encontradas na implementação de máquinas interativas no campo da improvisação musical. Rowe acredita que:

A improvisação interativa representa o maior desafio para musicalidade de máquina. Aqui, a máquina deve contribuir convincentemente com uma voz em um ambiente totalmente não-estruturado e imprevisível. Justamente por isso, é na improvisação que algum tipo de musicalidade de máquina talvez seja a mais importante: porque o material e seu desenvolvimento é desconhecido de antemão, o programa deve ser capaz de apoiar-se em algum tipo de senso comum musical como forma de derivar estrutura do que escuta, e impôr estrutura em sua resposta. (p. 277)

O autor apresenta, no capítulo, implementações de diferentes sistemas musicais interativos para a improvisação, dentre os quais são apresentadas perspectivas *selecionadas*, *gerativas*, *transformadoras*, e dentro dos paradigmas de *tocadores* e de *ensembles*. A seguir, segue um resumo de alguns trabalhos apresentados por Rowe, utilizando também, quando disponíveis, fontes externas ao livro. Quando mais de uma fonte for utilizada como referência, as duas serão notadas. Caso contrário, toda a subseção é uma referência direta ao apresentado no texto de Robert Rowe (ROWE, 2001).

#### 4.6.1 Edmund Campion - Natural Selection (1996)

*Natural Selection* se trata de um ambiente para performance de música interativa em tempo real, originalmente programado para performance com piano MIDI na linguagem *Max*, rodando em um computador *Macintosh*. Vários parâmetros podem ser alterados para adaptar o sistema a diferentes situações, e seu autor garante que “todas as performances realizadas são sempre diferentes”, mesmo com as restrições intrínsecas do sistema (CAMPION, 1996).

Apesar da existência de algoritmos sequenciais, ativados por ações tomadas pelo pianista, a peça não se organiza de forma linear, uma vez que o programa pode decidir se usará ou não as “influências” alimentadas pelo toque do pianista. Os processos incitados pelo pianista também podem se organizar a partir de algoritmos generativos (ROWE, 2001). Tanto o músico quanto o sistema se influenciam, em um processo não hierárquico. Para a escuta de máquina, “em adição à informação de alturas, o computador também analisa e é influenciado por velocidades, intervalos entre eventos, divisões de teclas, etc” (CAMPION, 1996).

A possibilidade de influência sobre a máquina parte do reconhecimento de 64 combinações de conjuntos de três notas, bem como 24 combinações de conjuntos de seis notas. Entretanto, ao ser reconhecido um padrão, o sistema ainda possui a liberdade de escolher não ser influenciado por tal. O sistema é capaz de se adaptar aos interesses do musicista com quem interage, uma vez que ao instrumentista é dada a liberdade de gravar quantos materiais quiser antes de uma performance, mas que somente serão ativadas conforme as escolhas do programa, bem como serão transformadas por algum processo algorítmico (ROWE, 2001).

#### 4.6.2 Amnon Wolman - New York (1998)

Um exemplo apresentado por Rowe de métodos transformativos de interação computacional com a improvisação é a peça *New York*, estreada pela pianista Ursula Oppens. O programa é pensado para a interação, no palco, de dois pianos, novamente equipados com entrada e saída MIDI, em que a pianista recebe como resposta a interação do computador expressa no segundo piano.

Assim como o sistema implementado por George Lewis em *Voyager*, o programa de Wolman trabalha a partir de geradores de números aleatórios. Entretanto, aqui esses valores são utilizados para aplicar variações de duração, intensidade e altura nos valores tocados pela pianista.

Também implementado em *Max*, o sistema utiliza o objeto *urn*, que seleciona valores aleatórios antes que um deles se repita, como uma urna, que posteriormente são escalados a valores em dentro de um maior alcance, que por sua vez controlam a ativação

do novos geradores de números aleatórios, novamente escalados, para sua adequação como valores de intensidade e duração. Constantes garantem que a saída fique dentro da extensão do piano, bem como multiplicam os valores de altura para designar a saída.

#### 4.6.3 Mari Kimura - Izquierda e Derecha (1998)

Mari Kimura, violinista virtuose e compositora de música interativa, desenvolve na peça *Izquierda e Derecha* um diálogo entre violino e piano MIDI, em uma relação horizontal, mas com momentos de controle da performer sobre o sistema.

A peça se organiza pela escolha de diferentes algoritmos pré-programados, cada qual capaz de desenvolver uma textura musical diferente, como escalas cromáticas e acordes baseados em escalas diatônicas, ambos influenciados por parâmetros da improvisação desenvolvida por Kimura em tempo real.

### 4.7 OMaxist Dialectics: Capturing, Visualizing and Expanding Improvisations (LÉVY; BLOCH; ASSAYAG, 2012)

OMax é um programa capaz de improvisar com um musicista em tempo real, utilizando algoritmos de escuta e segmentação de áudio e MIDI, de forma a extrair informações musicais relevantes para a modelagem de um “mapa” sonoro de uma improvisação, que é simultaneamente acessado pelo sistema para reprodução de recombinações e variações do material tocado. O processo dá origem, no termo dos autores, a ‘clones’ do próprio músico (uma vez que o resultado produzido utiliza o próprio som de entrada como material de saída).

OMax não possui conhecimentos ou regras a priori, e também não realiza sua análise a partir de nenhuma teoria musical específica (é tido como *agnóstico*). Daí sua versatilidade e adaptabilidade. (LÉVY, 2012)

O algoritmo para criação do mapa sonoro em OMax foi batizado *Factor Oracle*, tratando-se resumidamente de um grafo que “incrementalmente reconhece fatores repetidos (padrões e subpadrões) de qualquer *string* simbólica”. O mapa criado “permite OMax navegar pelos dados para criação de vários ‘clones’ do músico que alimenta o sistema. Esses clones são recombinações do discurso original justificados pelo modelo e realizado por recortes e colagens do material em tempo real”. Os autores acreditam que tal processo enfatiza os “efeitos de memória e autorreferência” encontrados em contextos de improvisação livre e jazz.

Tal estrutura é organizada sob a ideia de *alfabetos*, e de forma similar ao sistema implementado em *Cypher*, promove agrupamentos em um “espaço Euclidiano multidimensional”.

mensional que permite ao sistema identificar as unidades simbólicas da performance dos musicistas”.

Regiões de espaço de timbre minuciosamente exploradas pelo músico terão mais agrupamentos do que outras regiões, que não foram trazidas durante o improviso. Por outro lado, valores de classes de altura e vetores espectrais quantizados constituem um alfabeto fixo e pré-determinado”.

O sistema representa uma implementação para a produção de improvisos em que musicistas desejarem uma “resposta” muito similar às “perguntas” como retorno sonoro, uma vez que a agência da eletrônica é diretamente dependente do escopo sonoro trazido para a performance.

## 4.8 Marcos Campello - Michel Temer Trio

Dada a ausência de material escrito publicado sobre o trabalho, o autor entrou diretamente em contato com Marcos Campello, organizando uma entrevista, via *e-mail*, sobre o projeto *Michel Temer Trio*. A transcrição das perguntas e respostas vem a seguir.

1. O que lhe incita à prática da improvisação livre?

“A vontade de criar sempre de maneiras distintas e a curiosidade acerca do resultado disso.”

2. Quais são suas motivações para tocar com uma máquina improvisadora?

“A vontade de me expor ao desconhecido que existe nisso e a curiosidade sobre que tipos de relação podem surgir daí, sobre como isso pode se retroalimentar.”

3. De onde surgiu a ideia para o projeto *Michel Temer Trio*?

“Desde muito tempo quis desenvolver máquinas de improvisação interativas. Com o impedimento da Dilma e a subida do Temer me veio a idéia de usar a figura dele - que pra mim é um arquétipo de um determinado tipo de elite brasileira - como material de base para desenvolver um trio.”

4. Como se deu a implementação da criatividade do *Michel*?

“Eu gosto de programar de forma simples pequenas máquinas e processos e então, pela recombinação delas, chegar talvez a resultados mais complexos. Basicamente o Michel é uma combinação de maquininhas de aleatoriedade com controles restritivos para que ele toque ao mesmo tempo de forma musical e solta. Por outro lado, existem várias camadas de processo que ocorrem paralelamente, sendo que apenas uma delas age de modo efetivo, até que uma outra camada seja trazida à tona e jogue a que estava atuando para esse banco de reservas.”

#### 5. Como foi elaborada a interação da máquina com os músicos?

“O Michel ouve apenas ritmos, então idealmente ele recebe um canal de baixo e outro de guitarra e responde, seja atacando/respondendo de forma pontual e imediata, seja colando num dos instrumentos e seguindo, como se fosse uma sombra. Ao vivo nunca conseguimos alimentá-lo com guitarra e baixo porque o Felipe Zenícola sempre esquecia de levar outro cabo pra isso. Existe uma outra implementação, mais complexa, que está até ligada à minha pesquisa de mestrado, que é a criação de bancos de ideias musicais em tempo real - no caso apenas rítmicas. A idéia por trás é que talvez isso traga uma identidade mais profunda pra improvisações, trabalhando com recorrências e propostas do tipo. Seria como a memória do Michel. Mas por enquanto ainda não terminei de programar isso, é um trabalho mais complicado.”

#### 6. Como você enxerga as perspectivas da improvisação com máquinas?

“Eu adoro improvisar com máquinas. E com gente também. Acho que a máquina reflete muito de quem a criou, então, na minha cabeça, o que acontece de fato é que você está como que improvisando com um porta-voz, um representante de alguém. Isso me lembra aqueles tecladinhos que tem uns ritmos pra você tocar por cima, e aí você escolhe o de ‘samba’ e percebe que não é bem um samba. . . e é claro, porque aquele teclado está ali como emissário de um fabricante chinês, japonês ou norte-americano que não sabe de samba e nem chamou ninguém que soubesse pra dirigir essa parte. Então essas máquinas são em si afirmações de visões de mundo, e isso é muito interessante, ainda mais nesse nível de máquinas de improvisação. E a gente cada vez mais fala com máquinas, até conversa, acena, e realiza vários tipos de interação, então acho que existe realmente a tendência de que isso aumente. Até porque, em termos de grana, você já tem faz tempo coisas como shows usando no palco figuras holográficas de pessoas falecidas, filmes usando inteligência artificial para simular a voz de pessoas falecidas lendo textos que nunca leram. . . tem esse lado necrófilo que é barra pesada, que levanta sérias questões éticas sobre o que é ou pode propriedade intelectual ou material, e sobre os limites para se dispor disso. Daria pra fazer modelagens em IA de John Coltrane, Rosinha de Valença, Edison Machado e Charles Mingus e, dependendo da grana investida pra pagar os donos desses direitos, talvez lançar como disco usando os nomes deles. Tem esse lado bizarro. E também o de inibir o contato entre pessoas ou atrapalhar tradições que são passadas no contato, no aprendizado - dá pra usar como exemplo os play-alongs de choro ou jazz. Você fica ali tocando sobre uma base fixa, que não reage, não muda nunca. . . e esses tipos música são o contrário disso. Mas, enfim, tocar com máquina e com gente, pra equilibrar as coisas.”

## 4.9 Ideias aproveitadas

Como todo trabalho artístico e acadêmico, a realização do protótipo de *engenhoca* possui, ainda que inconscientemente, alguma relação com modelos já implementados anteriormente. Alguns exemplos dessa ressonância entre ideias anteriores e o realizado no trabalho serão apresentadas a seguir.

Assim como a maioria dos modelos estudados, a escuta do que é tocado pelo instrumentista que interage com *engenhoca* é essencial no funcionamento do sistema,



sendo a produção das ideias tocadas pela máquina realizada por meios transformativos. Entretanto, o material melódico e rítmico realizado por *engenhoca* tende a variar com frequência, sendo inclusive capaz de continuar seu improviso com variações de seu próprio material, como verificou-se em testes realizados com saída MIDI.

As escolhas realizadas pelo sistema possuem, em várias etapas de seu “pensamento”, algum grau de liberdade, realizado por geradores de números aleatórios, assim como exemplos vistos anteriormente.

Processos transformativos realizados em *engenhoca* também estão descritos nos modelos estudados, como reordenamento de alturas como forma de manter um conjunto de notas sem uma repetição sequencial delas e acumulação de processos transformativos para um resultado complexo derivado de um material anterior.

O sistema também não possui uma metodologia de análise do material musical escutado. Entretanto, é evidente que determinadas escolhas de algoritmos de composição e da organização da segmentação do material possuem forte influência de um pensamento de linha *eurológica* como coloca George Lewis. Ainda assim, o sistema não é compatível com um ambiente de concerto centrado em composição, mas sim em processos de interação entre os agentes, uma vez que as ações da máquina não podem ser diretamente controladas durante uma performance.



## 5 Desenvolvimento do protótipo

Batizada de *engenhoca*, a máquina foi pensada como um protótipo de sistema capaz de criar música, de forma autônoma, em um cenário de improvisação livre não-idiomática.

Apesar da justificativa de sua implementação estar diretamente relacionada às possibilidades de aperfeiçoamento pessoal na prática da improvisação livre, nada no sistema foi pensado a partir de uma utilidade “estritamente” pedagógica, principalmente porque, dado o vasto cenário que a improvisação livre representa, seria difícil assumir algum ponto de vista definitivo, por exemplo, sobre a análise do que é tocado pelo instrumentista, ou sobre que tipo de sugestões musicais poderiam ser apresentadas aos usuários do sistema. Dessa forma, as relações construídas com a máquina estão em um campo intuitivo, enquanto proveitos tirados da máquina por musicistas em um campo subjetivo.

Popularmente definida como uma máquina precária, uma *engenhoca* costuma ser construída de forma improvisada. A escolha do nome busca absorver a dualidade no uso do sufixo “-oca”:

Formações do corpúsculo: — motoca (derivado de moto, forma abreviada de motocicleta). O sufixo -oca é usado na formação de palavras diminutivas. Ele não significa, porém, simplesmente ‘pequeno’. Ele contém uma coloração emocional. Em motoca indica apreço, enquanto em belezoca e engenhoca a conotação é mais de despreço. (SANDMANN, 1988, p. 47)

Ao mesmo tempo, seu nome questiona as noções pejorativas tradicionalmente atribuídas ao ato de improvisar como “organizar algo às pressas, sem preparo prévio”, “mentir” ou “desempenhar uma função (...) sem que esteja qualificado para tal” (IMPROVISADO... , s.d.).

### 5.1 Pure Data e bibliotecas

Como apresentado anteriormente, o protótipo de *engenhoca* foi inteiramente programado na linguagem de programação visual Pure Data (PD)<sup>1</sup>. A escolha se fez por alguns fatores que se complementaram: familiaridade do autor com o ambiente, o que permitiria que a programação fosse realizada no curto espaço de tempo disponível para este trabalho; o fato de PD ser *software* livre, garantindo acessibilidade a qualquer um interessado; inteligibilidade do modelo de programação visual oferecido no PD a músicos não experientes em programação e/ou professores interessados na exploração do ambiente com alunos; integração direta entre o código e sua interface visual.

<sup>1</sup> <https://github.com/pure-data/pure-data>

Pure Data se trata de um ambiente de programação musical compilado na linguagem de programação *C*, utilizando internamente o paradigma de programação *fluxo de dados*. É compatível com os sistemas operacionais GNU/Linux, Mac OS X, iOS, Android e Windows. É similar à linguagem Max, base de várias máquinas discutidas no capítulo anterior, também originalmente desenvolvida por Miller Puckette. Códigos criados com a linguagem, por sua característica visual, são comumente denominados de *patches*.

Para otimizar tanto a produção do modelo, quanto a execução de seus algoritmos, foram utilizadas *bibliotecas* desenvolvidas para o ambiente, capazes de estender funcionalidades da linguagem, bem como trazer novas funcionalidades que seriam dificilmente implementadas de forma autônoma no tempo de elaboração deste projeto. O uso de bibliotecas com algoritmos já otimizados em linguagens de programação é uma prática comum e por vezes recomendada. Uma vez que as bibliotecas utilizadas no *patch* também são de código aberto, seus objetos poderão ser compilados a partir da linguagem de programação *C*, caso necessário, se fazendo compatível com futuras versões do Pure Data.

As bibliotecas utilizadas estão disponíveis via *Deken*, um gerenciador de bibliotecas dentro do ambiente Pure Data, acessível pelo menu “*Help > Find externals*”. As funcionalidades básicas de cada biblioteca necessária serão apresentadas a seguir:

- *Freeverb*: distribuída sob a licença GNU GPL 2.0; implementa uma versão em estereo de um algoritmo de reverberação baseado no modelo Schroeder/Moorer (SCHROEDER, 1962); de autoria de Olaf Matthes.
- *timbreID*: distribuída sob a licença GNU GPL 3.0; implementa descritores para análise e classificação de timbres; desenvolvida por William Brent. (BRENT, 2010)
- *mjlib*: distribuída sob a licença GNU GPL; implementa algumas funcionalidades em campos diferentes. Utilizada pelo objeto `[metroplus]`, uma variação do objeto `[metro]`, disponível em PD *vanilla*, consistindo em um tocador de ritmos a partir de uma lista de valores; elaborada por Mark Williamson.
- *list-abs*: distribuída sob a licença 2-Clause BSD; implementa objetos para manipulação de listas; de autoria de Frank Barknecht.

Além disso, o algoritmo `[sum]`, que realiza a soma dos valores de uma lista, implementado originalmente na biblioteca *zexy* (distribuída sob a licença GNU GPL 2.0), é utilizado no código, renomeado como `[list-sum]`, mas com seus créditos mantidos.

## 5.2 Organização do patch

Pela revisão bibliográfica apresentada anteriormente, percebe-se que a maioria dos sistemas desenvolvidos na área necessitam de uma organização interna subdividida

em componentes para a realização de diferentes tarefas. A arquitetura mais comumente encontrada é baseada na divisão entre um componente de escuta e um componente de toque. Em *engenhoca*, a arquitetura foi pensada em uma subdivisão tripartida, em que as escolhas de cada frase musical a ser tocada fossem realizadas de forma separada de sua execução. A vantagem do modelo está em permitir que o projeto seja estendido para a execução a partir de outros sintetizadores a serem implementados no próprio *patch*, ou utilizando-se da saída MIDI para conexão a outros ambientes.

Além dos componentes principais, o modelo conta com mais dois *subpatches*: uma interface gráfica e um pequeno componente para controle de saída e gravação do áudio. A organização interna de cada componente será vista em maior detalhamento a seguir. O *patch* encontra-se disponível em: <https://github.com/ofefo/engenhoca>.

## 5.3 Interface

A interface gráfica apresenta osciloscópios correspondentes à entrada do microfone e à saída da eletrônica; um visualizador da amplitude do RMS do sinal de entrada, usado para coordenar o valor de *offset* com o ruído de fundo; e botões de controle: um botão indica as instruções para utilização do *patch*, dois botões para início e fim de um improviso (que também ativam e finalizam gravações de áudio da entrada e da saída dos sintetizadores) e dois botões para controle de saída (áudio/MIDI). Por fim, a interface apresenta cinco botões *toggle*, indicando o estado de cada voz (ativa/inativa).

O cálculo do valor de *offset*, o momento identificado como fim de um som tocado e início de um “silêncio”, é realizado pela média dos picos de RMS atingidos quando se escutado o ruído de fundo da sala. O RMS é calculado em janelas de 64 samples, e seus valores escritos em uma tabela a cada 100ms. Ao se apertar o botão presente na interface uma vez, o algoritmo é disparado, acrescentando o pico encontrado na tabela em uma lista, da qual é calculada a média a cada 10ms, enviada para várias instâncias do sistema. Ao se desligar o *toggle*, o algoritmo é pausado e o último valor é impresso no console do PD, bem como a lista com os picos é zerada.

## 5.4 Escuta

O processo inicial para a construção da performance da máquina começa pelos algoritmos de “escuta” do que é tocado. Esses algoritmos estão subdivididos em 4 *subpatches* principais ([*attack*], [*timbre*], [*dynamics*], [*pitch*]), que calculam informações musicais diretamente a partir do áudio e dois secundários ([*rhythm*] e [*legato*]), recebendo somente informações indiretas do áudio, calculadas pelos *subpatches* principais. As saídas de cada *subpatch* são utilizadas pelo resto do programa como variáveis globais.

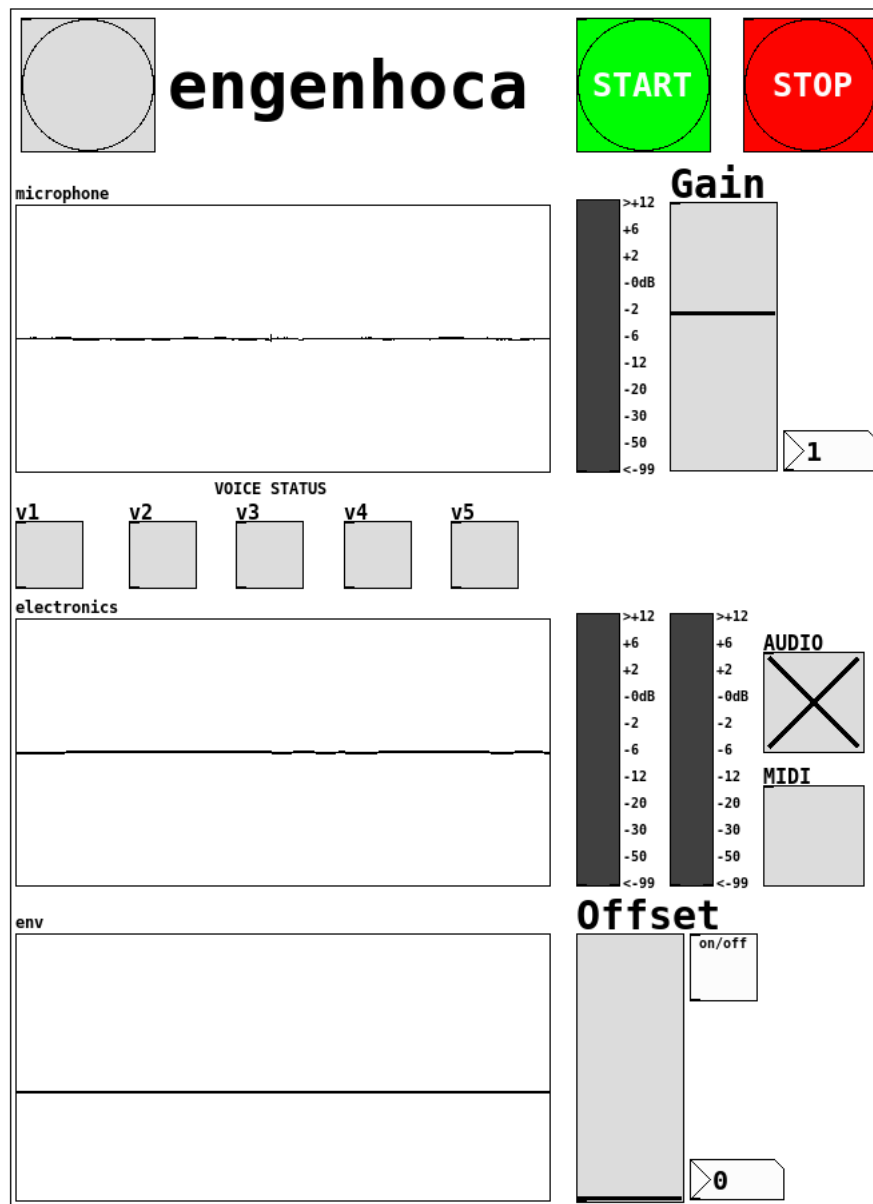


Figura 1 – Interface gráfica desenvolvida para o protótipo

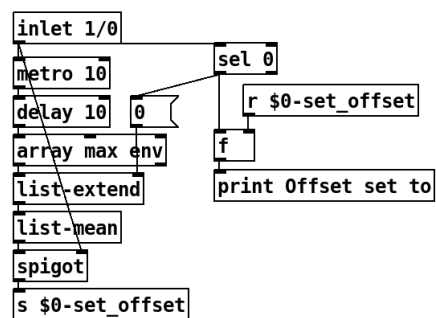
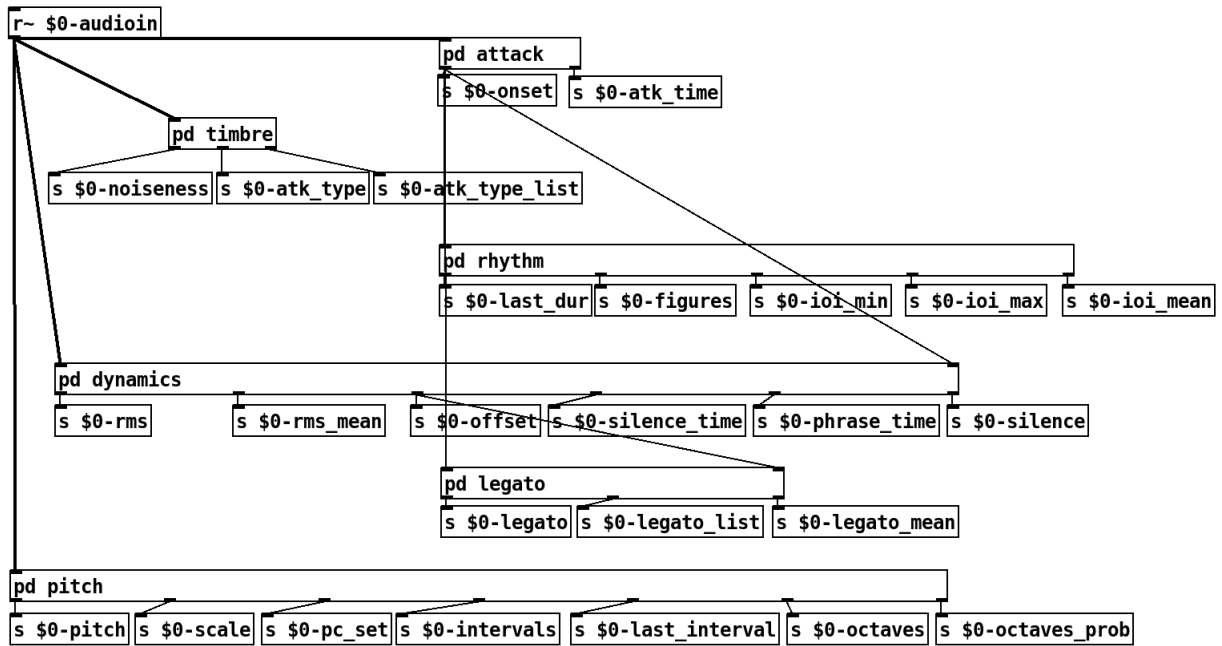
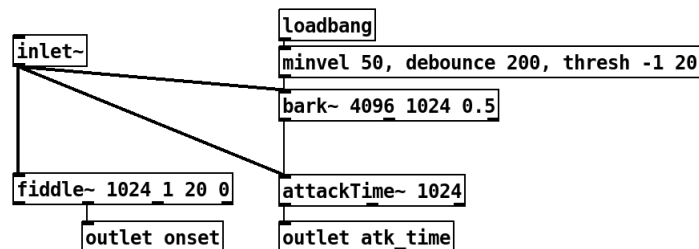


Figura 2 – Algoritmo para definição da intensidade do ruído de fundo

Figura 3 – *Subpatch* contendo os módulos para a escuta

### 5.4.1 Ataque

O *subpatch* responsável pela identificação de ataques possui dois algoritmos que realizam funções diferentes: O primeiro, [fiddle~] (PUCKETTE; APEL; ZICARELLI, 1998), trata-se de um objeto próprio do Pure Data, e funciona como um “canivete suíço” para as extrações mais básicas do conteúdo musical de um áudio, como alturas, momento do ataque, (neste trabalho interpretado como *onset*, ou o “momento correspondente ao início de um evento sonoro musical” (LERCH, 2012, p. 119)), amplitudes e harmônicos, sendo aqui utilizado para identificação do momento do onset. Um segundo algoritmo coordena dois objetos da biblioteca *timbreID* ([bark~] e [attackTime~]) para identificação do final do ataque, e por seguinte sua duração.

Figura 4 – *Subpatch* contendo a identificação do momento de *onset* e duração do ataque.

O momento dos *onsets* identificados são repassados a três outros subpatches: [rhythm], [dynamics] e [legato].

### 5.4.2 Altura

A identificação das alturas é realizada pelo [sigmund~], objeto que “analisa a entrada de um som em componentes sinusoidais, que podem ser relatados individualmente ou combinados para formar a estimativa de uma altura”, de acordo com sua própria documentação. O único parâmetro definido em sua saída são alturas estimadas em valores MIDI.

Aqui foi escolhido um importante fator de segmentação dos eventos sonoros, definindo-se as tabelas para uma escuta de curto prazo a partir dos últimos doze sons. Tal definição foi realizada por constituir a única maneira para identificação de todos os conjuntos de alturas até doze sons (garantindo que se escute o total cromático). Essa triagem repercutirá em todas as outras escutas. Entretanto, como se verá mais à frente, garantiu-se que a segmentação de frases tocadas pela máquina não se restringisse a conjuntos de doze sons. Os valores MIDI correspondentes às notas tocadas são escritos em na tabela *midi\_table*.

Valores de classes de altura (FORTE, 1973), tendo a nota Dó como referência (correspondente ao número 0), são circularmente escritos na tabela *pc\_table*. Para redução dos últimos doze valores encontrados a um conjunto sem repetições, é utilizado o objeto [list-unique-text], uma implementação mais eficiente de um objeto similar presente na biblioteca *list-abs*. O *patch* foi obtido via um fórum de Pure Data na *Web*, e seu autor assentiu em sua utilização sem créditos, com a condição de que o projeto fosse distribuído como *software* livre.

Uma tabela com os últimos intervalos melódicos entre alturas é realizada pela diferença entre os últimos dois valores escritos na tabela *midi\_table*.

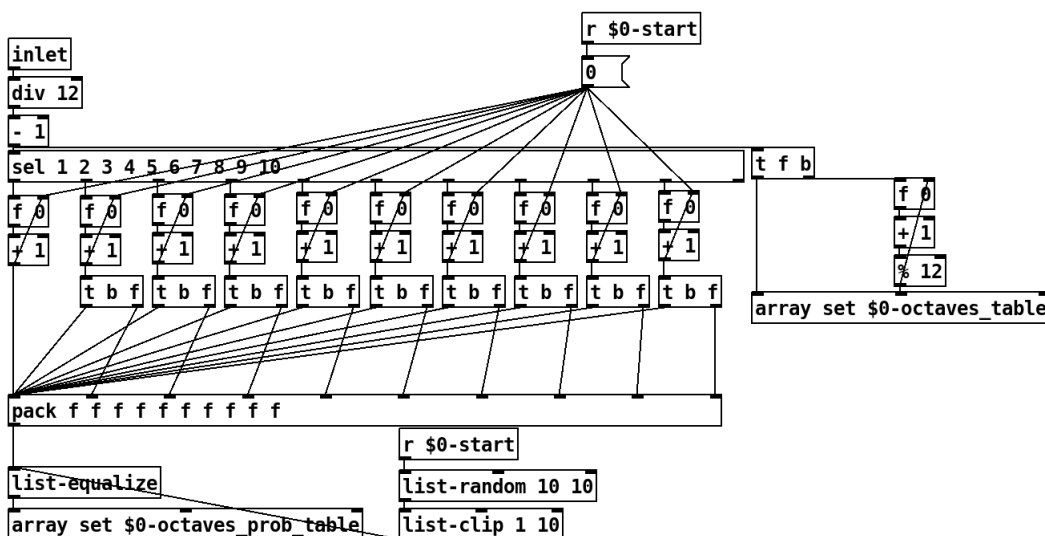


Figura 5 – Subpatch para identificação de oitavas.



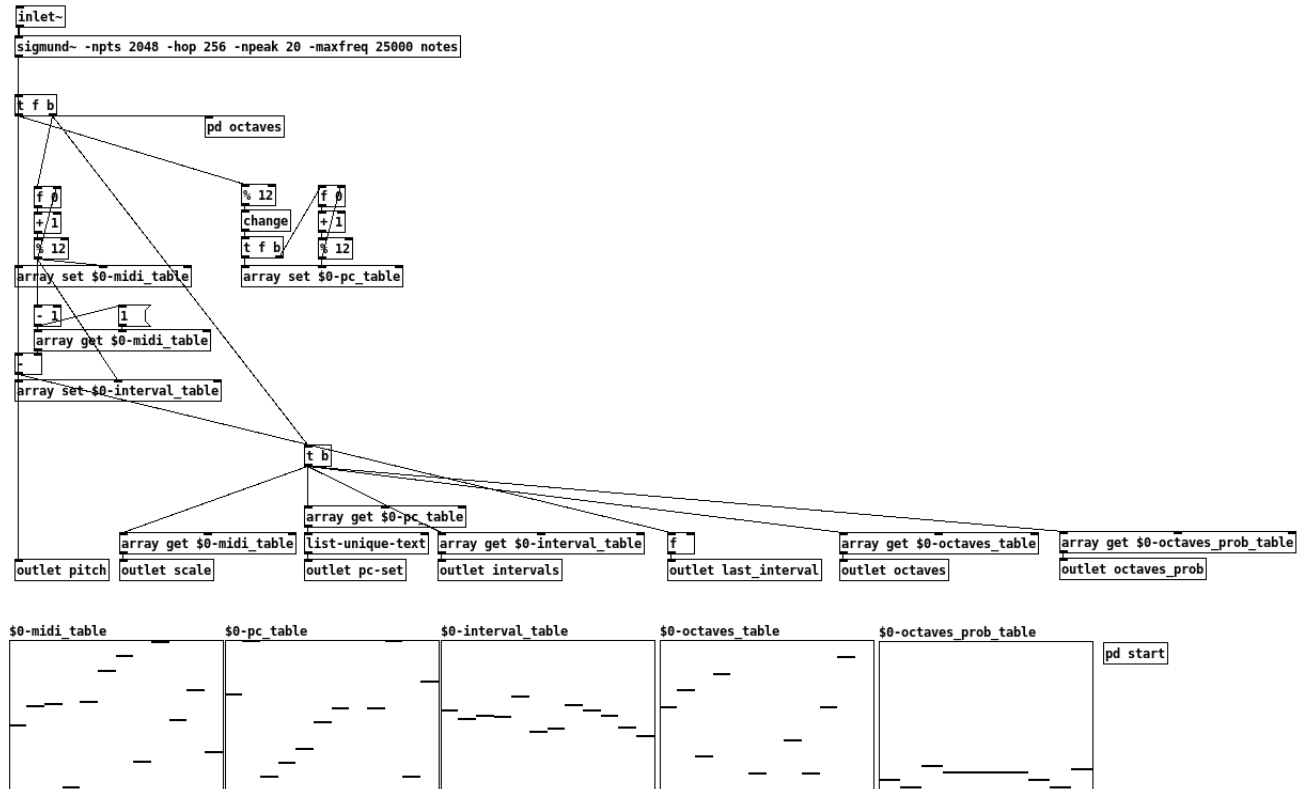


Figura 6 – Módulo com identificação de alturas, intervalos, e conjunto de classes de altura.

O cálculo de oitavas é realizado pela divisão inteira do último valor MIDI encontrado por 12. Do resultado, subtrai-se 1 para ajustar o valor à numeração usual de um piano (em que o Dó central corresponde à oitava 4). As últimas doze oitavas são salvas em uma tabela, enquanto a soma de valores encontrados para cada oitava são continuamente equalizados pelo objeto `[list-equalize]` em valores entre 0 e 1, representando, dessa forma, a probabilidade de ocorrência de cada oitava desde o início de um improviso com o sistema.

Assim como nos *subpatches* apresentados a seguir, todas as tabelas recebem valores aleatórios dado o início de uma performance.

### 5.4.3 Ritmo

A segmentação rítmica é realizada a partir da identificação do intervalo entre *onsets*. Tal processo ocorre no objeto `[ioi]` pela contagem em milissegundos entre os dois últimos *onsets*, realizada pelo objeto `[timer]` (assim como descrito em (ROWE, 2001, p. 223)).

Em sequência, as últimas doze durações medidas são salvas em uma tabela circular, em que ao se atingir o último índice, retorna-se ao primeiro, mantendo-se os demais, que são substituídos a cada *onset*.

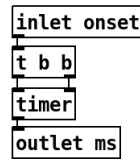


Figura 7 – Abstração para contagem do intervalo entre *onsets*, em milissegundos.

Além disso, o *subpatch* consta com um gerador de uma lista aleatória de doze valores, que ocupam a tabela ao início de uma performance, de forma que, independente da ausência de toque por parte do musicista, a máquina já possui valores rítmicos com os quais pode trabalhar.

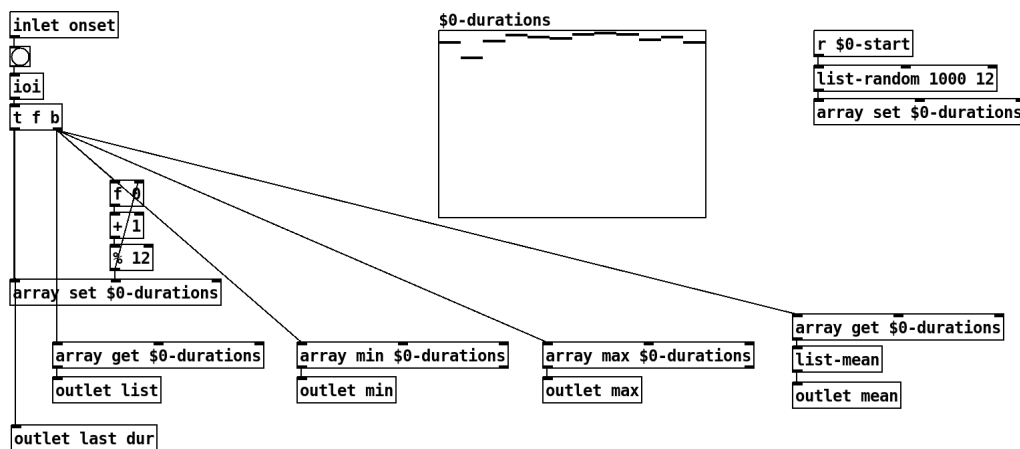


Figura 8 – *Subpatch* com identificação das últimas doze durações rítmicas.

As saídas do *subpatch* correspondem a: lista dos últimos doze valores rítmicos, duração mínima da lista, duração máxima da lista, e média dos valores rítmicos.

#### 5.4.4 Timbre

Apesar de toda riqueza de informações possível de se extrair com descritores sonoros de atributos do timbre, *engenhoca* utiliza somente um medidor de *nivelamento espectral*, também implementado pela biblioteca *timbreID* no objeto [specFlatness~], para classificar eventos sonoros por características tímbricas.

O algoritmo de nivelamento espectral é utilizado pelo programa para uma medição do “nível de ruído” de um evento sonoro. A descrição do algoritmo de acordo com a própria documentação da biblioteca *timbreID* é apresentada a seguir:

Nivelamento Espectral é a razão da média geométrica do espectro de magnitude pela média aritmética do espectro de magnitude. Um espectro muito nivelado com igual energia em todas as bandas (por exemplo, o do ruído branco) deve ter um valor de nivelamento espectral alto, onde 1 é o nivelamento perfeito.

O *subpatch* calcula, logo após a identificação de um *onset*, até 10 valores de nivelamento espectral pelo próximo décimo de segundo (106.7ms), salvando-os na tabela *noiseness\_table*. Decorrido este processo, calcula a média dos valores encontrados na tabela, de forma a medir uma média do nível de ruído encontrado no ataque. Este valor é retornado como medida de *noiseness* (aqui, outra nomenclatura para nivelamento espectral), uma variável.

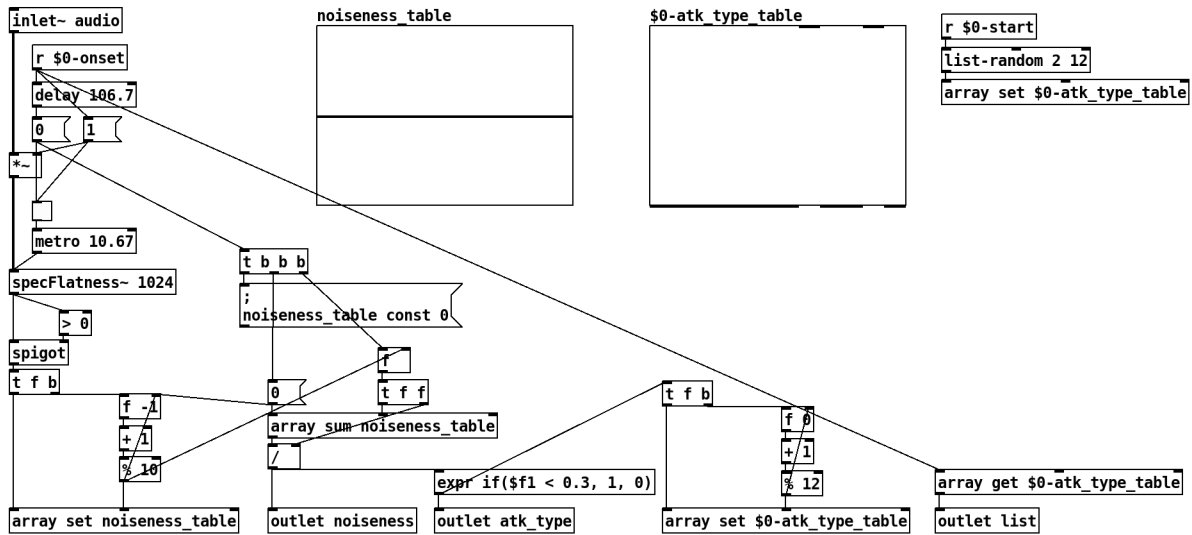


Figura 9 – *Subpatch* com identificação do timbre pelo nivelamento espectral.

Caso o *noiseness* de um ataque seja medido acima de 0.3 (um valor arbitrário encontrado em testagens do sistema), ele é considerado um ataque percussivo, caso contrário, é considerado um ataque “melódico”. É importante ressaltar que nada impede que haja o reconhecimento de melodias realizadas com instrumentos percussivos afinados como metalofones, pianos ou marimbas. O algoritmo tenta na verdade realizar uma interpretação da escuta ou não-escuta de alturas determinadas por um humano. A lista das últimas doze classificações é atribuída à variável *atk\_type\_list*.

Valores aleatórios também são providos à tabela dado o início de uma performance.

### 5.4.5 Dinâmica

O *subpatch* responsável pela segmentação de valores de dinâmica trabalha com uma tabela onde são salvos os últimos doze valores de RMS medidos logo após o ataque em uma pequena janela de 1024 samples pelo objeto `[env]`. A saída *rms\_mean* entrega a média dos últimos valores, representando uma memória de curto prazo da dinâmica tocada (sem entretanto perceber se houveram grandes variações).

A abstração `[offset]` realiza o cálculo do momento em que um evento sonoro termina, a partir da simples identificação de quando um som se tornou menos intenso do

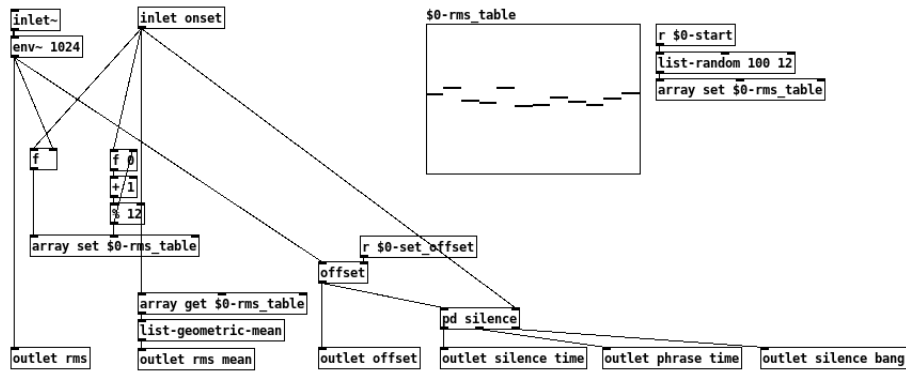


Figura 10 – *Subpatch* para reconhecimento de elementos referentes à dinâmica.

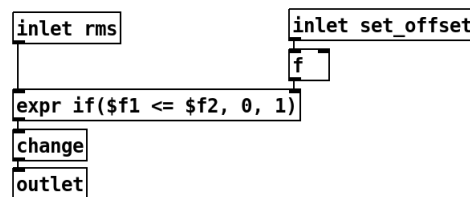


Figura 11 – Abstração para identificação do fim de um evento sonoro.

que a média do pico de RMS calculado para o ruído de fundo pelo sistema implementado na interface gráfica.

Um *subpatch* realiza a segmentação de períodos de silêncio, bem como contabiliza o tamanho das frases musicais todadas. A duração de 300 milissegundos (contado pelo primeiro *timer*) entre o início de um silêncio e o próximo *onset* foi escolhido arbitrariamente para identificação do início de um momento de silêncio. Quando isso ocorre, o último valor é entregue na primeira saída, e um novo *timer* é ligado para a contagem do decorrer do tempo entre o início de uma frase até o próximo silêncio.

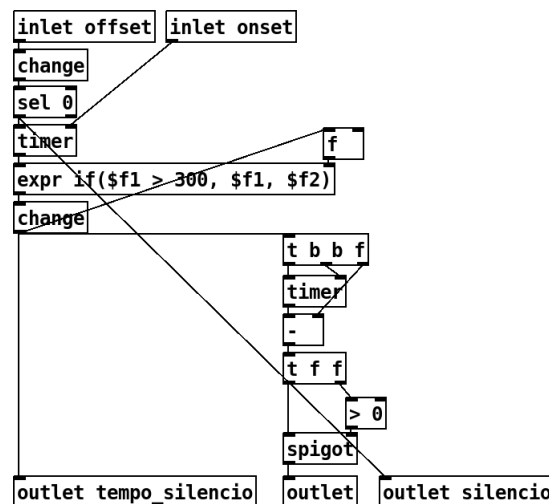


Figura 12 – Algoritmo para segmentação de durações dos silêncios e das frases.

### 5.4.6 Legato

A identificação do legato é realizada novamente por uma abstração, medindo-se a proporção do último trecho de silêncio com a duração do último IOI (em milissegundos), utilizando a medida de *offset* encontrada no *subpatch* anterior.

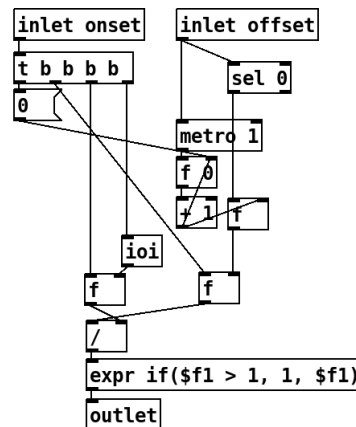


Figura 13 – Medidor de legato, com saída de valores entre 0 (máximo staccato) e 1 (máximo legato).

Os valores medidos são salvos em uma tabela, da qual será por fim extraída a média, continuamente enviada a uma variável, assim como o último valor de legato, e toda a lista dos últimos doze eventos. Valores aleatórios também são escritos na tabela dado o início de uma performance.

## 5.5 Pensamento

A elaboração do “pensamento” do modelo, naturalmente o componente mais desafiador para implementação, se deu por um longo processo de testagem e experimentação com diferentes ideias de texturas musicais. Apesar da tentação pela escolha de programação de texturas musicais determinadas, nitidamente funcionais a partir exemplos da literatura de máquinas criativas ((RISSET; DUYNE, 1996), (ESSL, 1995)), o desenvolvimento preliminar do protótipo focou-se nas possibilidades auferidas por texturas indeterminadas a priori.

Uma vez que texturas musicais complexas emanam espontaneamente em um cenário de improvisação livre, imaginou-se que não haveria a necessidade de organização de um material bem definido, que poderia inclusive atrapalhar o fluxo de interação com um musicista no momento em que o sequenciamento fosse observado.

Por consequência, foram implementados somente dois modelos de organização de frases musicais a serem tocadas: um planejador de *homorritmo*, e um planejador de *contraponto*. Ainda que uma homorritmia possa ser considerada uma textura determinada,

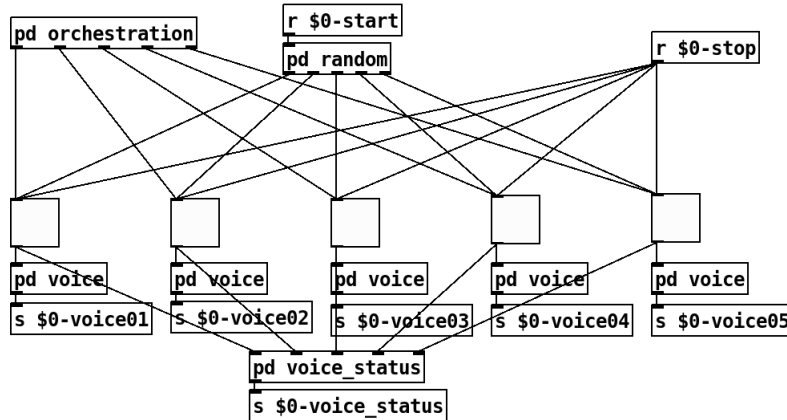


Figura 14 – Vista do *subpatch* contendo as operações de *raciocínio* da máquina.

sua escolha como elemento a ser tocado não é determinada ou controlada pelo improvisador, trazendo um elemento de espontaneidade à interação. O contraponto, por sua vez, também não obedece regras de consonância e dissonância, e trata-se, portanto, de uma resposta livre ao material tocado pelo músico, observadas algumas restrições.

Uma vez que a produção sonora pode ser realizada por até cinco vozes simultâneas, a seção do *pensamento* também conta com um *subpatch* para tomada de decisões de cunho orquestral. Internamente, cada voz se comporta dentro das mesmas possibilidades, podendo realizar simultaneamente ideias distintas dentro desse escopo.

### 5.5.1 Vozes

A implementação de cada voz é baseada na seleção dentre um dos métodos disponíveis como texturas musicais. Tal implementação permite a posterior inclusão de outros métodos para elaboração de uma textura a ser tocada. Internamente, cada método realiza operações sobre os valores de entrada, que são então concatenados em uma lista a ser enviada aos sintetizadores.

Um *subpatch* define o tempo para o sorteio dentre os dois métodos presentes a partir da média de durações das últimas dez frases.

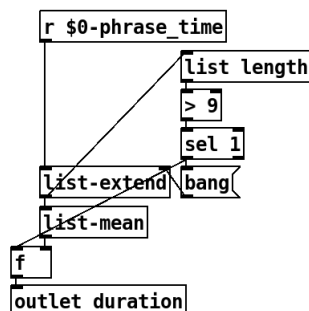


Figura 16 – Algoritmo para definição do tempo de uma textura musical.

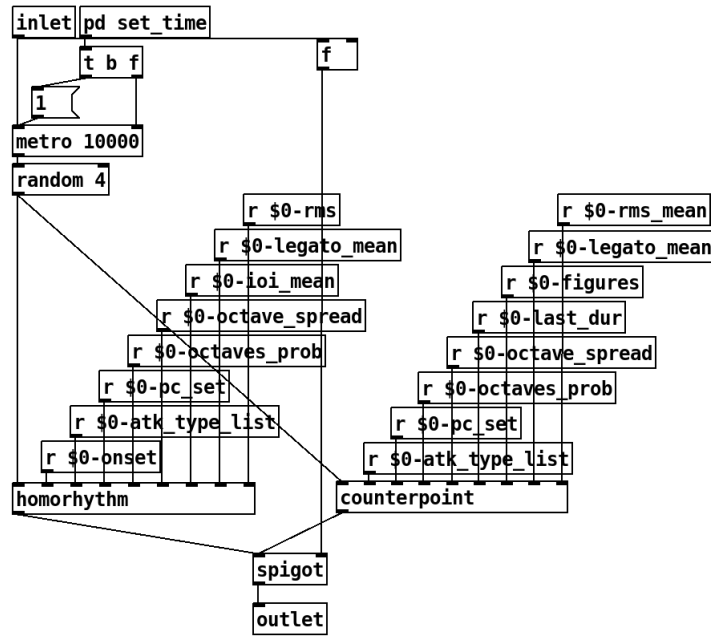
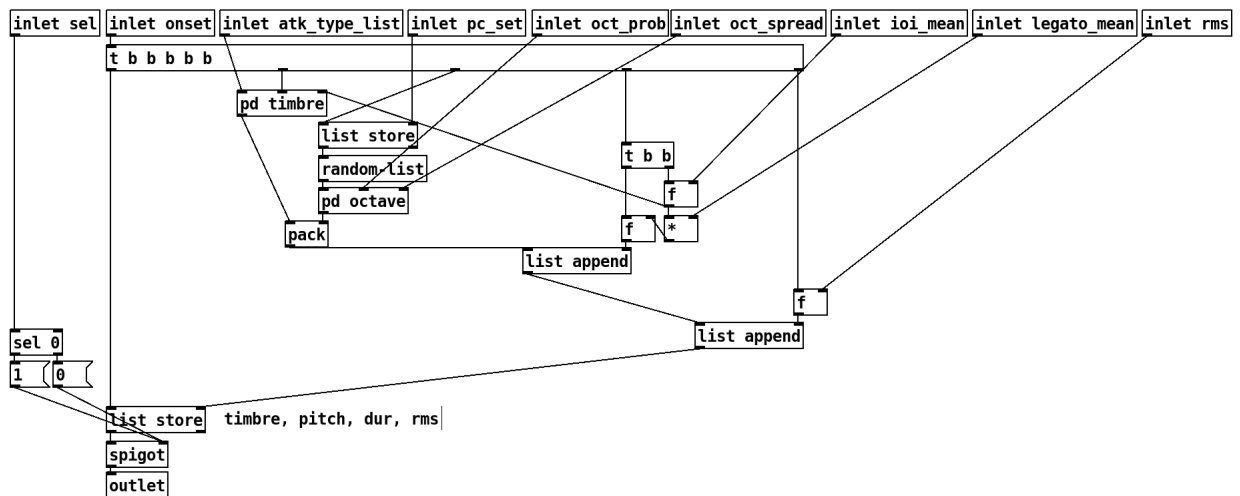


Figura 15 – Organização interna de uma voz.

### 5.5.2 Homorritmo

A organização de uma textura tocada em homorritmo com a entrada depende diretamente de sua execução junto do onset identificado. Dessa forma, uma lista é elaborada pela concatenação de vários parâmetros medidos no momento de cada ataque, para então ser enviada ao sintetizador. Os parâmetros escolhidos para execução de uma figura em homorritmo são: lista dos últimos doze timbres (percussivo/melódico); lista do conjunto de classes de altura; lista com as probabilidades de ocorrência de cada oitava; lista com a dispersão das oitavas dentre as últimas doze notas; média dos últimos doze IOIs; média dos últimos doze índices de legato; RMS do último ataque.

Figura 17 – Organização interna da abstração *homorhythm*.

Para decidir-se a duração de cada nota, a média dos últimos valores de IOI é multiplicada pela média dos últimos doze índices de legato. Desse modo, busca-se manter uma articulação próxima da que está sendo executada pelo músico.

É interessante notar que a escolha de alturas é realizada pela escolha de um valor correspondente a um índice aleatório do último conjunto de classes de alturas tocado. Assim sendo, não há a preservação de sequências criadas pelo músico na resposta da eletrônica. A escolha de oitavas está dissociada da escolha de alturas, e é realizada a partir de uma seleção probabilística, selecionando-se aleatoriamente uma das oitavas mais tocadas durante toda a performance. A quantidade de oitavas que entrarão no sorteio (dentre as dez possíveis), é definida pela dispersão (quantas oitavas diferentes foram tocadas) das últimas doze notas.

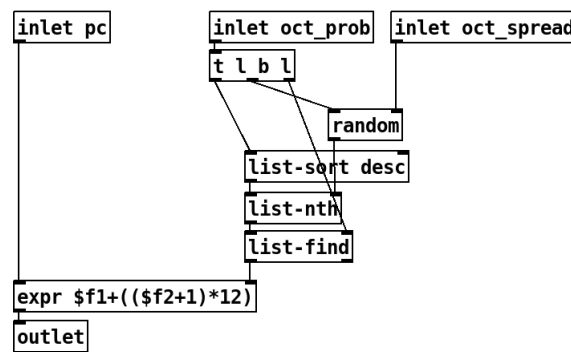


Figura 18 – Detalhe de implementação do *subpatch* octave.

A escolha de qual timbre será tocado também é realizada de acordo com a identificação dos últimos timbres tocados pelo músico. Diferentemente de parâmetros vistos anteriormente, a escolha do timbre varia de acordo com o andamento que está sendo tocado no momento. Caso a média dos IOIs seja menor (indicando uma textura mais rápida), o timbre selecionado será mantido por mais tempo, tendo como limite 30 eventos sonoros. Essa escolha permite que se mantenha uma coerência tímbrica maior, impedindo que o sistema troque constantemente de material tímbrico. O detalhe de como funciona cada timbre será demonstrado na descrição dos sintetizadores. Por hora, é importante entender somente que existem no sistema três opções de timbres “percussivos” e quatro opções de timbres “não-percussivos”.

A lista dada como saída da abstração contém um identificador de timbre, o valor MIDI para a altura escolhida, a duração da nota e sua intensidade em RMS.



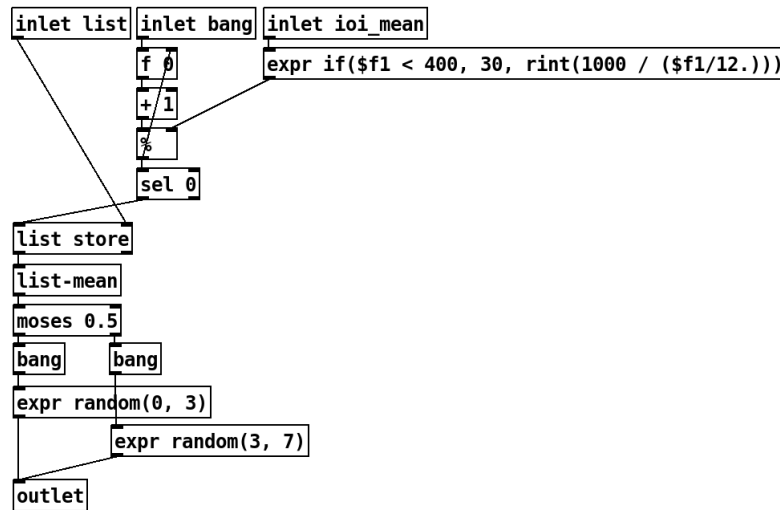


Figura 19 – Detalhe de implementação do *subpatch* timbre.

### 5.5.3 Contraponto

Muito similar à abstração contendo a geração de uma nota em homorritmo, a concepção do material contrapontístico possui apenas um detalhamento maior em relação à escolha de alturas, bem como implementa internamente sua reprodução rítmica. Em comparação, também trabalha com a média dos últimos doze RMS, estabelecendo uma pequena memória em relação à dinâmica local de uma performance. Sua saída também se trata de uma lista contendo os mesmos parâmetros da abstração responsável pelo homorritmo.

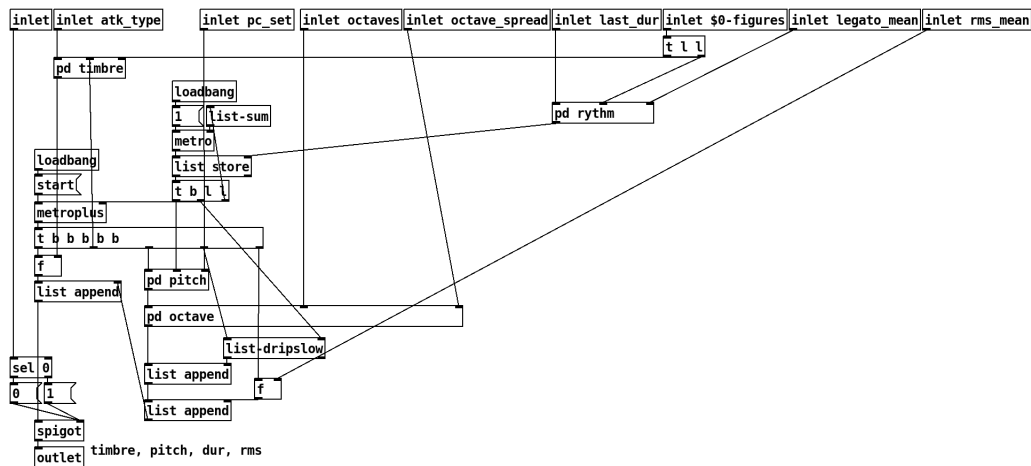


Figura 20 – Abstração contendo o gerador de material contrapontístico.

A realização rítmica é elaborada a partir de dez operações diferentes sobre o ritmo recebido como entrada. Uma importante escolha para o sistema foi a desvinculação do ritmo da dependência dos últimos doze eventos. Isso foi realizado a partir de um algoritmo similar ao utilizado para a escolha do timbre. No caso, uma lista é continuamente estendida

com o último valor de IOI identificado pelo sistema, e é resumida a no máximo 30 valores, de acordo com a média tocada nos últimos doze ritmos. Logo, o tamanho dos eventos rítmicos gerados serão diretamente relacionados ao andamento local da performance de um musicista. A lista gerada é diretamente escalada, em que cada valor é multiplicado pela média dos índices de legato das últimas doze notas.

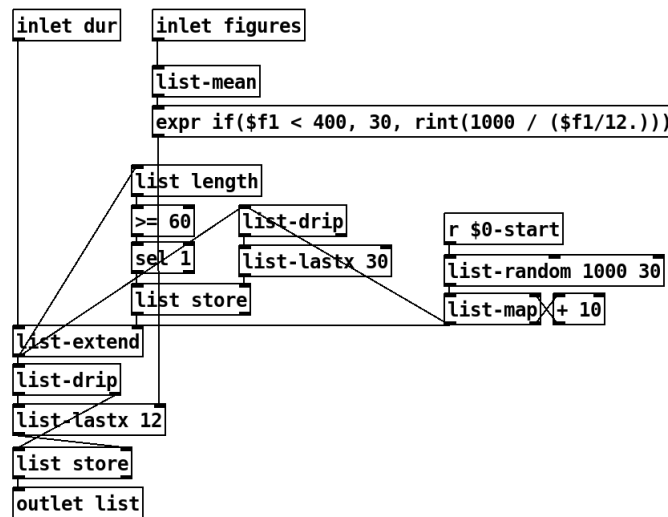


Figura 21 – Detalhe do gerador de listas de ritmos.

Em seguida, a lista de valores rítmicos é passada a dez *subpatches* simultaneamente. Enquanto um randomizador seleciona qual desses *subpatches* será escolhido para executar sua operação sobre a lista recebida. Uma vez que seria muito prolongado explicitar como funcionam os algoritmos de cada operação, seu funcionamento será brevemente abordado a seguir:

- [same]: reproduz a lista de durações exatamente na mesma ordem.
- [acc]: organiza a lista em ordem decrescente, gerando um *accelerando*.
- [rall]: organiza a lista em ordem ascendente, gerando um *rallentando*.
- [shuf]: ao contrário dos demais, só é capaz de trabalhar com os últimos doze valores. Reorganiza o ritmo internamente, aleatoriamente trocando as durações de posição.
- [mult]: realiza uma multiplicação de valores a partir de uma escala de durações como descrita em (STOCKHAUSEN, 1957), tendo como valores máximo e mínimo a metade e o dobro da duração de cada IOI.
- [add]: insere o menor IOI da lista em um ponto aleatório dela mesma, em alusão ao processo de valor acrescentado descrito em (MESSIAEN, 1944, p. 16).
- [ost]: realiza um ostinato com a menor, a maior ou média das durações.

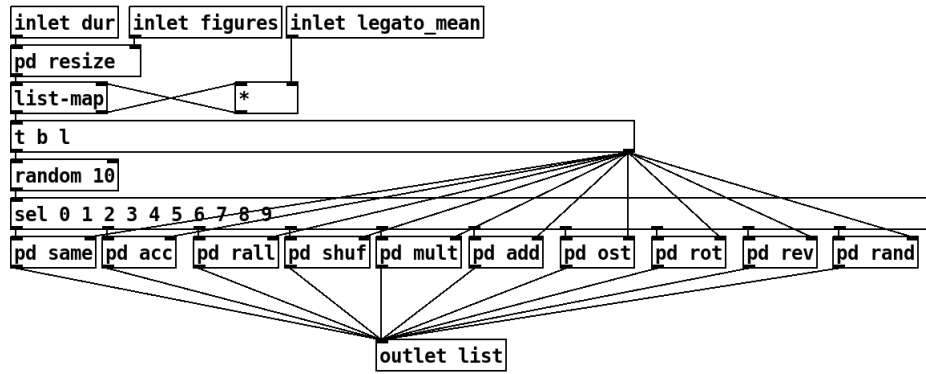


Figura 22 – *Subpatch* rhythm, contendo as operações realizadas sobre a lista gerada.

- [rot]: rotaciona os valores da lista em índices escolhidos ao acaso (dentro de seu tamanho).
- [rev]: inverte a ordem original da lista de durações.
- [rand]: cria uma nova lista, de mesma dimensão, utilizando valores aleatórios dentro dos limites encontrados na lista original.

De forma similar, sorteios são realizados para definir operações sobre o conjunto de classes de alturas, ou seu conjunto complementar (FORTE, 1973, p. 209) e serão abordados a seguir. O conjunto e operação escolhida só são alterados quando uma nova lista de durações é gerada, de forma que o material melódico mantém-se estável durante toda uma frase.

- [same]: escolhe uma altura aleatória do conjunto.
- [asc]: reproduz o conjunto em ordem ascendente de classe de alturas.
- [desc]: reproduz o conjunto em ordem descendente de classe de alturas.
- [tran]: realiza uma transposição aleatória das alturas do conjunto.
- [mult]: realiza uma multiplicação aleatória das alturas do conjunto.
- [comp]: escolhe uma altura aleatória do conjunto complementar.
- [casc]: reproduz o conjunto complementar em ordem ascendente de classe de alturas.
- [cdes]: reproduz o conjunto complementar em ordem descendente de classe de alturas.
- [ctra]: realiza uma transposição aleatória das alturas do conjunto complementar.
- [cmul]: realiza uma multiplicação aleatória das alturas do conjunto complementar.

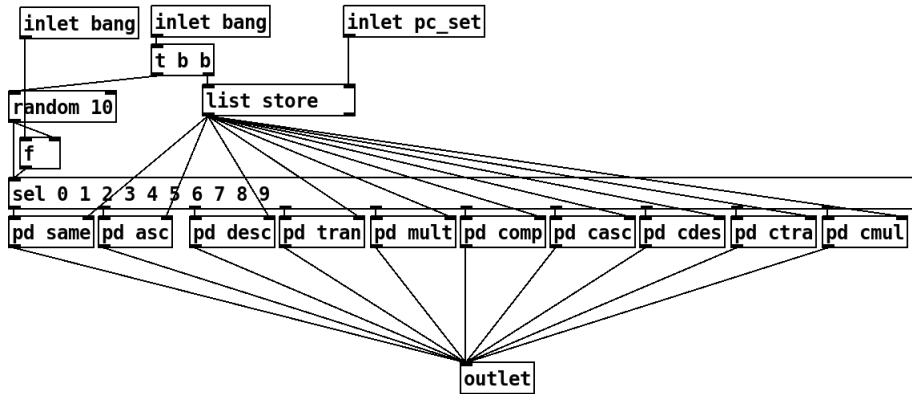


Figura 23 – Conteúdo do *subpatch* pitch, com as operações realizadas sobre o conjunto de alturas.

A reprodução da lista de durações é realizada pelo objeto [metroplus], que retorna pulsos em intervalos correspondentes às durações recebidas. A cada pulso é concatenada uma lista contendo, da mesma forma que no [homorhythm], o timbre escolhido, a altura, a duração e o RMS da nota.

#### 5.5.4 Orquestração

O *subpatch* responsável pela orquestração das vozes funciona a partir de três módulos: [activity], [anti\_silence] e [anti\_tutti].

[anti\_silence] e [anti\_tutti] observam o estado de ativação das vozes (pela variável *voice\_status*) para realizar operações simples: identificar quanto tempo decorreu desde que se atingiu silêncio total (nenhuma voz aberta) ou atividade total (todas as vozes abertas), respectivamente, duração que é comparada com o a duração máxima de silêncio ou tamanho de frase ocorrido nas últimas doze ocasiões de cada um. Caso ultrapassado este limite, uma nova voz é aberta ou fechada. Tal sistema impede que se passe muito tempo tanto em atividade máxima quanto em nenhuma atividade na eletrônica, reduzindo efetivamente o escopo orquestral de uma a quatro vozes simultâneas.

O restante das decisões para abertura ou fechamento das vozes ocorre a partir do *subpatch* [activity], que compara a média dos últimos valores correspondentes a intervalos (em valores absolutos), durações e intensidade, com uma média de mais ou menos valores, também em acordo com a velocidade do material tocado nos últimos doze ritmos. Caso a atividade de pelo menos dois fatores momentâneos for maior do que a da média variável de acordo com o andamento, mais vozes serão abertas. No cenário oposto, em que ao menos dois fatores momentâneos tenham uma média menor, vozes serão fechadas.

O processo de abertura ou fechamento de vozes pode ser realizado de duas formas distintas: uma escolha randômica de uma voz a ser aberta ou fechada; e a escolha contínua

de vozes a serem abertas ou fechadas em intervalos de tempo determinados pela média das durações das últimas dez frases.

## 5.6 Toque

A produção sonora em *engenhoca* pode ser realizada por sintetizadores implementados no próprio *patch*, bem como por uma saída MIDI, com a mesma quantidade de vozes, permitindo que usuários definam os próprios instrumentos virtuais para uma performance.

### 5.6.1 Síntese

Uma vez que cada evento sonoro carrega uma lista com valores correspondentes a seus parâmetros principais, foi possível implementar um sintetizador capaz de integrar tais parâmetros em sua produção sonora em tempo real. Optou-se pela implementação de síntese FM, descrita em (CHOWNING, 1973). A escolha foi realizada pela combinação de diversos fatores, dentre os quais a simplicidade de implementação, a versatilidade sonora provida pelo modelo, e o baixo custo computacional do algoritmo, que permite sua execução em tempo real com grande flexibilidade.

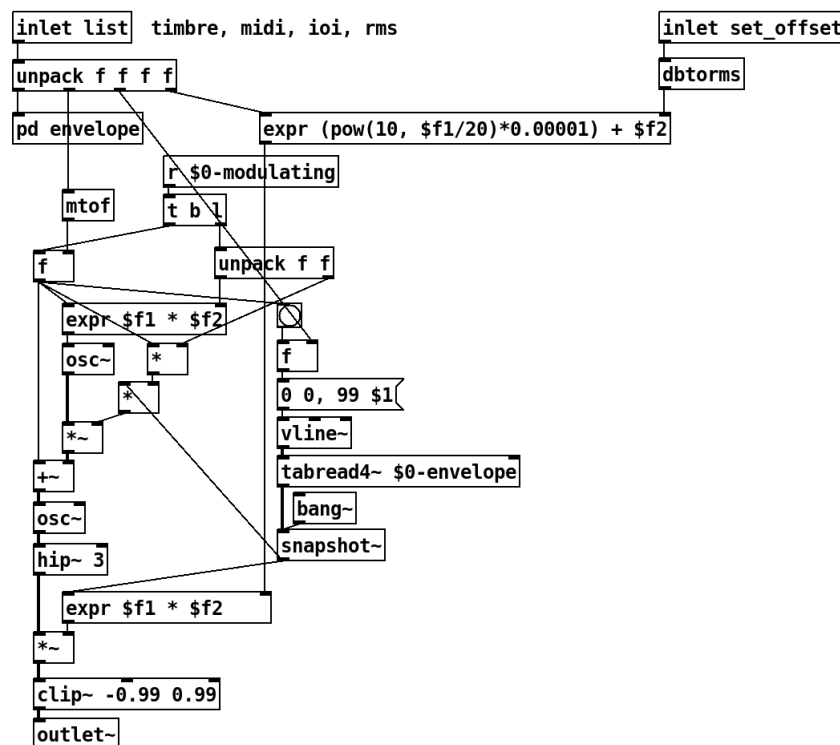


Figura 24 – Abstração responsável pela síntese sonora baseada no modelo de modulação de frequência.

A abstração recebe, para cada voz, a lista contendo as informações referentes ao timbre, altura em MIDI, duração do evento e intensidade em RMS. A lista é distribuída

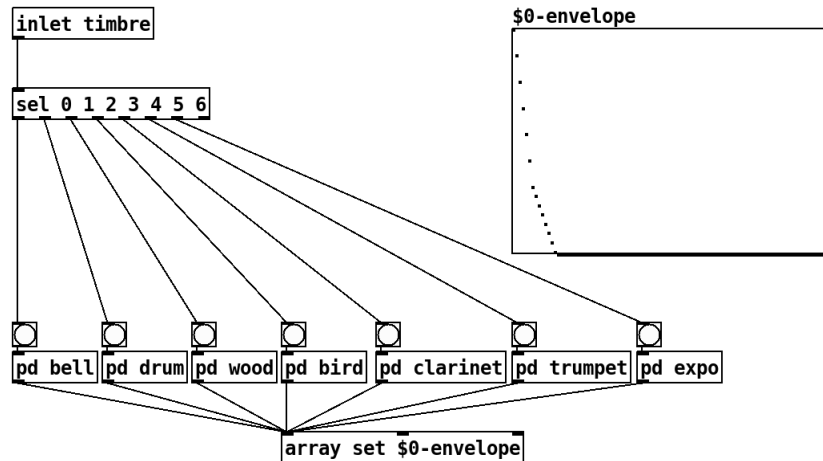


Figura 25 – *Subpatch* responsável pela definições do envelope e outros elementos responsáveis pelo timbre de cada evento sonoro.

pelo algoritmo, que é ativado após a escolha do envelope de intensidade, parte fundamental para a percepção tímbrica do modelo.

Os timbres disponíveis apresentam pequenas variações dos exemplos presentes em (CHOWNING, 1973), permitindo uma variação sonora maior a cada evento. Há também o acréscimo de sons que remetem a cantos de pássaro, bem como a inversão do envelope presente em sons de sinos, gerando sons de caráter artificial. As modificações realizadas nos exemplos originais permitem a construção de sons ligeiramente mais sintéticos, uma escolha baseada na preferência do autor, bem como no desejo de musicistas que interagiram com o sistema.

## 5.6.2 Espacialização

É evidente que uma parte importante da percepção musical está relacionada à disposição dos instrumentos no espaço, como argumenta Diana Deutsch:

(...) a disposição espacial de instrumentos deve de fato promover efeitos profundos em como a música é percebida. Quando dois grupos de instrumentos são espacialmente separados, e além disso uma clara separação temporal existe entre os sons produzidos por esses instrumentos, o resultado da dissociação perceptual pode ser tão pronunciada a ponto de prevenir que o ouvinte integre diferentes sons em um só fluxo. (DEUTSCH, 1983)

Portanto, todos os sons (de saída *mono*) executados pelos sintetizadores vistos na seção anterior passam por um algoritmo que realiza sua espacialização em um espaço *estéreo*. A localização de cada som é escolhida ao acaso, e modificada em intervalos de cinco segundos.

### 5.6.3 Reverberação

Além da espacialização, os eventos sonoros, agora em estéreo, passam por uma abstração que promove sua reverberação por meio do objeto [freeverb~]. De forma a evitar uma relação direta com o som tocado, cada parâmetro do processo de reverberação é definido ao acaso, em intervalos de tempo que também variam continuamente.

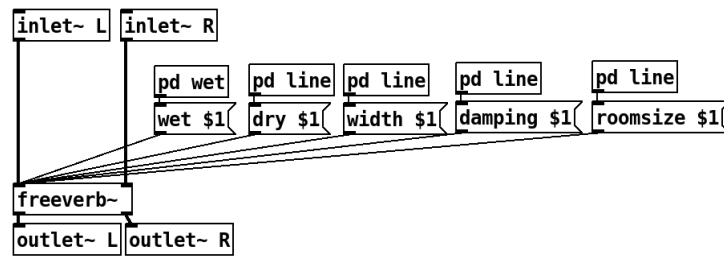


Figura 26 – Abstração que realiza a escolha dos parâmetros para incrementação de reverberação ao som de saída.

### 5.6.4 MIDI

O *patch* também oferece, ainda que de forma preliminar, a possibilidade de saída MIDI, também separada para cada voz. O processo de construção de uma mensagem MIDI também é realizada a partir da mesma lista de parâmetros de cada evento sonoro, que são então traduzidos para valores correspondentes ao protocolo MIDI. Uma vez que testes preliminares foram realizados utilizando um piano digital como referência, o alcance de valores de altura foi reduzida para o alcance do piano, realocando alturas maiores ou menores do que esse alcance para a última e primeira oitavas, respectivamente.

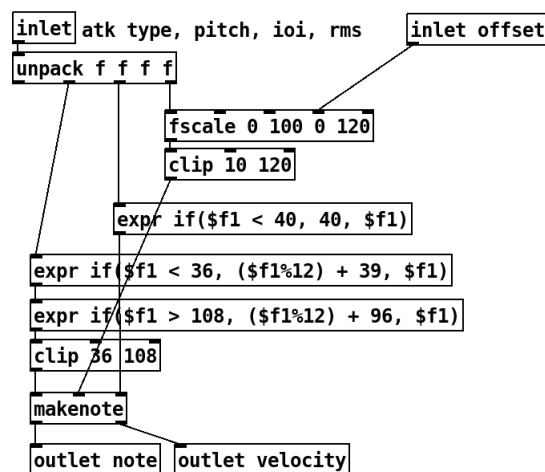


Figura 27 – Abstração para organização das mensagens MIDI.





## 6 Primeiros testes e impressões

Uma vez que o processo de revisão bibliográfica e implementação do protótipo ocorreu durante somente um período acadêmico, realizado totalmente à distância por medidas de segurança sanitária praticadas em nível Universitário devido à pandemia do novo coronavírus, não foi possível realizar uma testagem que permitisse um retorno da experiência dos musicistas participantes para melhorias na implementação do modelo.

Sendo assim, os relatos transcritos a seguir servirão como apontamentos dos musicistas sobre sua experiência inicial com *engenhoca* e suas opiniões sobre a prática individual da improvisação livre com máquinas.

### 6.1 Testagem inicial

Para uma testagem preliminar do protótipo de *engenhoca*, foi realizado um convite a três musicistas com experiência na prática da improvisação livre para que tocassem com o sistema, demonstrando a interação na prática por meio de gravações (automatizadas pelo programa).

Os participantes relacionam-se com a improvisação livre, dentre outras vivências, por integrarem o *ensemble* de improvisação livre *GILU*, surgido a partir de uma disciplina de graduação ofertada na Escola de Música da UFMG pelo professor Fernando Rocha e outros pesquisadores de pós-graduação relacionados à área. Seus perfis serão brevemente apresentados a seguir:

- Marcos Vinício Alves é percussionista e graduando nos cursos de Bacharelado em Percussão e Licenciatura em Música pela UFMG.
- Pedro Henrique Gilberto Alves Souza é violonista e graduando nos cursos de Licenciatura em Música e Bacharelado em Violão Clássico pela UFMG.
- Vanessa Bubgniak é trombonista e graduanda no curso de Licenciatura em Música pela UFMG.

A testagem foi conduzida de maneira individual, à distância, com orientações somente a respeito da instalação do ambiente Pure Data, suas bibliotecas, e o funcionamento básico da máquina. Não foi dado aos participantes nenhuma diretriz musical a ser seguida, bem como não foi exigido o uso de nenhuma instrumentação específica. Cada participante

teve em torno de uma semana para a utilização do sistema e o retorno das gravações realizadas<sup>1</sup>.

## 6.2 Primeiras impressões

Para a coleta das impressões dos musicistas, foram elaboradas somente duas indagações, que serão apresentadas a seguir, seguidas das respostas de cada um como citações. Por fim, será apresentada uma impressão do autor a respeito das interações realizadas pelos musicistas convidados. A ordem das respostas foi alterada para evitar a identificação pessoal de cada participante.

### 1. Descreva brevemente sua experiência com a máquina.

“Tive um problema com a instalação das bibliotecas. Mas depois da ajuda consegui fazer funcionar facilmente. Achei a interface e a resposta sonora da máquina bem instintivas e acessíveis. Testei uma vez só o áudio para entender o funcionamento e da segunda vez já comecei a improvisar utilizando instrumentos. Em alguns momentos senti a experiência um pouco repetitiva, no que diz respeito aos timbres. Mas de modo geral minha experiência foi boa e bastante fluida.”

“Bom, minha experiência com a máquina partiu da vontade de suprir um pouco a ausência de se ter alguém para tocar junto, e apesar de não ser um humano, eu senti que estava tocando junto com alguma coisa. O fato da máquina ter qualquer contato em relação ao que eu toco, independente de não ser o mesmo material/ideia, é interessante, pois faz com que você improvise a partir disso, e senti que isso pode abrir vários caminhos. Em alguns momentos eu senti falta de haver algum tipo de silêncio ou uma resposta mais natural do material que a máquina toca. A escuta da máquina pelo fone de ouvido dificulta bastante a relação sonora com ela, pois sinto mais isolamento e o contato fica superficial. Apesar de ter realizado várias improvisações sem o fone de ouvido, eu não consegui realizar a gravação final saindo nas caixas acusticas e isso com certeza muda a sensação da improvisação para mim.”

“Gostei dos timbres que respondem aos sons emitidos, mas achei difícil explorar os silêncios pelas respostas que a máquina me devolvia. Inclusive uma das dificuldades, em todas as práticas, foi terminar. Me coloquei muito atento o tempo inteiro por conta dessa imprevisibilidade e considerei esse o ponto mais desafiador e positivo. Tentei forçar diálogos e insisti em algumas ideias para provocar respostas, mas por vezes não houve a resposta que eu esperava e isso me forçou a tomar outro rumo.”

### 2. Como você avalia o potencial de máquinas improvisadoras para seu “estudo” da improvisação livre?

“Acho interessante os recursos que as máquinas improvisadoras podem oferecer para a improvisação. Acredito que tudo depende de como o

<sup>1</sup> Disponíveis para escuta em <https://ofefo.com.br/mus/engenhoca>

improvisador lida com a improvisação, porque acredito que dependendo das escolhas que o improvisador possa ter diante da máquina e vice versa, o resultado sonoro é a dissociação de ambas as partes.”

“Avalio positivamente. Principalmente na falta de pessoas para improvisarem junto, como no caso do isolamento social. A aleatoriedade na resposta da máquina exige um exercício de escuta muito intenso e nos leva para lugares na improvisação que estão muito longe de serem óbvios. Mas, certamente, a falta de diálogo pode cansar num certo período de tempo. Sinto que podemos abrir o leque de exploração de ideias com essa ferramenta e levamos esses recursos descobertos como repertório.”

“Achei bem interessante. Foi a primeira experiência interativa que tive e percebi muito potencial para treinar e exercitar a improvisação, sobretudo neste momento pandêmico, pela questão de ser virtual. Não consegui compreender completamente o funcionamento e as vezes tinha a impressão de que ela respondia ao som que eu produzia. Por vezes achei difícil pela questão dos timbres que me geraram uma sensação repetitiva e pela falta de presença de “corpo” da máquina, falta do que seria uma interação humana ou presencial. Mas achei muito promissora e uma excelente saída para o “estudo” da improvisação. A interação com os sons provocaram um estímulo pra tocar junto.”

### 6.2.1 Comentários do autor

Como se observa pelas impressões dos participantes, apesar de haver uma fluidez na interação e a experiência no geral ser considerada intuitiva, *engenhoca* está longe de satisfazer vários requisitos que improvisadores esperam ao tocar com outros indivíduos.

Destaco a impossibilidade de prever a resposta da máquina como um ponto positivo, uma vez que garante uma interação natural da improvisação, bem como coloca a interação em um nível horizontal, já que não é permitido à pessoa o controle sobre o sistema.

Entretanto, fica evidente que a máquina peca pela ausência de uma consistência maior da variedade tímbrica. Penso que o maior problema não é necessariamente o fato de existirem somente sete categorias de timbres implementados, mas sim o fato de que o sistema circula dentre as mesmas sete categorias com uma frequência muito alta. Uma solução, por exemplo, seria garantir uma restrição para que dentro de uma mesma performance só fosse possível à máquina a utilização de uma quantidade definida de timbres. Quando a mudança de timbres é muito recorrente e não existe uma variedade grande, os improvisos carecem de uma personalidade tímbrica própria, e portanto uma restrição poderia ser uma solução simples ao problema.

Outro ponto notado pelos participantes foi a escassez ou mesmo completa ausência de momentos de silêncio por parte da máquina. Apesar de implementado, fica claro pela percepção dos musicistas que os silêncios escutados tendem a não funcionar de uma maneira natural, e é possível que isso só seja solucionado com a imposição de períodos maiores,

inclusive enquanto o musicista toca, permitindo a ocorrência de solos, ou de alternâncias entre ambos.

Também é interessante notar como parte das respostas da máquina foram entendidas como aleatórias, apesar de todo o processo empregado ser transformativo. Isso pode demonstrar tanto que os métodos transformativos atingem uma variação tão distante que a fonte de referência se torna imperceptível, como demonstra que existe uma falta de material auto-referente no pensamento da máquina, que pode soar muito variado e portanto pouco consistente para musicistas em uma improvisação em tempo real.

O distanciamento social imposto pela pandemia do novo coronavírus, uma motivação indireta deste trabalho, foi apresentado em duas respostas como um potencial interessante de máquinas improvisadoras. Acredito que sistemas do tipo possam realmente ser interessantes dado esse tipo de contexto, mas além disso, acredito que poderiam auxiliar, por exemplo, estudantes de música interessados em improvisação livre sem acesso a musicistas dedicados a essa prática.

## 7 Considerações finais

O trabalho de desenvolvimento de *engenhoca* partiu de uma revisão bibliográfica de modelos de máquinas improvisadoras descritas na literatura acadêmica, ao mesmo tempo em que o sistema era desenvolvido na linguagem de programação Pure Data. Finalizada sua implementação, o projeto pôde ser brevemente testado por três musicistas com experiência prévia na prática da improvisação livre, contribuindo com as primeiras impressões sobre o sistema.

Concluído o protótipo, é possível observá-lo a partir de seus parâmetros de funcionamento utilizando a taxonomia para improvisação de máquina apresentada em (GIFFORD et al., 2018), como demonstrado a seguir:

Tabela 1 – Classificação taxonômica de *engenhoca*.

Controle		
Aprendizagem		
Análise Musical	Melodia	X
	Classes de Alturas	X
	Tonalidade	
	Harmonia	
	Ritmo	X
	Som (timbre)	X
	Fraseado	X
	Densidade	
	Intensidade	X
	Tempo	
	Acompanhamento	
	Orquestração	X
Fonte Estética	Design do Sistema	
	Performer	X
	Corpus	
	Aprendizado	

Processos de avaliação do desempenho de máquinas improvisadoras no campo da música representam grandes desafios para a área de HCI, que costumeiramente emprega medidas de análise de tarefas para tal. E, como indicam (STOWELL et al., 2009, p. 1), “interações musicais possuem aspectos criativos e afetivos, o que significa que elas não podem ser descritas como tarefas para as quais, por exemplo, taxas de conclusão possam ser confiavelmente medidas”.

Além disso, como observado neste trabalho, a contribuição para a testagem de sistemas do tipo muitas vezes carece de uma grande população de musicistas, bem como

do acompanhamento de participantes por longos períodos de tempo, tornando o processo de avaliação consistente muitas vezes impraticável (STOWELL et al., 2009). Portanto, não é surpresa se deparar com uma escassez de processos avaliativos na pesquisa de máquinas improvisadoras de forma geral:

Pesquisadores em improvisação computacional fizeram uso limitado de métodos de avaliação formais, e muitos dos sistemas não foram descritos em detalhamento suficiente para que permitissem sua reimplementação. (GIFFORD et al., 2018)

Consequentemente, não será apresentado nenhum método de avaliação formal do protótipo desenvolvido, abrindo espaço somente para uma exposição de perspectivas para futuros desenvolvimentos sobre o que foi realizado até o momento.

## 7.1 Perspectivas futuras

Percebe-se, pela Tabela 1, uma ausência de alguns princípios não necessariamente compatíveis com um pensamento de música não-idiomática (como tonalidade ou tempo). Entretanto, tais parâmetros musicais deveriam ser facilmente distinguíveis por musicistas profissionais em qualquer contexto, e logo poderiam ser pensados como necessidades a serem implementadas futuramente. Outros elementos faltantes ao protótipo, como identificação harmônica e de intensidade, evidentemente também precisariam ser incluídos.

Além disso, é perceptível que existe uma dissociação clara entre a interpretação de uma frase musical com a forma que o modelo o representa, isto é, por meio de uma tabela musical circular com base em doze diferentes valores. Essa metodologia não é satisfatória, uma vez que ignora a organização de frases musicais maiores ou menores. Dessa forma, a identificação de características no curto prazo poderia se orientar a partir da identificação do início e do fim de uma frase, possível por exemplo pela concatenação de valores em listas iniciadas após um silêncio e finalizadas no momento em que se identifica o início de um novo silêncio.

Além disso, é evidente que carece ao sistema um trabalho com contrastes de maneira consistente, uma vez que foca-se principalmente em um acoplamento com o músico por meio da conciliação com características sonoras observadas no curto prazo.

Também não existe no *patch* a possibilidade de trabalhar com entrada MIDI, o que permitiria, por exemplo, uma maior exatidão para identificação harmônica, bem como permitiria a utilização de instrumentos eletrônicos e virtuais.

Acredito também que o desenvolvimento de *engenhoca* poderia contar com a implementação de métodos que permitissem algum nível de aprendizado, dentro do observado em uma performance, para identificação de padrões de comportamento na escuta,

que poderiam ser imitados, ou levados em consideração para os métodos transformativos já implementados, bem como para uma organização bem definida das relações entre frases musicais. Uma possibilidade nesse sentido seria o agrupamento de materiais musicais semelhantes, como acontece em *Cypher* e *OMax*, estudados no capítulo quatro.

Por fim, o autor acredita que as discussões e a demonstração da implementação apresentadas neste trabalho poderão servir de inspiração a outros pesquisadores da área, principalmente para improvisadores sem experiência com este tipo de projeto, uma vez que os testes preliminares indicam que é possível desenvolver máquinas improvisadoras com resultados satisfatórios sem a necessidade de grandes recursos humanos e computacionais.





# Referências

- BAILEY, Derek. **Improvisation: its nature and practice in music**. [S.l.]: Da Capo Press, 1993. ISBN 9780306805288.
- BODEN, Margaret. Computer Models of Creativity. **AI Magazine**, v. 30, p. 23–34, 2009. DOI: [10.1609/aimag.v30i3.2254](https://doi.org/10.1609/aimag.v30i3.2254).
- BRENT, William. A Timbre Analysis And Classification Toolkit For Pure Data. In: PROCEEDINGS of the 2010 International Computer Music Conference, ICMC 2010, New York, USA, 2010. [S.l.]: Michigan Publishing, 2010.
- BSD License Definition. [S.l.: s.n.], 2004. <http://www.linfo.org/bsdlicense.html>. Acesso: 06/08/2021.
- CAMPION, Edmund. **Natural Selection**. [S.l.: s.n.], 1996. [https://edmundcampion.com/project\\_natsel/index.html](https://edmundcampion.com/project_natsel/index.html). Acesso: 12/08/2021.
- CHORDIA, P.; RAE, Alex. Tabla Gyan: An Artificial Tabla Improviser. In: ICCM. [S.l.: s.n.], 2010. P. 155–164.
- CHOWNING, John M. The Synthesis of Complex Audio Spectra by Means of Frequency Modulation. **Journal of the Audio Engineering Society**, v. 31, n. 7, 1973.
- COLEMAN, Steve. **Rameses 2000 Project**. [S.l.: s.n.], 1999. <http://brahms.ircam.fr/works/work/7487/>. Acesso: 11/08/2021.
- COPE, David. Computer Modeling of Musical Intelligence in EMI. **Computer Music Journal**, The MIT Press, v. 16, n. 2, p. 69–83, 1992. ISSN 01489267, 15315169.
- CORBETT, John. **A Listener's Guide to Free Improvisation**. [S.l.]: University of Chicago Press, 2016. ISBN 978-0-226-35380-7.
- DEAN, Roger T. **New structures in jazz and improvised music since 1960**. [S.l.]: Open University Press, 1992. ISBN 0335098983.
- DEUTSCH, Diana. Auditory Illusions, Handedness, and the Spatial Environment. **Journal of the Audio Engineering Society**, v. 31, n. 9, p. 607, 620, 1983.
- ESSL, Karlheinz. Lexikon-Sonate. An Interactive Realtime Composition for Computer-Controlled Piano. In: SBCM. [S.l.: s.n.], 1995. P. 95–106.
- FORTE, Allen. **The Structure of Atonal Music**. [S.l.]: Yale University Press, 1973. ISBN 0300021208.
- GIFFORD, Toby et al. Computational systems for music improvisation. **Digital Creativity**, v. 29, p. 19–36, jan. 2018. DOI: [10.1080/14626268.2018.1426613](https://doi.org/10.1080/14626268.2018.1426613).

- GILLICK, Jon; TANG, Kevin; KELLER, Robert M. Machine Learning of Jazz Grammars. **Computer Music Journal**, v. 34, n. 3, p. 56–66, 2010.
- IMPROVISADO | Michaelis Online. [S.l.: s.n.].  
<https://michaelis.uol.com.br/busca?r=0&f=0&t=0&palavra=improvisar>. Acesso em 19/08/2021.
- JOHNSON-LAIRD, Philip N. Jazz Improvisation: A Theory at the Computational Level. In: HOWELL, Peter; WEST, Robert; CROSS, Ian (Ed.). **Representing Musical Structure**. [S.l.]: Academic Press, 1991. ISBN 0123571715.
- LERCH, Alexander. **An Introduction to Audio Content Analysis**. [S.l.]: IEEE Press, 2012. ISBN 978-1-118-26682-3.
- LÉVY, Benjamin. **OMax: The Software Improviser**. [S.l.: s.n.], 2012.
- LÉVY, Benjamin; BLOCH, Georges; ASSAYAG, Gérard. OMaxist Dialectics: Capturing, Visualizing and Expanding Improvisations. In: NIME. [S.l.: s.n.], 2012.
- LEWIS, George E. Improvisation and Pedagogy: Background and Focus on Inquiry. **Critical Studies in Improvisation**, 3, n. 2, 2007. DOI: <https://doi.org/10.21083/csieci.v3i2.412>.
- \_\_\_\_\_. Improvised Music after 1950: Afrological and Eurological Perspectives. **Black Music Research Journal**, v. 22, 1996. DOI: [10.2307/1519950](https://doi.org/10.2307/1519950).
- \_\_\_\_\_. Interacting with latter-day musical automata, 1999.
- \_\_\_\_\_. Why Do We Want Our Computers to Improvise? In: THE Oxford Handbook of Algorithmic Music. [S.l.]: Oxford Handbooks, 2018. DOI: [10.1093/oxfordhb/9780190226992.013.29](https://doi.org/10.1093/oxfordhb/9780190226992.013.29).
- MESSIAEN, Olivier. **The Technique of my Musical Language**. [S.l.]: Alphonse Leduc, 1944.
- PUCKETTE, Miller S. Pure Data: another integrated computer music environment. In: SECOND Intercollege Computer Music Concerts. [S.l.: s.n.], 1997. P. 37–41.
- PUCKETTE, Miller S.; APEL, Theodore; ZICARELLI, David D. Real-time audio analysis tools for Pd and MSP. In: ICMC. [S.l.: s.n.], 1998.
- AL-RIFAIE, Mohammad Majid; BISHOP, Mark. Weak and Strong Computational Creativity. In: KÜHNBERGER, Kai-Uwe (Ed.). **Computational Creativity Research: Towards Creative Machines**. [S.l.]: Atlantis Press, 2015. v. 7. P. 37, 48. ISBN 978-94-6239-084-3.
- RISSET, Jean-Claude; DUYNE, Scott Van. Real-Time Performance Interaction with a Computer-Controlled Acoustic Piano. **Computer Music Journal**, The MIT Press, v. 20, n. 1, p. 62–75, 1996.

- ROWE, Robert. **Interactive Music Systems**. [S.l.]: MIT Press, 1992. ISBN 0262181495.
- \_\_\_\_\_. Machine Listening and Composing with Cypher. **Computer Music Journal**, The MIT Press, v. 16, n. 1, p. 43–63, 1992. ISSN 01489267, 15315169.
- \_\_\_\_\_. **Machine musicianship**. [S.l.]: MIT Press, 2001. ISBN 978-0-262-18206-5.
- SANDMANN, Antônio José. **Formação de Palavras no Português Brasileiro Contemporâneo**. [S.l.: s.n.], 1988.
- SCHROEDER, M. R. Natural-sounding artificial reverberation. **Journal of the Audio Engineering Society**, v. 10, n. 3, p. 219–223, 1962.
- STOCKHAUSEN, Karlheinz. How Time Passes... **Die Reihe**, p. 10, 40, 1957.
- STOWELL, Dan et al. Evaluation of live human–computer music-making: Quantitative and qualitative approaches. **International Journal of Human-Computer Studies**, v. 67, p. 960–975, nov. 2009. DOI: [10.1016/j.ijhcs.2009.05.007](https://doi.org/10.1016/j.ijhcs.2009.05.007).
- YAVELOW, Christopher. Personal Computers and Music. **Journal of the Audio Engineering Society**, v. 35, n. 3, p. 161–188, 1987.
- ZICARELLI, David. M and Jam Factory. **Computer Music Journal**, The MIT Press, v. 11, n. 4, p. 13–29, 1987. ISSN 01489267, 15315169.