

Interação do Gesto Percussivo com Sistema Computacional Interativo na Geração de Texturas Sonoras

Cesar Adriano Traldi¹, Jônatas Manzoli²

¹Núcleo de Música e Tecnologia, Curso de Música – Universidade Federal de Uberlândia (UFU)

Avenida João Naves de Ávila, 2121 - Santa Mônica – Uberlândia – MG – Brasil.
CEP:38400-902

²Núcleo Interdisciplinar de Comunicação Sonora, Departamento de Música – Universidade Estadual de Campinas (UNICAMP)

Cidade Universitária "Zeferino Vaz" Distrito de Barão Geraldo - Campinas – SP - Brasil
CEP:13083-970

ctraldi@hotmail.com, jotamanzo@hotmail.com

Abstract. *This article presents a computer system applied to performance with percussion and live electronics. AutomaTom was designed with a set of iterative functions that generate numerical sequences in real time and these sequences control the sound output. In opposition to the computer automatic behavior, the performer interferes in the system trajectory, i.e., he/she perturbs the automaton stability by means of percussion improvisation: two sensor microphones capture variation on intensity of the performer's gestures played on a Cajon and MIDI events triggered by a control pedal. As result, this man-machine interaction produces a dynamic variation of sound textures emerging from moments of stability and instability of the system.*

Key-words: *man-machine interaction, iterative functions, sound textures, percussion*

Resumo. *Este artigo apresenta um sistema computacional aplicado à performance com percussão e eletrônicos ao vivo. O AutomaTom foi projetado a partir de um conjunto de funções iterativas, as quais geram sequências numéricas para controlar o output sonoro. Em contrapartida ao comportamento automático do computador, o performer interfere na trajetória do sistema, ou seja, perturba a estabilidade do autômato com improvisações. A captura da variação dos gestos percussivos executados no Cajon e eventos MIDI disparados por um pedal de controle é feita por dois microfones bastante sensíveis. Como resultado, esta interação homem-máquina produz uma variação dinâmica de texturas sonoras que emergem das situações de estabilidade e instabilidade que o sistema sofre.*

Palavras-chave: *interação homem-máquina, funções iterativas, texturas sonoras, percussão.*

1. INTRODUÇÃO

O uso atual de modelos computacionais e processamento em tempo real possibilita o

desenvolvimento de novas estratégias para composição musical. Em linha com esta vertente contemporânea, apresentamos um novo sistema computacional aplicado à performance com percussão e eletrônicos em tempo real: “*AutomaTom*”. Neste artigo descrevemos a aplicação de funções iterativas no processo de geração de texturas sonoras variantes no tempo. A cada passo do sistema, os resultados de uma iteração são usados como ponto de partida para a consecutiva. Essa sucessividade ou processo iterativo gera sequências numéricas que são utilizadas para controlar altura, intensidade e duração de eventos MIDI e de *wavetables*.

O desenvolvimento do AutomaTom foi derivado de oficinas de interpretação mediadas, nas quais testamos a interação de técnicas instrumentais expandidas com o processo computacional. O objetivo foi produzir dados para desenvolver novos modelos de controle sonoro e interação em tempo real, levando em consideração a interação entre músico, suporte tecnológico e gesto musical, apresentado em (Cadoz e Wanderley, 2000). Neste artigo apresentamos um recorte dessa pesquisa, através da qual criamos uma formulação utilizando funções iterativas, tendo como objetivo aproximar o modelo computacional do processo interpretativo através de mecanismos de interação em tempo real. Para isso, foram utilizados um instrumento de percussão (i.e. Cajón) e dois microfones com sensores de intensidade.

Nas próximas seções está exposto o contexto teórico e prático que fundamenta o desenvolvimento do novo sistema proposto. Bem como está descrita a implementação e estão discutidas as estratégias de performance musical que regem a interação do músico com o sistema computacional.

2. CONTEXTUALIZAÇÃO

2.1 Texturas sonoras

Do ponto de vista composicional, estamos interessados no desenvolvimento de uma escritura textural algorítmica, ou seja, queremos produzir texturas sonoras como resultado da superposição de camadas guiadas por órbitas numéricas e pelas variações de articulações e dinâmicas produzidas pelo percussionista e, para isso, partimos da análise da escritura do compositor Gyorgy Ligeti. Vários autores destacam o automatismo algorítmico e a superposição de camadas para construir texturas complexas utilizados por Ligeti; por exemplo, Clendinning (1993) denomina este método de “*composição baseada em padrões mecânicos*”.

O nosso projeto tomou como ponto de partida estas noções de automatismo no mecanismo variacional das alturas e o processo de entrelaçamento temporal de camadas, a partir dos quais abstraímos estratégias para desenvolvermos o modelo computacional aqui reportado, bem como o conceito de perturbação das órbitas numéricas geradas pelo sistema. Entendemos também que há necessidade de estabelecer-se um jogo sonoro entre o intérprete e o computador, o qual foi projetado a partir da noção de *estabilidade* e *instabilidade* produzidas no processo de performance com AutomaTom, discussão esta que será proposta mais adiante neste artigo.

2.2 Sistema dinâmico e Interação em Tempo real

O uso de sistemas dinâmicos não-lineares na composição, síntese sonora e performance ao vivo com computador foi estudado por (Manzolli; 1993). Assim como o pesquisador, Di Scipio (2001) também apresenta um método de síntese digital e composição algorítmica a partir de funções iterativas não-lineares.

Em linha com esses dois estudos anteriores, a pesquisa aqui reportada utiliza simples funções iterativas, que geram sequências de números inteiros, como mecanismo de controle de eventos sonoros. Vale a ressalva de que este tipo de interação entre sistemas computacionais e instrumentos acústicos é um tema largamente estudado na atualidade, temos trabalhos como o de Vercoe e Puckette (1985), o qual foi pioneiro na aplicação de computadores digitais na análise da performance com instrumentos acústicos, ou ainda o livro de (Miranda e Wanderley, 2006).

2.3 PRISMA: sistema e conceitos de partida

O objetivo de PRISMA foi possibilitar que o computador e o intérprete interagissem entre si e, dessa forma, produzissem comportamentos emergentes (Traldi e Manzolli, 2010), tendo como foco de estudo a possibilidade do computador realizar mudanças paramétricas influenciadas pelas ações do percussionista. O sistema, desenvolvido por (Traldi, 2009) a partir da noção de sistemas interativos e auto-organização descritos em (Manzolli, 1996; Debrun, 1996), tratava-se de uma instalação multi-modal que possibilitou a interação entre percussionista e computador, integrando os domínios sonoro, visual e espacial.

Utilizou-se também o conceito de adaptação ao *ruído informacional* desenvolvido por Ashby (1956), de forma a conciliar o sistema computacional à informação produzida pelo intérprete. Relacionou-se *ruído informacional* com distribuições de probabilidade, transformadas em tempo real pela taxa de variação dos eventos produzidos pelo intérprete. Ou seja, “*a determinação da distribuição de probabilidade de eventos futuros gerados pelo computador sujeita a uma interação dinâmica com o intérprete*” (Traldi, 2009, pg110).

O computador alterava a probabilidade dos eventos futuros de acordo com o número de ataques realizados pelo intérprete, o que foi possível através da instalação de sensores piezelétricos acoplados aos instrumentos de percussão. Estes enviavam impulsos elétricos transformados de sinal analógico para digital, os quais eram recebidos e acumulados pelo computador. A quantidade de pulsos realizados a cada segundo modificava a distribuição de probabilidade utilizada pelo computador para controlar arquivos de áudio, processamento sonoro e padrões de iluminação.

3. Novo sistema: AutomaTom

O AutomaTom foi desenvolvido a partir dos princípios apresentados na seção anterior, bem como de um modelo determinista que vale-se de funções iteradas, em contra partida ao PRISMA, no qual era utilizado um modelo estocástico. Diferente também do sistema anterior, no qual o performer interagia com um *setup* de percussão múltipla, no AutomaTom o mecanismo de interação é estabelecido a partir da execução de apenas

um instrumento de percussão (Cajon).

3.1 Automata Computacional: Função Iterativa

A unidade fundamental de controle do sistema é uma função iterativa definida através da Equação 1 abaixo:

$$X_{k+1} = A + [(X_k + B) \bmod C] \quad (1)$$

Os valores X_k são gerados recursivamente e os parâmetros de controle A, B, C são números inteiros. A título de exemplificação, associamos a sequência $\{X_0, X_1, \dots\}$ gerada pela função iterativa com valores numéricos do protocolo MIDI para *Note Number*. Neste exemplo, chamamos as sequências finitas com 7 valores de órbitas (é possível determinar o número de valores das órbitas em função dos parâmetros B e C, mas este cálculo foge do escopo deste artigo). Na Figura 1 apresentamos a partitura de sete órbitas com sete notas cada uma. Para obter este resultado, fixamos os valores dos parâmetros A=60, C=7 e o valor inicial $X_0 = 60$ e variamos o parâmetro B, restrito ao valor máximo 7. Para cada novo valor do parâmetro B geramos uma órbita finita de sete notas que estão em cada um dos compassos da Figura 1.



Figura 1. Órbitas numéricas geradas pela função iterativa definida na Eq. 1

O segundo aspecto da modelagem matemática do sistema AutomaTom é desenvolver um mecanismo para perturbar a órbita numérica gerada pela iteração da Eq. 1. Sendo assim, temos a Equação 2:

$$X_{k+1} = A + [(X_k + B) \bmod C] + D \cdot P_k \quad (2)$$

Os valores X_k são gerados recursivamente e os parâmetros de controle A, B, C, D são números inteiros. O termo da equação P_k representa a perturbação do sistema que é descrita por uma simples função aleatória de três valores, representada na Equação 3 abaixo:

$$P_k = \text{rnd}(3) - 1 \quad (3)$$

Em que $\text{rnd}(3)$ gera um número aleatório escolhido no conjunto $\{0,1,2\}$. Observa-se, também, na Eq. 2 que o tamanho do desvio da trajetória produzido pela Eq. 3 é controlado pelo parâmetro D. Por exemplo, para $D = 1$, a órbita numérica é perturbada dentro do âmbito de semitons, para $D = 2$ a perturbação será em tons inteiros e assim por diante.

3.2 Implementação

A implementação do sistema AutomaTom foi realizada em Pure Data. Utilizamos um controlador MIDI de pedal para comunicação do intérprete com o computador, dois microfones para captar os sons do Cajon (um para os sons graves e o outro para os sons agudos do instrumento) através de uma placa de som conectada a porta USB do computador. O diagrama do sistema de performance é apresentado na Figura 3.

O Cajon é um instrumento em forma de caixa que teve sua origem no período colonial do Peru. Os escravos africanos utilizavam caixas de madeira e gavetas para tocarem seus ritmos tradicionais. O Cajon moderno utilizado em AutomaTom possui um sistema duplo de captação original de fábrica com dois captadores distintos instalados em diferentes lugares para a melhor captação das frequências desejadas. O instrumento produz um som grave quando tocado na região central (também conhecido como bumbo) e um som agudo na região superior (também conhecida como caixa). Apesar da posição dos microfones valorizarem a captação dos sons graves para um e sons agudos para o outro, existe interação entre os dois microfones o que foi melhorado no AutomaTom utilizando-se um filtro de banda baixa para o microfone grave e vice-versa. A Figura 2 apresenta imagens do modelo de Cajon utilizado em AutomaTom.



Figura 2. Imagem do instrumento de percussão Cajon e detalhe do sistema duplo de captação em dois canais distintos.

Com o objetivo de construir uma hierarquia estruturada, a estratégia de programação foi partir de um conjunto de abstrações na linguagem Pure Data, as quais iniciavam-se no elemento mais básico do sistema, neste caso a função iterativa com perturbação, definida na Eq. 2. Esta primeira abstração, denominada de *iterationABC*, é apresentada na Figura 4. Na segunda camada do sistema utilizamos três abstrações *iterationABC* para controlar três parâmetros: 1) *Note Number*, 2) *Velocity* e 3) *Duração*.

Na Figura 5 apresentamos a segunda abstração do sistema denominada de *note-iteration*. As órbitas numéricas geradas por cada função iterativa são mapeadas em âmbitos diferentes e, por conseguinte, utilizamos valores iniciais diferentes para A, B, C, D. Na Tabela 1 descrevemos as condições de contorno para a aplicação das três funções. O âmbito numérico dos dois primeiros parâmetros é $[0 \dots 127]$ inteiros, relativo ao protocolo MIDI. O parâmetro *Duração* é dado em mili-segundos (ms) e nos experimentos que realizamos fixamos o seu âmbito no entorno de 250 a 3000 ms,

também valores inteiros.

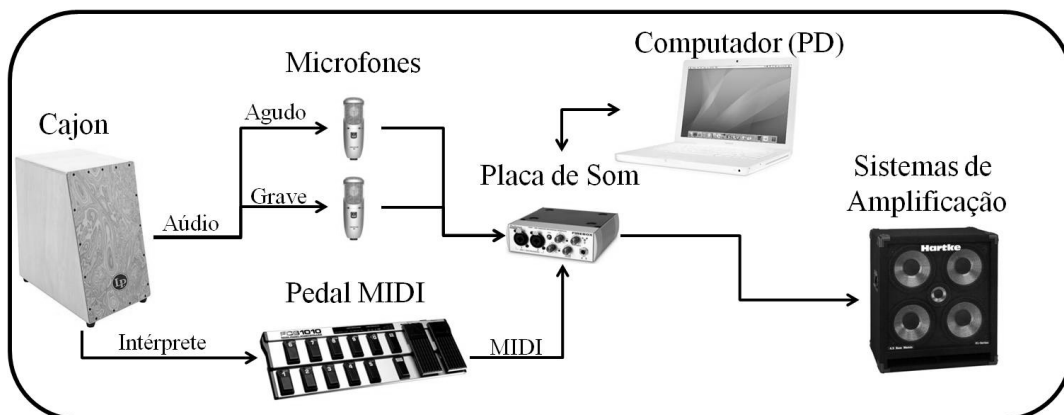


Figura 3. Diagrama do sistema de AutomaTom (MUDA SETA!!!!)

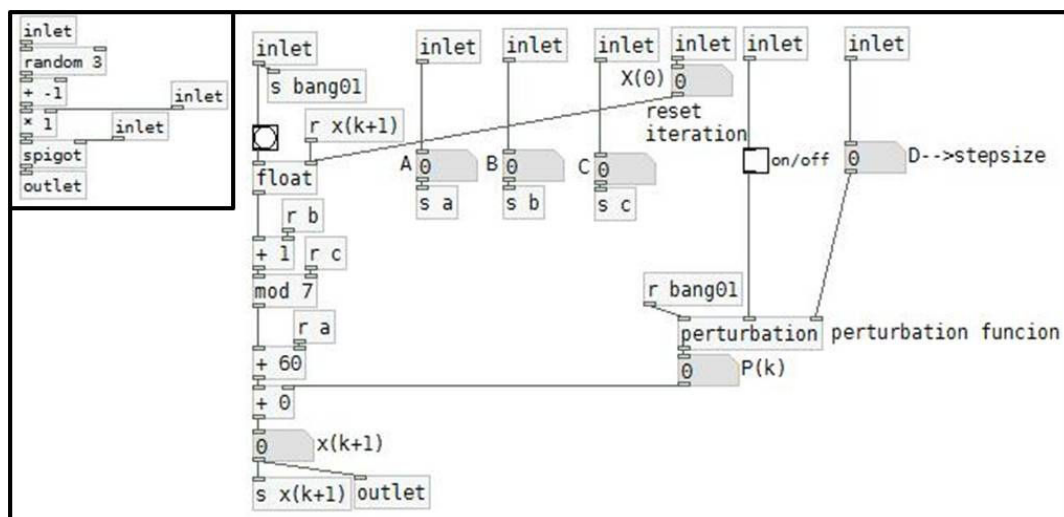


Figura 4. Implementação em Pure Data da abstração iterationABC. à direita a função aleatória apresentada na Eq. 3 e à esquerda a função iterativa descrita na Eq. 2.

Tabela 1: condições iniciais de controle das 3 iterationABC utilizadas em note-iteration

	A	B	C	X_0	D
Note Number	48	7	12	60	1
Velocity	100	7	127	60	12
Duração	250	1000	2500	1	250

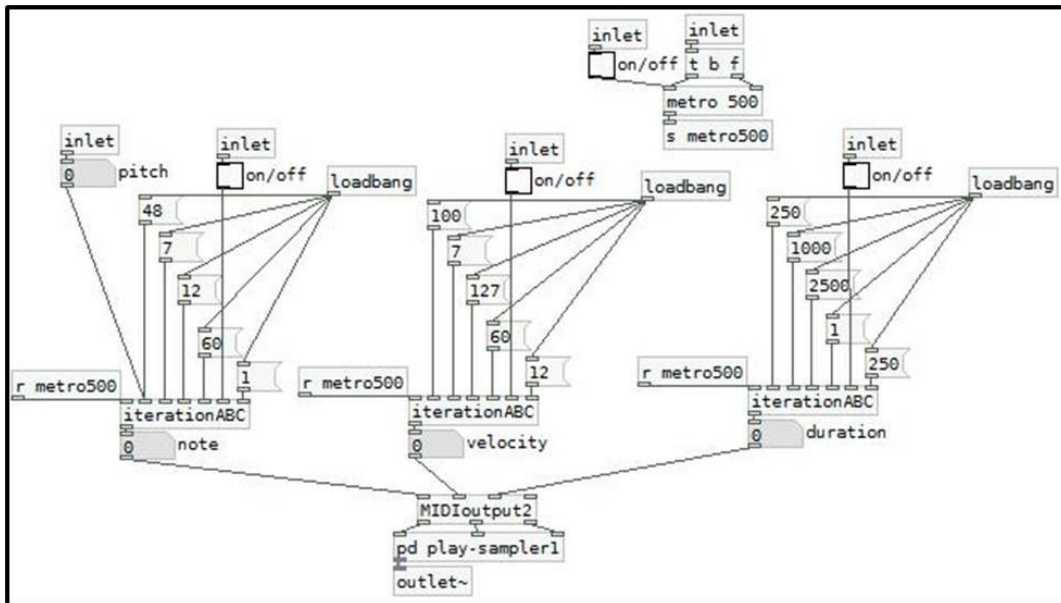


Figura 5. Implementação em Pure Data da abstração note-iteration.

O conjunto de abstrações de controle das órbitas numéricas e os seus respectivos parâmetros são chamados de musicAutomata na interface gráfica do programa (vide Figura 8). Este grupo pode ser usado, posteriormente, de forma variada em diversas implementações em Pure Data. Nas próximas seções tudo que se referir ao uso de musicAutomata será denominado simplesmente de *automata*.

3.2.1 Geração Sonora

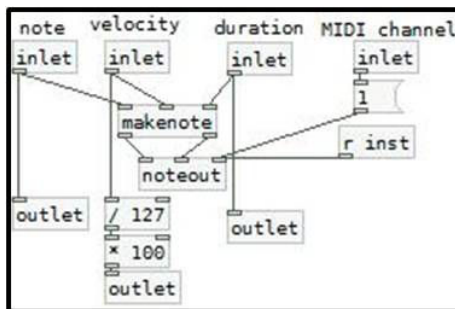


Figura 6. Abstração que gera MIDI Note eventos a partir de 3 órbitas numéricas.

Os eventos sonoros sequenciados pelas órbitas numéricas são controlados por: a) MIDI Note, eventos em que se utiliza os primitivos de Pure Data makenote e noteout, (vide Figura 6 para abstração MIDIoutput) e b) um sampler com quatro wavetables com amostras digitais do som de Cajon (vide Figura 7 para sub-patch play-sampler).

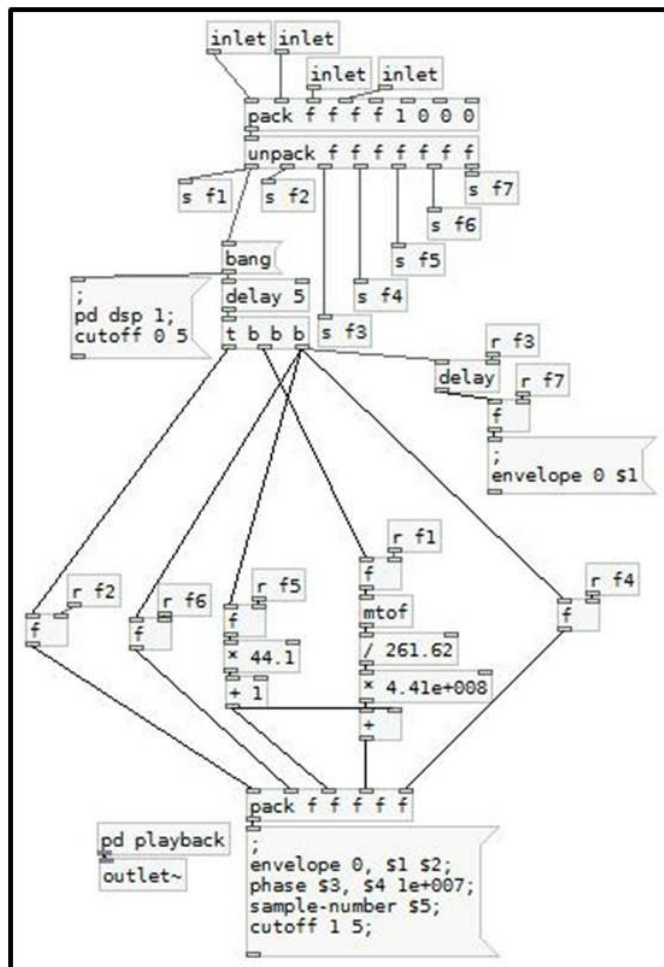


Figura 7. Sub-patch que controla a geração de notas da wavetable com amostras de Cajon.

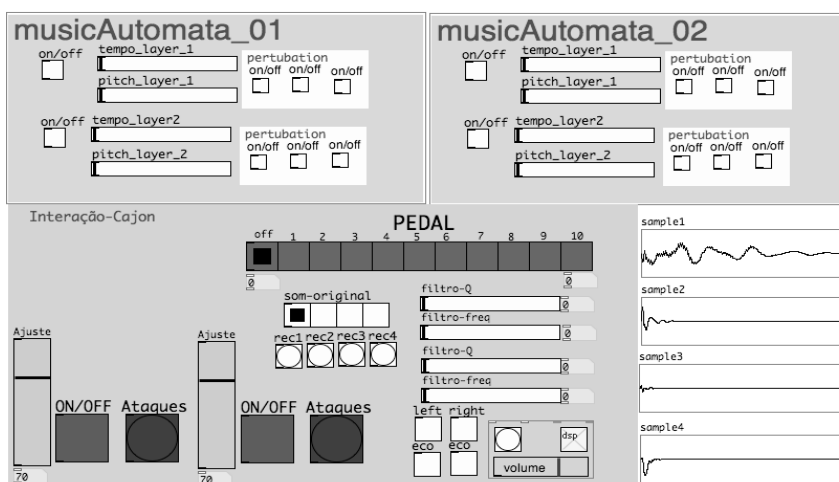


Figura 8. Interface Gráfica (GUI) do AutomaTom.

A interação com o percussionista foi realizada através de dois microfones e de um pedal de controle MIDI. A programação realizada em Pure Data possibilitou ao intérprete acionar dez controles durante a performance. Na área da GUI “*Interação-Cajon*” (Figure 8) vê-se estes controles no canto direito superior. Há abaixo do termo PEDAL as dez funções que podem ser controlados pelo mouse ou através do envio de mensagens MIDI com o pedal de controle. Através do pedal MIDI o intérprete tem controle das diferentes sessões da obra que serão apresentadas na Figura 9. As 10 possibilidades de controle com o pedal são apresentadas a seguir na Tabela 2.

Tabela 2: Funções dos Pedais MIDI	
Pedal	Função
01	Liga Automata 1 e 2. O primeiro acionamento: liga o automata 1; o segundo, o automata 2 e o terceiro desliga os dois automatatas.
02	Liga/desliga a perturbação do automata 1.
03	Liga/desliga a perturbação do automata 2.
04	Liga Automata 3 e 4 (da mesma forma que o Automata 1 e 2).
05	Liga/desliga a perturbação do automata 3.
06	Liga/desliga a perturbação do automata 4.
07	Altera aleatoriamente os valores de <i>tempo</i> e <i>pitch</i> dos quatro automatatas.
08	Aciona o sensor de intensidade do microfone 01. Após acionamento, a cada vez em que o intérprete realizar um ataque em dinâmica forte na região grave do instrumento será ligada ou desligada a perturbação dos automatatas 01 e 02.
09	Aciona sensor do microfone 02 (da mesma forma que do microfone 01), mas acionando perturbação dos automatatas 03 e 04.
10	Altera as amostras sonoras nas 4 <i>wavetables</i> entre os sons originais (pré-gravados) e sons gravados em tempo real.

Os pedais 08 e 09 ativam os sensores dos microfones do Cajon. Esses sensores de intensidade aparecem no canto esquerdo inferior da área *Interação-Cajon* na GUI (Figura 8). Antes de iniciar a performance, o intérprete ajusta com dois *sliders* o nível de dinâmica necessária para que o sensor responda de forma satisfatória. Quando os sensores são ativados durante a performance, sempre que a amplitude dos ataques for igual ou maior que a intensidade pré-definida, é disparado um *bang* que ativa/desativa as perturbações nos automatatas.

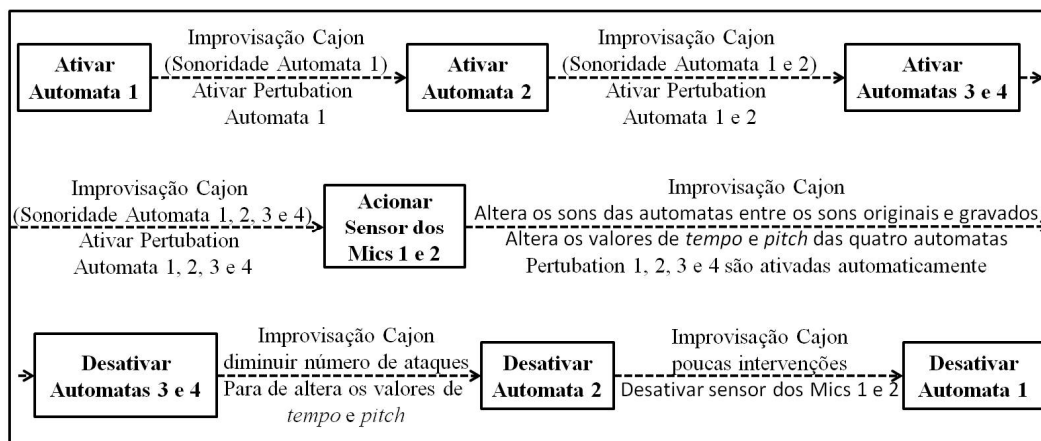


Figura 9. Guia de Improvisação utilizado na performance de AutomaTom.

3.4 Performance e Estratégias de Improvisação

Para o sistema de performance decidimos trabalhar com quatro automatadas com controle independente (vide Figura 8, há dois pares de musicAutomata). Assim, foi possível desenvolver uma estratégia de improvisação em que a performance inicia-se com um automatada ligado e há poucos eventos sonoros. Trata-se de uma textura sem superposições de camadas, por conseguinte deixa claro a constância das sonoridades e permite identificar a ocorrência de perturbações acionadas pelo percussionista. A partir desta textura inicial e com o objetivo de guiar o desenvolvimento da performance, criamos um guia de improvisação (vide Figura 9).

No processo de improvisação o intérprete buscou interagir com as sonoridades geradas pelos quatro automatadas. Nos momentos em que as perturbações eram ativadas existia grande instabilidade entre os padrões rítmicos produzidos pelo intérprete e as sonoridades geradas pelos automatadas. A cada nova perturbação ativada pelo pedal de controle, o intérprete buscou adaptar-se o mais rápido possível ao novo campo textural. Estes momentos de variação são mostrados a seguir em dois espectrogramas de gravações diferentes da performance com o sistema (vide Figura 10).

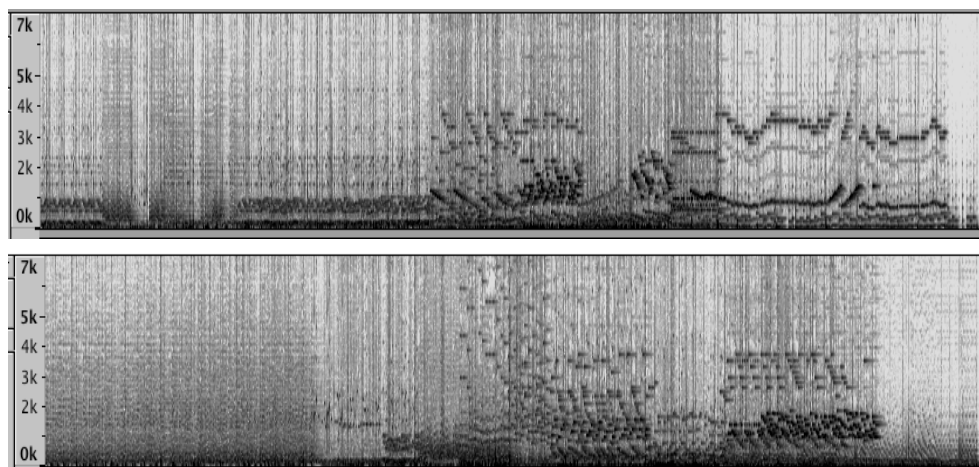


Figura 10. Dois espectrogramas de duas performances diferentes com AutomaTom.

Nas duas performances registradas nos espectrogramas da Figura 10 há um acúmulo gradual de padrões que estão grafados, aproximadamente, a partir da metade do gráfico. Esta mudança de padrão espectral coincide com o bloco “Acionar Sensor dos Mics 1 e 2” da Figura 9. A partir deste ponto há um aumento de atividade que é descrito pelo acúmulo maior de camadas horizontais que são produzidas pela perturbação que o intérprete produz improvisando padrões rítmicos com o Cajon. Após esta seção de maior atividade rítmica, o intérprete aciona sucessivamente o pedal MIDI para desativar os automatados 3 e 4, depois o automatado 2 e a peça se encerra quando o automatado 1 é desativado (vide Figura 9). Esta diminuição de atividade é registrada na Figura 10, as linhas horizontais diminuem gradualmente no final dos dois espectrogramas.

4. Discussão e Conclusão

O processo criativo e o estudo aqui reportados focam a interação entre sistemas computacionais automáticos e instrumentos acústicos via gesto interpretativo e para tal fim desenvolvemos mecanismos para interagir com o intérprete em tempo real. Estudamos o uso de processos iterativos na cadeia de ação e reação entre o computador e a improvisação do intérprete, neste caso particular um percussionista.

Com a aplicação recursiva de funções iterativas geramos órbitas numéricas que podem ser reiteradas infinitamente, mas, a fim de contrapor este comportamento automático, causador de um caráter repetitivo ao material gerado, criamos mecanismos para perturbar a estabilidade destas órbitas. Em síntese, produzimos texturas sonoras superpondo camadas rítmicas guiadas por órbitas numéricas e por variações produzidas pela improvisação do percussionista.

Em relação a interação em tempo real, quando analisamos comparativamente o processo interacional em PRISMA e em AtomaTom, notamos estratégias de improvisação completamente diferentes. O primeiro propõe o uso de distribuição de probabilidade, também uma postura adaptativa do intérprete em praticamente todo o período da performance. Enquanto que o segundo propõe funções iteradas, um intérprete que age diretamente nos resultados sonoros do computador gerando alternância de previsibilidade, bem como neste há momentos em que o intérprete antecipa as sonoridades do computador intercalados com momentos de complexidade que também exigem uma postura adaptativa.

Com relação a geração de texturas sonoras através do sistema, a Figura 10 mostra que há uma construção gradual de camadas sonoras durante o processo de improvisação. Ao comparar-se os dois espectrogramas com o guia de improvisação apresentado na Figura 9, vê-se que há a formação de textura com superposições de camadas a cada etapa da performance. Há, como pode-se constatar por uma simples inspeção visual, uma constância de padrões que permite identificar a ocorrência de perturbações acionadas pelo percussionista, apontando para existência de padrões emergentes produzidos através da interação entre o computador e a improvisação do intérprete.

5. Agradecimentos

Este estudo foi realizado em parceria entre o NICS/Unicamp e o NUMUT/UFU como desdobramento da pesquisa de doutorado (Traldi 2009) e projeto Universal CNPq. Manzolli tem o apoio de bolsa de produtividade do CNPq e Cesar Traldi coordena projeto de pesquisa financiado pela FAPEMIG.

Referências

- Ashby, W. R. (1956) "An Introduction to Cybernetics", Chapman & Hall, London.
Internet (1999): <http://pcp.vub.ac.be/books/IntroCyb.pdf>
- Cadoz, C. Wanderley, M. M. (2000) Gesture-Music in: "Trends in Gestural Control of Music". M.M. Wanderley and M. Battier, eds. 2000, Ircam - Centre Pompidou.
- Clendinning, Jane Piper. (1993) "The Pattern-Meccanico Compositions of Gyorgy Ligeti." *Perspectives of New Music* 31, pp. 193-234.
- Debrun, M. A. (1996) "Ideia de Auto-Organização", In Debrun, M., Gonzales, M., Pessoa, Jr. (orgs) *Auto-Organização: estudos interdisciplinares em filosofia, ciências naturais e humanas, e artes*. Campinas: CLE/Unicamp (coleção CLE, v. 18)., p.03-23.
- Di Scipio, A. (2001). "Iterated Nonlinear Functions as a Sound-Generating Engine". *Leonardo*, Vol. 34, No. 3, pp. 249–254.
- Manzolli, J. (1992) *FracWave sound synthesis*. Proceedings of the International Workshop on Models and Representations of Musical Signals, Capri, 1992.
- _____ (1993) "Non-linear Dynamics and Fractals as a Model for Sound Synthesis and Real Time Composition. PhD Dissertation at the University of Nottingham, UK.
- _____ (1996) "Auto-Organização: Um Paradigma Composicional", In Debrun, M., Gonzales, M., Pessoa, Jr. (orgs) *Auto-Organização: estudos interdisciplinares em filosofia, ciências naturais e humanas, e artes*. Campinas: CLE/Unicamp (coleção CLE, v. 18)., p.417-435.
- Miranda, E.R. & Wanderley, M. (2006). *New Digital Musical Instruments: Control And Interaction Beyond the Keyboard*. Publisher: A-R Editions, Inc. ISBN 0-89579-585-X.
- Traldi, C. A. (2009) "Percussão e Interatividade, PRISMA: Um modelo de Espaço instrumento Auto-organizado". 2009. 121 f. Tese (Doutorado em Música) - Instituto de Artes, Universidade Estadual de Campinas, Campinas.
- Traldi, C. A; Manzolli, J. (2010) "O Espaço Interativo Sonoro Dirigido por Distribuição de Probabilidade", *Revista EIMAS* 2010. <http://www.eimas.net/revi.html>
- Vercoe, B.L., And Puckette, M.S. (1985) "Synthetic Rehearsal: Training the Synthetic Performer". in *Proceedings, ICMC, Burnaby, BC, Canada*, pp. 275-278.