

PAPER 573

cvq: entre o meta-instrumento e a pseudo-obra

Introdução

O trabalho de composição na música eletroacústica, em especial o trabalho com sistemas interativos, nos coloca diante de situações dificilmente abarcadas por conceitos musicais tradicionais: as noções de composição e interpretação, instrumento e escritura, abstração e concretude perdem seus claros limites no processo criativo. Esta situação, do ponto de vista da criação, é antes uma vantagem do que um empecilho; mas não se deve também esquecer que estas noções e seus limites, ainda que tênues, são parte importante da apreciação estética de uma apresentação musical ao vivo. O presente artigo descreve uma experiência (em aberto) da criação simultânea de um meta-instrumento e de uma obra musical, na qual este meta-instrumento fornece os materiais sonoros e alguns limites para as idéias composicionais, ao mesmo tempo em que novas idéias composicionais sugerem modificações ou ampliações do instrumento. Importante nesse processo é o fato de que toda a geração (síntese e processamento) sonora é comandada pela própria atuação do instrumentista, seja na forma de uma resposta imediata a um gesto de execução, seja como resultado da análise de gestos realizados durante um breve intervalo de tempo imediatamente anterior ao surgimento desses novos eventos sonoros. Não há, portanto, partes pré-gravadas que devam ser reproduzidas autonomamente durante a execução da obra. Conceitualmente, instrumento e obra são homônimos; mas em notas de programa, p. ex., não vejo problemas em constar “cvq” como o título da obra, e “cavaquinho e eletrônica ao vivo” como a instrumentação, devido ao caráter fundamentalmente instrumental-interpretativo dessa proposta composicional.

O texto que se segue está dividido em quatro seções: a primeira descreve os componentes de *cvq*, a segunda os algoritmos de síntese e processamento utilizados, a terceira as possibilidades de controle nesse sistema, e por último uma descrição da versão atual desse instrumento-obra.

Descrição do sistema

O instrumento musical utilizado em *cvq* é o cavaquinho, cujo som é captado por um microfone de contato e enviado a um conversor analógico-digital. O processamento e síntese digitais são realizados com a linguagem de programação Max-Msp versão 4.1 (Cycling'74 - Ircam). Esta versão traz um novo objeto (*poly~*), que permite, dentre outras possibilidades, uma estratégia de programação dividida em módulos de processamento que podem ser efetivamente desligados quando não estiverem em uso, possibilitando assim a utilização de um número maior de processamentos diferentes dentro de um mesmo patch.

As informações de dois controladores (pedais de sustain e volume) são transformadas em mensagens Midi através de um micro-processador programável BasicStamp (Parallax), e enviadas ao patch.

Os sinais de saída de áudio são amplificados e enviados a dois alto-falantes, que, dependendo do local de execução, podem também já fornecer o retorno sonoro (monitor) indispensável para o executante.

Possibilidades sonoras

Até o presente momento, são três as principais motivações para o desenvolvimento de módulos de síntese e processamento em *cvq*: aumento da tessitura do cavaquinho, ampliação de sua paleta sonora e realização de notas sustentadas. Para as duas primeiras necessidades, existe uma referência praticamente obrigatória: o algoritmo de Karplus-Strong (1983) e suas diversas implementações e modificações. O prolongamento artificial de notas será discutido mais adiante.

A implementação do algoritmo de Karplus-Strong aqui utilizada é a baseada em *waveguides*, segundo o modelo simplificado de *single delay-loop*, como descrito em Karjalainen et alli (1998).

Sua estrutura básica é a seguinte (fig. 1):

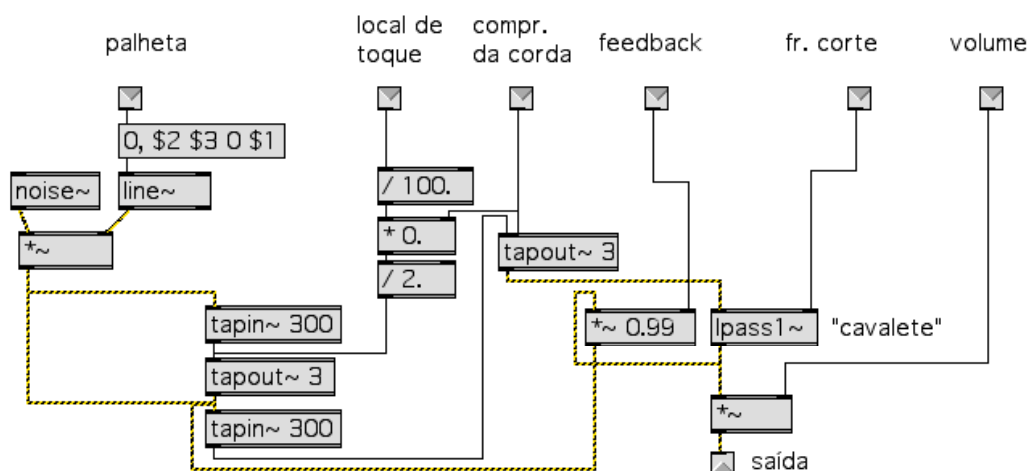


figura 1

As variáveis são: duração e forma da “palhetada”, local de toque na corda (simulado pela repetição do mesmo impulso inicial com um atraso variável, o que causa o efeito de *comb-filtering*), o comprimento da corda (que define o tamanho do loop), amortecimento/impedância da corda (controlado por um filtro passa-baixa e pela multiplicação por um coeficiente < 1.0 , de forma independente). A essa estrutura básica foram acrescentados um controle geral de amplitude e um filtro passa-alta (complementar ao passa-baixa), que também recebe o sinal do “cavalete” e se soma ao sinal de saída. A amplitude do sinal de entrada pode controlar tanto a frequência de corte dos filtros (maior amplitude corresponde a um timbre mais brilhante) quanto a amplitude e forma da

“palhetada”. O envelope triangular que modula o impulso (ruído) inicial tem três variáveis: tempo de ataque, valor máximo e decay, sendo que os dois primeiros são controláveis pelo valor máximo da amplitude de entrada. Também foi implementada uma corda grave, de afinação fixa, que ressoa por simpatia com o que está sendo tocado. O algoritmo é essencialmente o mesmo, apenas não conta com o ataque inicial; o loop é alimentado pelos sons (filtrados e com ganho reduzido) das outras “cordas”.

Através da variação destes parâmetros, mesmo que ainda não se leve em conta as possibilidades de controle por parte do instrumentista, pode-se obter sons de uma variedade considerável. Neste caso específico, são especialmente interessantes a ampliação do registro para a região grave e a produção de sons com um caráter menos metálico.

Duas questões relativas à afinação são fundamentais na implementação desse algoritmo de síntese: uma diz respeito à quantização de alturas causadas pelo número necessariamente inteiro de amostras em cada loop, a outra se deve ao atraso (variável com a frequência) causado pela filtragem. Estas questões devem ser tratadas separadamente, pois, ao contrário do filtro do algoritmo básico de Karplus-Strong, que apresenta um atraso constante (em número de amostras) para todas as frequências (ver Jaffe e Smith, 1983), o filtro aqui utilizado oferece a possibilidade de regulação de sua frequência de corte, apresentando assim um comportamento um pouco mais complexo: Mesmo assim, pode-se considerar que para cada frequência de corte há um valor relativamente constante (em número de amostras) de atraso.

Nessa discussão, é ainda necessário levar em conta peculiaridades do sistema em tempo-real implementado, que se baseia em informações vindas de um detector de alturas aplicado ao sinal de entrada. O detector utilizado é o objeto *fiddle~* (Puckette e Apel, 1998), que funciona otimamente com um vetor de processamento de 64 amostras. Este fato deveria limitar o atraso mínimo (com retroalimentação) realizado pela linguagem Max-Msp em cerca de 1.45 ms, o que em uma amostragem de 44.1 KHz representa 689 Hz, ou seja, uma altura ca. de 24 cents abaixo do fá 4 temperado¹. Esta limitação pode, no entanto, ser contornada com o objeto *poly~*, que permite a utilização de um vetor de processamento de tamanho independente do vetor do patch principal. Optou-se então pela utilização de um vetor de 32 amostras, o que transfere esse limite para uma oitava acima. Como a proposta composicional de *cvq* não tem pretensões realistas nem depende essencialmente de alturas determinadas com uma grande precisão, optou-se também pela não correção de afinação² – que pode se tornar crítica no registro agudo -, bastando o mapeamento de regiões e notas que apresentam desvios consideráveis (neste caso, maior do que 30 cents).

A tabela seguinte mostra o desvio – devido somente ao tamanho do loop - em relação à escala temperada dos sons da oitava e4-e5:

¹ A latência mínima desse sistema pode ser calculada pela soma entre o vetor de entrada/saída da linguagem Max.Msp e a latência da placa; nesse caso 128 + 298 amostras, o que significa um atraso de um pouco menos de 10 ms, valor aceitável para a proposta criativa de *cvq*.

² Realizável através de filtros passa-tudo (ver Jaffe e Smith, 1983).

nota	desvio (cents)	no. de amostras do loop
mi 4	- 3	67
fá 4	4	63
fá# 4	- 12	60
sol 4	8	56
sol# 4	3	53
lá 4	4	50
sib 4	11	47
si 4	-14	45
dó 5	6	42
dó# 5	-10	40
ré 5	-21	38
re# 5	21	35
mi 5	23	33

Tabela 1

As demais alturas temperadas mais graves não apresentam desvio acima de 10 cents, um valor aceitável para as idéias composicionais de cvq. A estes desvios deve-se somar o desvio causado pelo filtro passa-baixa utilizado ao final de cada passagem do sinal pelo loop. O filtro passa-baixa utilizado é o objeto *lpass~1* (Settel, 1998), um filtro de resposta de impulso infinita (IIR) de primeira ordem com a seguinte equação: $y(n) = k * x(n) + (1-k) * y(n-1)$, com k variando entre 0 e 1 de acordo com a freqüência de corte. Para cada freqüência de corte utilizada (entre 8 e 15 KHz), o atraso expresso em número de amostras é relativamente constante; o desvio em cents para diferentes notas do registro utilizado, nestas duas freqüências extremas de corte, é expresso pela seguinte tabela:

nota	freq. corte (KHz)	desvio (amostras)	cents ³
mi 0	8	1.75	-3
mi 1	8	1.75	-6
mi 2	8	1.75	-11
mi 3	8	1.75	-22
mi 4	8	1.72	-44
mi 5	8	1.64	-84
mi 0	15	0.47	-1
mi 1	15	0.47	-2
mi 2	15	0.47	-3
mi 3	15	0.47	-6
mi 4	15	0.47	-12
mi 5	15	0.46	-24

Tabela 2

Pode-se notar que quanto mais alta a freqüência de corte menor é o desvio causado pelo filtro; e também que quanto mais alta a freqüência desejada, maior o desvio. A exploração do registro agudo – que, por sinal, não é prioritário nessa proposta composicional - leva em conta estas peculiaridades do procedimento de síntese escolhido.

³ Esses valores foram calculados com o auxílio do objeto *filtergraph~* do Max-Msp, que fornece as informações de amplitude e fase para qualquer filtro de até segunda ordem.

Também foi desenvolvido para *cvq* um módulo de processamento especial, que prolonga indefinidamente uma nota por ele tocada. Após várias tentativas (com procedimentos granulares e transformada rápida de Fourier) chegou-se a uma solução efetiva com a seguinte estratégia:

(a) grava-se a porção inicial de uma nota

(b) cria-se um loop de cerca de 200 ms a partir desta gravação, cujos pontos de início e final são levemente variados a cada nova leitura

(c) superpõe-se quatro versões (defasadas de 1/4 de ciclo) destes loop.

Um último módulo, contrastante com os anteriormente descritos, explora timbres vocais e outros sons pré-gravados. O processamento do texto pré-gravado se dá da seguinte maneira: a frase gravada é dividida em sílabas; a cada ataque desejado, estas sílabas são misturadas, cada uma podendo sofrer uma variação de afinação (ligada a uma variação de sua duração). Os demais sons são acionados em looping a partir do momento em que a execução de notas rápidas ultrapassa determinado andamento.

Controle interpretativo

O controle da síntese e processamento sonoros em *cvq* é feito pelo próprio instrumentista, durante a execução da obra. Para isto são utilizados, além de dois pedais (sustain e volume), também informações tiradas do próprio sinal de áudio. Como já foi anteriormente mencionado, o objeto fiddle~ é utilizado com três finalidades: detecção de um ataque, detecção da nota correspondente a este ataque, detecção da amplitude deste ataque. As alturas detectadas comandam diferentes processos de seleção de alturas para a síntese de uma ou mais notas. Os ataques podem ser contados, e também o intervalo de tempo entre eles; estes dados também comandam diferentes estratégias de síntese. O nível de amplitude a cada ataque pode ser utilizado não só na definição da amplitude do som sintetizado como também na definição da frequência de corte dos filtros passa-baixa e alta. Neste caso, têm-se uma aproximação da realidade física dos instrumentos de corda, onde um ataque mais forte significa um espectro mais rico.

Os pedais assumem diferentes funções em cada seção de *cvq*. O pedal de sustain pode trocar a sub-rotina de síntese, pode determinar o momento de início da gravação do som do instrumento (no caso de seu prolongamento artificial), pode ainda disparar a frase com as sílabas embaralhadas. Dois acionamentos rápidos deste pedal servem para interromper os loopings de sons pré-gravados que não puderam ser desligados pela própria execução (especialmente nos *rallentandos* da última seção, ver abaixo).

O pedal de volume, além de sua função tradicional, pode também definir a distância entre duas notas sintetizadas, o volume do sinal do filtro passa-alta, o conjunto de notas que poderá ser sintetizado em determinado momento, o grau de desafinação de notas graves.

Uma versão

A versão atual de *cvq* consta de três grandes seções. Na primeira são explorados as diferentes possibilidades tímbricas do algoritmo implementado: sons graves e fortes,

sons menos metálicos em acordes, notas rápidas com ressonância, escolhidas em um conjunto variável de alturas.

A segunda explora as notas longas, em uma espécie de coral a três vozes.

A última seção segue a seguinte lógica: quanto maior a atividade do instrumentista (medida pelo intervalo entre cada ataque) maior a atividade de síntese e o registro por ela utilizado. A partir de certa velocidade, são disparados os loopings com sons de um relógio e de um reco-reco, ao qual se juntam também frases vocais embaralhadas em sílabas. A diminuição da atividade instrumental é acompanhada por uma menor atividade e uma maior fixidez em alturas na parte sintetizada, fato que é explorado para a finalização da peça.

A duração aproximada de uma performance é de 8 minutos.

Bibliografia

Jaffe, D. e J. Smith (1983). "Extensions of the Karplus-Strong Plucked-String Algorithm", *CMJ* Vol. 7, No. 2, pp. 56-69. Reimpresso em Roads, C. (ed.) *The Music Machine* (Cambridge: MIT Press, 1989), pp. 481-494.

Jaffe, D. (1995). "Ten Criteria for Evaluating Synthesis Techniques", *CMJ* Vol. 19, No. 1, pp. 76-87.

Karplus, K. e A. Strong (1983). "Digital Synthesis of Plucked-Strings and Drum Timbres", *Computer Music Journal (CMJ)* Vol. 7, No. 2, pp. 43-55.

Karjalainen, M., V. Välimäki e T. Tolonen (1998). "Plucked-String Models: From the Karplus-Strong Algorithm to Digital Waveguides and Beyond", *CMJ* Vol. 22, No. 3, pp. 17-32.

Puckette, M. e T. Apel, (1998). "Real-time audio analysis tools for PD and MSP" *Proceedings of the ICMC 1998*. San Francisco: International Computer Music Association, pp. 109-112.

Settel, Z. (1998). *Jimmies for MSP*, documento PDF distribuído pelo IRCAM.

Zicarelli, D. et alli. (2001). *Max4 Reference* e *MSP2 Reference*, documentos PDF distribuído por Cycling'74.