

COMPOSIÇÃO ALGORÍTMICA COM INSTRUMENTOS COMPLEXOS DE CSOUND USADOS EM “MÚSICA AMBIENTAL PARA UMA EXPOSIÇÃO”

Rodolfo Coelho de Souza

Universidade de São Paulo
rcoelho@usp.br

Resumo: O uso de funções recursivas de interrupção e reinicialização permite, em Csound, o projeto de instrumentos eficazes na composição de eventos complexos gerados através de controle algorítmico. Discute-se os comandos básicos de sua implementação e apresenta-se o exemplo de um algoritmo de instrumento complexo utilizado para gerar um fragmento da “*Música Ambiental para uma Exposição*” (2008) composta pelo autor do artigo.

Palavras-chave: Composição Algorítmica, Recursividade em Csound, Música Ambiental

Algorithmic Composition with Csound Complex Instruments used in “Ambient Music for an Exhibition”

Abstract: The use of recursive devices for interruption and retriggering allows, in Csound, the project of efficient instruments used in the composition of complex events generated by algorithmic control. The basic opcodes for its implementation are introduced and it is given the example of a complex instrument algorithm used to generate a fragment of “*Ambient Music for an Exhibition*” (2008) composed by the paper’s author.

Keywords: Algorithmic Composition, Retriggering in Csound, Ambient Music

INTRODUÇÃO

Uma das queixas frequentes dos usuários iniciantes de CSound relaciona-se ao dispêndio de tempo necessário, não só para a programação de instrumentos, mas especialmente à listagem de “notas” no arquivo “partitura” de CSound que, conforme erroneamente se supõe, envolve necessariamente tediosas entradas de dados intermináveis.

Na verdade o obstáculo que se apresenta a esses críticos é muito menos a limitação da ferramenta do que a dificuldade de se mudar a abordagem no modo de pensar a composição musical através de uma nova ferramenta. Não há dúvida que se o objetivo fosse orquestrar eletronicamente uma partitura concebida como notas executáveis manualmente em um teclado, CSound não seria o instrumento ideal, ainda que a arquitetura do programa de fato imite a estrutura mental da divisão de tarefas instrumento/partitura. Um teclado Midi, controlado por um sequenciador que executasse uma lista de notas geradas por um instrumentista seria mais prático para esse fim. Justamente por isso existem em CSound recursos para articular instruções MIDI com teclados eletrônicos.

Quanto à efetividade do uso de CSound, o problema central, como em qualquer outro instrumento de trabalho, é escolher os tratamentos que explorem as potencialidades idiomáticas da ferramenta. Por se tratar de uma linguagem de computador, são as vantagens

derivadas da automatização que justificam o uso de CSound, isto é, a recursividade e a manipulação de grandes quantidades de dados, como, no caso da música, ocorre com a forma da onda sonora digitalizada.

É mais fácil dar-se conta da utilidade de Csound para a geração de timbres novos pela extrema flexibilidade permitida em seus algoritmos de síntese, em comparação às limitações impostas pelos sintetizadores industrializados. Menos evidente é a possibilidade de embutir na “orquestra” de Csound rotinas de composição algorítmica que façam a geração automática de eventos que evitem a necessidade de longas listagens de notas nas “partituras” de CSound.

HIERARQUIAS NO DISCURSO MUSICAL

A partir da segunda metade do século vinte passou-se a dar crescente importância à inter-relação entre matéria e forma na música. Gradualmente a música passou a ser composta cada vez menos com “notas” e cada vez mais com “sons”, com os músicos agindo não “na produção de cada som da obra e sim num nível mais geral de sua estrutura” (Iazzetta 2009, p.2010). Isto é, pode-se dizer que o trabalho do compositor, em diversas circunstâncias, deslocou-se do nível local da escolha individual de cada nota, para outros níveis hierárquicos de controle.

No contraponto renascentista, o trabalho do compositor concentrava-se na escolha de cada nota. Com o advento da harmonia, as notas do acompanhamento passaram a ser escolhidas em conjuntos, com o uso da categoria hierarquicamente superior dos acordes. Durante o século XX, os agrupamentos se tornaram tão complexos que alguns compositores, como, por exemplo, Xenakis, deslocaram o nível de controle para uma hierarquia ainda superior, percebendo que a geração dos eventos primários podia ser feita por mecanismos automáticos sem prejuízo do resultado. A nota individual perdeu gradualmente sua importância como traço distintivo destes macroeventos, uma vez que não era mais percebida pelo ouvinte como evento significativo, mas apenas como parte de um evento mais complexo em que as condições da envoltória é que comandavam a significação.

Dentro desse contexto o computador é uma ferramenta muito efetiva. Apesar de que a música eletroacústica analógica já permitia a realização eficaz de texturas baseadas em eventos estocasticamente controlados, é certo que a digitalização permitiu o aprofundamento desse processo. O tratamento algorítmico em CSound permite, por exemplo, que cada detalhe da textura seja controlado individualmente, mas de modo automático, em seus parâmetros de altura, duração, ritmo, dinâmica e elementos de timbre, assim como a envoltória resultante.

O elemento chave que torna este método eficiente em CSound é a separação de tarefas entre o programa da “orquestra” e o programa da “partitura”. A “partitura” permite formatar apenas as características gerais do som, como a duração global do evento e outros parâmetros gerais da geração. Assim a “partitura” pode ficar breve e simples, mesmo quando o propósito é gerar um evento de duração relativamente longa, repleto de detalhes. O programa da “orquestra” é que se encarrega de dar conta da geração automática, mas detalhada, desses eventos. Esse tipo de estratégia permite que se supere o problema das intermináveis listas de notas que alguns usuários acreditam ser um aspecto inconveniente de CSound.

COMPOSIÇÃO ALGORÍTMICA, INSTRUMENTOS E META-INSTRUMENTOS

A composição musical através de algoritmos pareceu representar a promessa de surgimento de um novo gênero. Na prática essa possibilidade parece não ter se concretizado ainda, talvez porque importe pouco ao ouvinte por qual processo uma música é gerada quando o resultado sonoro se equipara.

Curtis Roads relata um dos principais pontos de discussão que dividem compositores e pesquisadores:

Em que nível da estrutura musical um algoritmo composicional deve ser aplicado? Um algoritmo fractal, por exemplo, pode ser usado em um nível elevado para fazer uma seleção entre diferentes frases, num nível intermediário para selecionar alturas individuais, ou num nível baixo para determinar as microvariações de uma forma de onda. Desconsiderando a consistência acadêmica, há um benefício musical a ser ganho usando-se o mesmo algoritmo em todos os níveis? Ou é melhor usar diferentes algoritmos para os vários níveis da estrutura, dando aos músicos um controle mais apropriado de cada camada? (ROADS 1996, p.908).

A resposta à questão de Roads que parece prevalecer entre os músicos, é que o controle algoritmo da obra em níveis muito altos não representa um ganho significativo, ao contrário, representa a possibilidade de decisões menos satisfatórias do que as que um compositor poderia fazer sem esforço, de modo que a relação custo-benefício do esforço de programação é negativa. Do mesmo modo, nos níveis muito baixos, a consistência algorítmica resulta num mero exercício acadêmico porque não pode ser percebida auditivamente. O terreno fértil parece estar no nível intermediário, onde se tomam decisões recursivas sobre elementos constitutivos do gesto musical, as quais, em certos casos, devido às evoluções estilísticas mencionadas acima, perderam individualmente o caráter significante.

Conforme Eduardo Reck Miranda (2000) os sintetizadores podem ser classificados pelo seu modelo de abordagem como sendo de modelagem frouxa, física ou espectral. Ele considera que:

As técnicas de modelagem frouxas (como, por exemplo, FM) são relativamente baratas e fáceis de industrializar em chips LSI, mas são de difícil controle. Ao contrário, as técnicas de modelagem física são difíceis e dispendiosas para industrializar, mas permitem controle direto. As técnicas de modelagem espectral situam-se entre esses dois extremos (MIRANDA 2000, CD-ROM).

Como nenhuma técnica de síntese pode atender sozinha às necessidades dos músicos, ele acredita que no futuro os músicos "vão deixar de colecionar módulos e teclados de sintetizadores e optar por computadores pessoais rodando softwares de síntese poderosos como CSound".

Seguindo a direção apontada por Miranda, o tipo de instrumento de CSound que estudamos neste artigo pode ser encaixado em qualquer das categorias de classificação propostas por Miranda, dependendo das unidades de síntese que ele empregar internamente ao algoritmo. Portanto a rigor este tipo de instrumento em CSound deve ser considerado um metainstrumento, ou instrumento de controle de outros instrumentos situados num nível inferior. Ele se situa num plano intermediário entre o instrumento como mero gerador de sons e o compositor enquanto gerador de linguagem.

O resultado sonoro deste tipo de instrumento agrega uma mistura de características das categorias de sons gestuais e texturais. Na leitura como "gesto", esses instrumentos permitem um bom controle de direcionalidade, de densidade de fluxo e ricas combinações contrapontísticas (empregando-se aqui o termo contraponto num sentido amplo). Enquanto "textura" eles permitem, através da geração de eventos complexos, flexibilidade rítmica e variedade de timbre do grão, características que podem faltar a outras técnicas texturais clássicas como a síntese granular.

IMPLEMENTAÇÃO EM CSOUND

Os comandos de CSound que permitem a implementação de eventos complexos formados pela geração automática de uma corrente de microeventos são principalmente o par **timeout/reinit**. A eles pode-se acrescentar os comandos **rigoto** e **rireturn** que surgem como alternativas importantes em algumas estratégias algorítmicas. Não é nosso propósito elaborar

aqui um tutorial do uso desses *opcodes*, o que pode ser encontrados em textos de referência, como no trabalho de Russell Pinkston (2000).

É necessário, porém, que se entenda o funcionamento básico dessas funções para que fiquem justificadas, conforme comentaremos adiante, as restrições apresentadas pelos instrumentos construídos com essas funções.

A essência desse tipo de algoritmo pode ser esquematizada assim:

```
; (definir idur)
start: timeout    0, idur, continue
  reinit      start
continue:
; (definir kamp, ifreq)
asig  oscil kamp, ifreq, 1
  out  asig
```

Quando o processamento encontra uma declaração **reinit**, a execução normal em “taxa-k” de controle é temporariamente suspensa e uma passagem de reinicialização começa no ponto do programa indexado pelo parâmetro do **reinit** (no caso, **start**). Nesse ponto todas as expressões processadas em “tempo-i” são recalculadas, assim como reconfigurados geradores de envelope e osciladores, além de apagados buffers de linhas de filtro e delay, até que se encontre uma declaração **endin** ou **return**.

A declaração **timeout** por sua vez, suspende a reinicialização imposta pela função **reinit** durante a vigência do seu parâmetro de duração **idur**. Com isso se fecha o loop lógico e é possível gerar-se automaticamente uma cadeia de eventos.

Os instrumentos projetados em CSound para gerar automaticamente eventos texturais complexos, através de algoritmos de decisão probabilística, apresentam resultados que na superfície podem se assemelhar aos de outros métodos, como síntese granular e síntese por *wavelets*, mas oferecem maior possibilidade de controle em pequena escala do material, permitindo assim variabilidade das texturas ao longo da evolução da frase musical.

Russell Pinkston aponta que a impossibilidade de sobrepor eventos seria uma séria limitação nesses processos de geração algorítmica de eventos múltiplos usando as funções acima descritas de **timeout/reinit**:

Técnicas como a síntese granular, que produz inúmeros “grãos” sonoros sobrepostos, ou a geração algorítmica de composições polifônicas, não podem ser implementadas dentro de um único instrumento [desse tipo]. Por outro lado, usando-se múltiplas instâncias do instrumento para gerar vozes monofônicas individuais sobrepostas, talvez se possa, algumas vezes, alcançar o resultado desejado (PINKSTON 2000, p.352).

Essa hipótese estava, naquele momento, relacionada à minha pesquisa em andamento – da qual Pinkston era orientador – que fundamentou a composição, ainda em 2000, de meu *Concerto para Computador e Orquestra*. Essa pesquisa confirmou a hipótese: a sobreposição contrapontística de fluxos lineares de eventos gerados algorítmicamente através das funções **timeout/reinit**, com múltiplas instâncias do mesmo instrumento, permite criar texturas de grande riqueza de detalhe, com interesse musical da mesma qualidade que as geradas por processos como a síntese granular.

Um instrumento complexo de Csound usado em “Música Ambiental para uma Exposição”

Em 2008, a Fundação Medeiros e Almeida, sediada em Lisboa, encomendou-me a composição de uma peça de música ambiental (isto é, do gênero *ambient music*) para ser

difundida no espaço de uma exposição da artista plástica portuguesa Leonor Alvim que apresentava uma coleção de painéis intitulada *Gueixas*, em técnica de *pano-colagem*. O prazo para a realização do projeto era curto, cerca de dois meses, e a demanda do projeto da artista era que fosse fornecida uma trilha sonora de sessenta minutos de duração (que correspondia ao prazo médio de permanência do público no prédio da exposição). A música devia ter uma correspondência com o caráter expressivo do material visual, assim como com a técnica utilizada em sua confecção (fotos dos painéis foram fornecidas para instruir esse propósito). A trilha sonora devia ser gravada em suporte digital e executada ininterruptamente enquanto a exposição estivesse aberta ao público.



Fig. 1- Foto da abertura da exposição da artista Leonor Alvim na Fundação Medeiros e Almeida de Lisboa. No canto direito vê-se um dos diversos alto-falantes que reproduziam continuamente a trilha criada para sonorizar a exibição.

Julguei que era impossível compor uma peça eletroacústica convencional no prazo exigido. Imaginei que a melhor solução era criar uma composição algorítmica em Csound que gerasse automaticamente o material sonoro com a duração desejada. A técnica predominante na trilha foi a da síntese aditiva, realizada com a sobreposição de harmônicos dinamicamente variáveis, em longos planos sonoros horizontais sobrepostos, como uma metáfora da técnica da artista que trabalhava com fragmentos de panos de muitas cores vivas, sobrepostos em composições de planos horizontais e verticais criando efeitos de perspectiva e profundidade. Para a geração dos planos sonoros, cujo principal interesse era manter uma permanente flutuação de timbres e qualidades de textura, usaram-se instrumentos complexos de Csound que permitiram gerar eventos controlados por regras probabilísticas através de laços de disparos sucessivos controlados pelo programa.

Abaixo fornecemos a transcrição de um extrato dos algoritmos utilizados. Este algoritmo de demonstração gera uma sequência sonora de 28 segundos com as características espectrais e texturais acima descritas. Note-se que marcamos em **negrito** os comandos que caracterizam as funções de recorrência dos instrumentos complexos de Csound. Enfatize-se a brevidade do arquivo-“partitura” “Leonor-01.sco” que é mais curto do que a codificação do

instrumento, uma vez que a parte mais significativa da composição algorítmica está embutida no programa do “instrumento”. Após as linhas de declaração das funções utilizadas (f01 a f11), o arquivo da partitura tem apenas onze linhas ativas de código que correspondem ao acionamento de apenas onze “notas” que geram os 28 segundos de material sonoro. A variedade do material sonoro e as qualidades do som em si, são geradas algorítmicamente pelo próprio programa do instrumento Csound.

Trecho do algoritmo de composição automática da “*Música Ambiental para uma Exposição*”

```
; "LEONOR-01.ORC"
instr 1
sr      =      44100
kr      =      4410
ksmps  =      10
nchnls =      2

idurfr =      p3      ;duração da frase
iamp    =      p4      ;amplitude média
ifreq   =      cpspch(p5) ;frequência
ifunc   =      p6      ;função
idens   =      p7      ;fator de densidade: .5(+denso) a 3(-denso)
iefat   =      sqrt(p8)      ;fator de pan esquerdo
idfath  =      sqrt(1-p8) ;fator de pan direito
iataq   =      p9
idecai  =      p10
isem1   =      p11
isem2   =      p12
idesv1  =      0.04
idesv2  =      0.02

;envoltória da frase
kgatefr  linen iamp, iataq*idurfr, idurfr, idecai*idurfr
idurnt init 1
kciclorandh 24, 1, isem1      ;kciclo = -24 a +24
kciclo =      kciclo+25      ;kciclo = 1 a 49
kciclo =      kciclo/idens      ;idens = 0.7 a 1.3
kdurnt =      10/kciclo      ;kdurnt = 0.14 a 13
start:      idurnt      =      i(kdurnt) ;idurnt = 0.14 a 13
      timeout      0, idurnt, continue
      reinit      start
continue:
kenv1 linseg      0, 0.1*idurnt, 0.5, 0.1*idurnt, 0.1,
0.15*idurnt, 0.7, 0.15*idurnt, 0.15, 0.25*idurnt, 1, 0.25*idurnt,0
kamp1 =      iamp*kenv1      ;envoltória da nota
return
kgatent      =      kamp1*kgatefr
;variação aleatória para a amplitude da nota
kvramprandh .5,kciclo/50,isem2
kgatent      =      kgatefr*(1+kvramprandh)
;geração de sinal com efeito chorus
asig11 oscili      kgatefr,ifreq,ifunc
asig12 oscili      kgatefr,(ifreq-idesv1*ifreq),ifunc
asig13 oscili      kgatefr,(ifreq+idesv1*ifreq),ifunc
asig14 oscili      kgatefr,(ifreq-idesv2*ifreq),ifunc
asig15 oscili      kgatefr,(ifreq+idesv2*ifreq),ifunc
asai =
(asig11+0.5*asig12+0.5*asig13+0.5*asig14+0.5*asig15)/10000
outs asai*iefat, asai*idfath
endin
```

```

;" LEONOR-01.SCO"
f01 0 1024 10 1 0.15 0.10 0.05
f02 0 1024 10 0 0.80 0.15 0.10 0.05
f03 0 1024 10 0 0.00 0.70 0.15 0.10 0.05
f04 0 1024 10 0 0.00 0.00 0.60 0.15 0.10 0.05
f05 0 1024 10 0 0.00 0.00 0.00 0.50 0.10 0.05 0.02
f06 0 1024 10 0 0.00 0.00 0.00 0.00 0.40 0.10 0.05
0.02
f07 0 1024 10 0 0.00 0.00 0.00 0.00 0.00 0.30 0.10
0.05 0.02
f08 0 1024 10 0 0.00 0.00 0.00 0.00 0.00 0.00 0.25
0.10 0.05 0.02
f09 0 1024 10 0 0.00 0.00 0.00 0.00 0.00 0.00 0.00
0.20 0.05 0.03
f10 0 1024 10 0 0.00 0.00 0.00 0.00 0.00 0.00 0.00
0.00 0.15 0.03
f11 0 1024 10 0 0.00 0.00 0.00 0.00 0.00 0.00 0.00
0.00 0.00 0.10
;inst start durfra amp nota ifunc density iefat iataq
idecai isem1 isem2
i01 0 24 5000 7.02 1 0.9 .3 .25 .07 .1958 .3439
i01 0 24 5000 7.07 2 1.1 .7 .25 .07 .6489 .8248
i01 4 24 5000 8.00 3 0.7 .5 .22 .23 .1170 .2226
i01 4 24 5000 8.05 4 1.2 .75 .15 .16 .5647 .6094
i01 5 18 7500 6.02 5 1.1 .5 .24 .23 .4986 .1376
i01 5 18 7500 6.07 6 0.9 .7 .17 .13 .9523 .6357
i01 6 15 7500 7.00 7 0.8 .55 .27 .24 .7944 .9609
i01 6 15 7500 7.05 8 1.2 .35 .14 .15 .4479 .4346
i01 6 15 7500 5.02 9 1.3 .65 .13 .16 .9462 .9531
i01 8 10 8500 5.07 10 0.8 .65 .26 .22 .6622 .3287
i01 8 10 8500 6.00 11 1.2 .7 .16 .14 .0944 .8248
endin

```

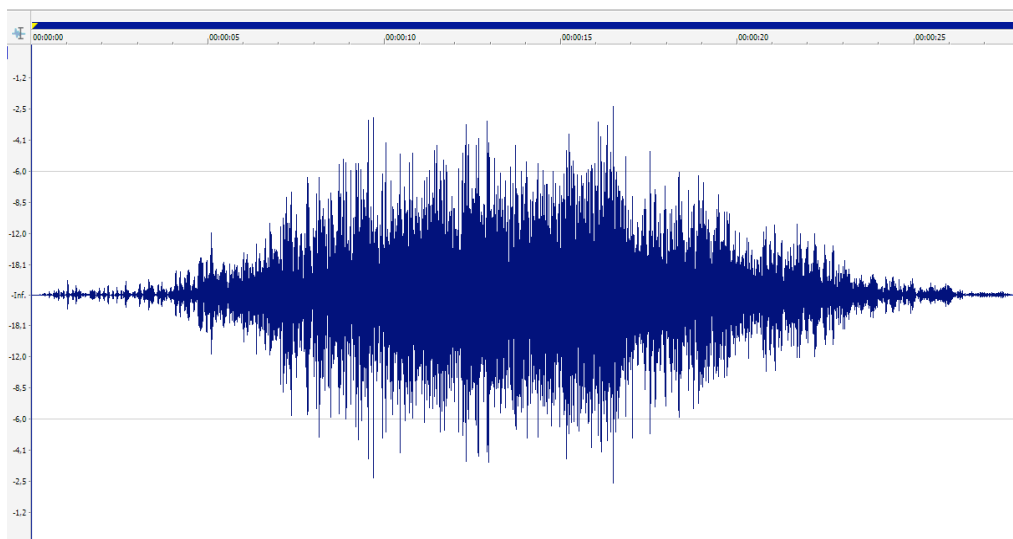


Fig. 2- Sonograma do material sonoro gerado em Csound pelos algoritmos "Leonor-01.orc" e "Leonor-01.sco"

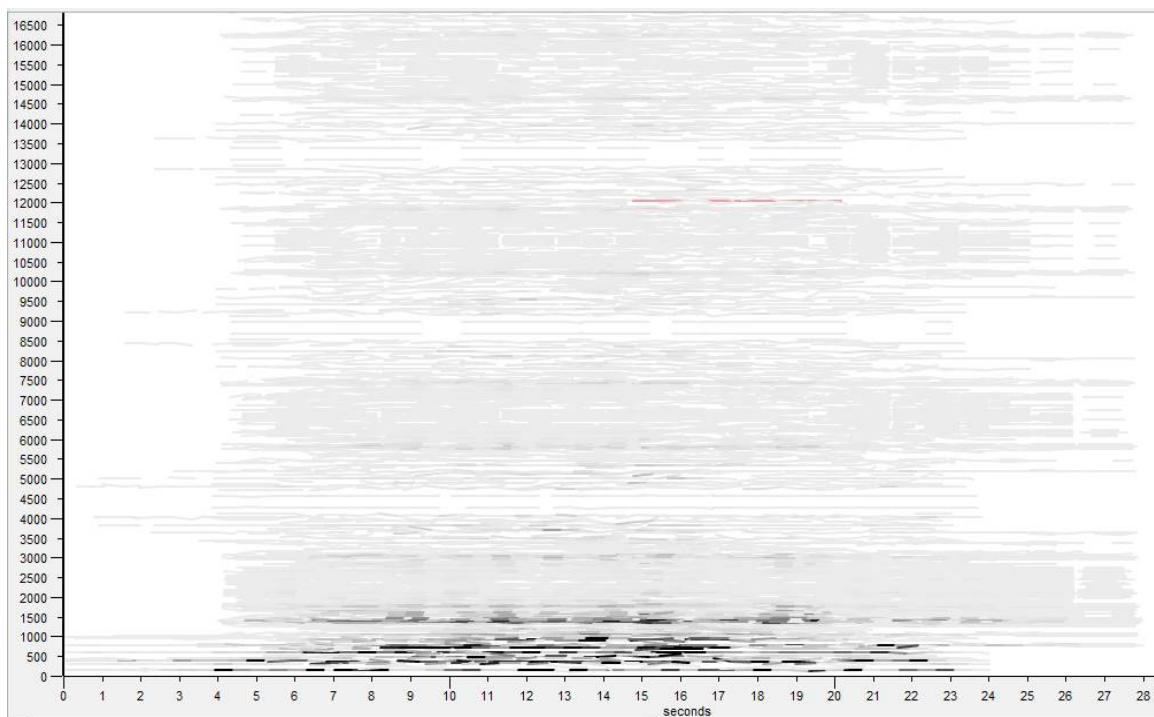


Fig. 3- Análise Espectral do material sonoro gerado em Csound pelos algoritmos "Leonor-01.orc" e "Leonor-01.sco"

As figuras 2 e 3 trazem, respectivamente, o sonograma (gerado pelo programa *SoundForge*) e a análise espectral (gerada pelo programa *Spear*) da amostra sonora gerada pelos arquivos de Csound "Leonor-01.orc" e "Leonor-01.sco" acima listados. Note-se no sonograma a grande quantidade de picos de dinâmica que indicam uma grande movimentação da textura mesmo não havendo uma predominância do caráter espectro-morfológico do tipo "ataque-impulso" e sim do tipo "gradação continuada", conforme a classificação de Smalley (1986, p.70). A análise espectral dá uma clara indicação da acumulação gradual de harmônicos das fundamentais das diversas notas disparadas ao longo da amostra, caracterizando o efeito de saturação espectral almejado.

REFERÊNCIAS

- IAZZETTA, F. **Música e Mediação Tecnológica**. São Paulo: Perspectiva, 2009.
- MIRANDA, E. R. "Three Modeling Approaches to Instrument Design." In: Boulanger, R. **The CSound Book – CD-ROM**. Cambridge, MA: MIT Press, 2000.
- PINSTON, R. "Constrained Random Event Generation and Retriggering in CSound." In: Boulanger, R. **The CSound Book**. Cambridge, MA: MIT Press, 2000.
- ROADS, C. **The Computer Music Tutorial**. Cambridge, Mass.: MIT Press, 1996.
- SMALLEY, D. "Spectro-morphology and Structuring Processes". In: Emmerson, S. *The Language of Electroacoustic Music*. New York: Harwood, 1986.