

```

(INS 24 A22
(1 43.06 ((0 0) (-1771 12) (-5355 24) (-3304 36) (0 720)) 1)
(1.3289 43.79 ((0 0) (833 11) (-2751 23) (0 720)) 1)
(0.0605 136.43 ((0 0) (4322 14) (0 720)) 1)
(0.245 43.57 ((0 0) (-3707 13) (0 720)) 1)
)
(INS 6 A23
(1 44.79 ((0 0) (5224 9) (0 720)) 0)
(1.4084 44.27 ((0 0) (6886 7) (0 720)) 0)
(1.5638 45.01 ((0 0) (4274 6) (0 720)) 0)
)
(INS 6 A24
(1 45.36 ((0 0) (2950 12) (0 720)) 1)
(0.4581 44.55 ((0 0) (-10723 13) (0 720)) 1)
(2.4536 44.80 ((0 0) (2710 10) (0 720)) 1)
)
(INS 41 A25
(1 29.92 ((0 0) (-778 46) (-428 91) (20 137) (0 720)) 0)
(1.532 40.61 ((0 0) (-575 10) (-225 20) (223 31) (0 720)) 0)
(1.9605 55.89 ((0 0) (-738 1) (-388 2) (60 3) (0 720)) 0)
)
(INS 41 A26
(1 135.47 ((0 0) (1317 93) (740 186) (264 279) (0 720)) 1)
(0.2397 139.32 ((0 0) (1202 111) (625 223) (150 334) (0 720)) 1)
(0.7683 40.93 ((0 0) (972 93) (395 185) (-80 278) (0 720)) 1)
)
(EXE 0 41)
((A24 (2 3)) 0      6      2166.678    0.8322)
((A23 (2 3)) 0      6      7127.652    0.8322)
((A22 (4)) 6       6      2826.039    0.8845)
((A21 (4)) 6       6      8340.816    0.8845)
((A22 (3)) 12      6      2826.039    0.8845)
((A21 (3)) 12      6      8340.816    0.8845)
((A22 (2)) 18      6      2826.039    0.8845)
((A21 (2)) 18      6      8340.816    0.8845)
(A26      23      18      3982.623    0.8666)
(A25      23      18      7822.052    0.8666)
(STP)
(FIM)

```

REFERÊNCIAS

- ARCELA, A. 1986. "Time-trees: the inner organization of intervals." *Proceedings of the International Computer Music Conference*, Haia.
- CASTRO, R.R.F. 1994. Implementação de SOM-A em Borland C++ para o ambiente MS-Windows, *Relatório Técnico LPE-9402*, Universidade de Brasília.
- MATHEWS, M.V. et. al. 1969. *The technology of computer music*. Cambridge, Mass.: The MIT Press.
- MEIRELES, A., GIOIA, O.G. e CASTRO, R.R.F. 1993. SOM-A em C para SUN SparcStation, *Relatório Técnico LPE-9303*, Universidade de Brasília.
- MOORER, J.A. 1977. "Signal processing aspects of computer music—A survey." *Proceedings of the IEEE* 65(8): 1108-1137.
- NOGUEIRA FILHO, V. 1988. Síntese aditiva modular — uma máquina espectral programável. *Dissertação de mestrado*, Universidade de Brasília.
- RAMALHO, G.L. 1991. SOM-A em Sun Common LISP para o ambiente Open Windows. *Relatório Técnico LPE-9105*, Universidade de Brasília.
- MULISP 1987. *muLISP-87, LISP Language Programming Environment*. SoftWarehouse Inc., Honolulu, Hawaii.

Processador de Efeitos em Sinais Digitais de Áudio

MÁRCIO DA COSTA PEREIRA BRANDÃO
 CARLOS AUGUSTO JORGE LOUREIRO
 TÚLIO DA COSTA ZANNON
Laboratório de Processamento Espectral
Departamento de Ciência da Computação
Universidade de Brasília - Brasília, DF
 CEP 70910-900 BRASIL

RESUMO

Um sistema para o processamento de sinais digitais de áudio por meio de estruturas geradas através da interligação de blocos básicos é aqui descrito. Podem ser utilizados diversos tipos de blocos básicos tais como somadores, multiplicadores, osciladores e blocos de retardo. As estruturas estão descritas por arquivos de configuração que contém a forma de interligação, juntamente com os parâmetros de funcionamento destes blocos. O objetivo principal deste trabalho é a construção de estruturas que correspondam a sistemas de ambientação de áudio, tais como filtros, reverberadores e câmaras de eco. Como é possível arbitrar-se as interligações entre os blocos, as configurações disponíveis não são limitadas como no caso da maioria dos processadores de efeitos implementados em 'hardware'. Além disto, o sistema foi idealizado de tal forma que seja fácil a criação de novos blocos através da simples incorporação de novas funções.

1. Introdução

O objetivo deste trabalho é descrever o desenvolvimento de um processador de efeitos para os sinais digitais de áudio produzidos no Laboratório de Processamento Espectral (LPE) da Universidade de Brasília. Estes sinais são gerados pelo programa de síntese ortoestereofônica SOM-A (Arcela, 1989), em conjunto com ferramentas de composição algorítmica que se fundamentam na Teoria da Árvore de Tempos (Arcela, 1984, 1986, 1991).

Através de configurações arbitrárias de interligações entre suas unidades básicas, sinais de áudio armazenados em arquivos são transformados pelo sistema, possibilitando a obtenção dos mais variados tipos de efeitos nos arquivos de áudio de saída. Como o sistema foi idealizado visando o processamento por software, sem o apoio de DSP's, não é possível a sua operação em tempo-real (Lobão, Martinelli, 1992). Esta limitação em parte é compensada pela possibilidade de sua utilização em diferentes plataformas que não disponham de recursos adicionais de hardware.

2. Estrutura Interna

A estrutura interna do processador de efeitos é composta por buffers de dados, buffers de operações e blocos básicos, como mostra a Figura 1. O sistema foi implementado em torno de uma estrutura totalmente baseada no uso de buffers para que seja possível a utilização de configurações arbitrárias para as interligações entre os blocos, além de possibilitar expansões futuras com facilidade.

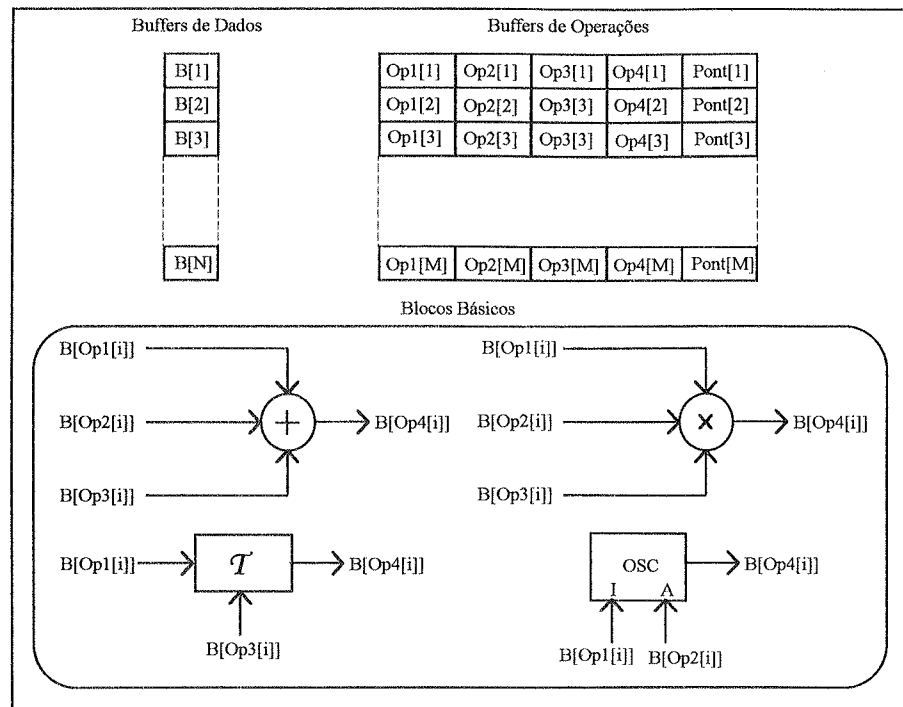


Figura 1 - Estrutura Interna do Processador de Efeitos

Na Figura 1 podemos observar que os buffers de operações armazenam dados pertinentes aos blocos utilizados na estrutura descrita pelo arquivo de configuração, sendo que os valores de entrada e saída de cada bloco correspondem a buffers de dados.

Buffers de Dados

Os buffers de dados são utilizados para armazenar dados de entrada, valores intermediários de operações e dados de saída, e foram agrupados em um vetor de elementos reais. Todas as operações do sistema recebem como parâmetros índices para este vetor.

Buffers de Operações

Os buffers de operações contém parâmetros para as funções, o índice do buffer de dados que receberá o resultado da operação, além de um ponteiro para a função a ser executada. Os buffers foram agrupados em um vetor onde cada elemento apresenta quatro campos do tipo inteiro (Op₁, Op₂, Op₃ e Op₄), e um campo do tipo ponteiro, que aponta para uma das funções que implementam os blocos básicos. Esta maneira de se efetuar as chamadas de funções permite que novas operações sejam facilmente implementadas.

O campo Op₄ indica sempre o buffer de dados que receberá o resultado da operação, e os campos Op₁, Op₂ e Op₃ indicam os buffers de dados que contém amostras de entrada ou parâmetros específicos a cada função.

Blocos Básicos

Foram implementadas funções correspondentes aos blocos fundamentais mais utilizados em processamento digital de sinais. Atualmente estão implementados os seguintes blocos básicos: somador, multiplicador, oscilador e bloco de retardo. Em sua utilização é necessário especificar quais são os buffers de entrada e de saída, além de parâmetros específicos a cada bloco, conforme mostrado na figura 1. Desta forma a interligação entre os blocos pode ser determinada pelo conteúdo dos buffers de operação. O significado dos parâmetros para cada função depende do bloco, como é visto a seguir.

Somador e Multiplicador

Os três primeiros parâmetros são índices que apontam para os buffers de dados que contém os elementos a serem somados ou multiplicados, ou seja, os valores de entrada para o bloco. O quarto parâmetro é um índice que aponta para o buffer de dados que receberá o resultado da operação, conforme mostrado na figura 1. Caso apenas duas entradas estejam sendo utilizadas, é necessário que a terceira entrada corresponda a um buffer que apresente valor inicial igual a zero no caso do somador, ou valor inicial igual a um no caso do multiplicador.

Bloco de Retardo

O primeiro e o quarto parâmetros são índices que apontam para os buffers de dados utilizados respectivamente como entrada e saída. O segundo parâmetro indica o número do bloco de retardo no esquema e o terceiro indica o retardo que será utilizado. Como este último valor corresponde a um buffer de dados que pode estar sendo utilizado como saída de outro bloco, é possível que o seu valor se altere durante o processamento do sinal.

Oscilador

O primeiro e o segundo parâmetros são índices que apontam, respectivamente, para os buffers que contém o incremento I a ser utilizado na busca em tabela, e o fator A de amplitude. O terceiro parâmetro indica o número do bloco de oscilação no esquema. Originalmente este bloco é capaz de produzir apenas a forma de onda senoidal. (Moorer, 1988).

3. O Arquivo de Configuração

É um arquivo de texto convencional, contendo linhas terminadas pelo par <Carriage Return> + <Line Feed>, de tal forma que pode ser gerado por um editor de texto convencional. Neste arquivo estão descritas as interligações, e os valores dos parâmetros específicos dos blocos a serem utilizados de acordo com o seguinte formato:

N						; Número de buffers utilizados
B _{ea}	B _{eb}					; Índices do Buffers Estéreo de Entrada
B _{sa}	B _{sb}					; Índices do Buffers Estéreo de Saída
Buf ₁						; Valor inicial para o Buffer 1
Buf ₂						; Valor inicial para o Buffer 2
...						
Buf _N						; Valor inicial para o Buffer N
M						; Numero de blocos utilizados
Op ₁₁	Op ₁₂	Op ₁₃	Op ₁₄	F ₁		; Operandos e função a ser chamada para o Bloco 1
Op ₂₁	Op ₂₂	Op ₂₃	Op ₂₄	F ₂		; Operandos e função a ser chamada para o Bloco 2
...						
Op _{M1}	Op _{M2}	Op _{M3}	Op _{M4}	F _M		; Operandos e função a ser chamada para o Bloco M

Figura 2 - O Formato do Arquivo de Configuração

As funções que serão chamadas em cada caso dependem de F_i , que atualmente pode apresentar os seguintes valores: soma = 1; multiplicação = 2; retardo = 3; oscilador = 4. Todos os valores são inteiros, exceto pelos valores Op_{ij} que são reais.

O exemplo abaixo mostra o conteúdo de um arquivo de configuração que implementa um sistema de reverberação simples (Oppenheim, 1978).

```

13          ; Número de buffers
1      7          ; Índice dos Buffers de Entrada
3      9          ; Índice dos Buffers de Saída
0.0        ; Valor inicial de B[1] = Buffer de entrada do canal A
0.0        ; Valor inicial de B[2] = Saída do somador do canal A
0.0        ; Valor inicial de B[3] = Buffer de saída do canal A
0.0        ; Valor inicial de B[4] = Saída do multiplicador do canal A
0.25       ; Valor inicial de B[5] = Retardo do Canal A
0.7        ; Valor inicial de B[6] = g
0.0        ; Valor inicial de B[7] = Buffer de entrada do canal B
0.0        ; Valor inicial de B[8] = Saída do somador do canal B
0.0        ; Valor inicial de B[9] = Buffer de saída do canal B
0.0        ; Valor inicial de B[10] = Saída do multiplicador do canal B
0.25       ; Valor inicial de B[11] = Retardo do canal B
0.0        ; Valor inicial de B[12] = Valor nulo para soma
1.0        ; Valor inicial de B[13] = Valor unitário para multiplicação
6          ; Número de blocos utilizados
3      6      13      4      2      ; B[4] = B[3] * B[6] * B[13]
1      4      12      2      1      ; B[2] = B[1] + B[4] + B[12]
2      1      5      3      3      ; B[3] = B[2] com retardo B[5]
6      9      13      10      2      ; B[10] = B[6] * B[9] * B[13]
7      10     12      8      1      ; B[8] = B[7] + B[10] + B[12]
8      2      11     9      3      ; B[9] = B[8] com retardo B[11]

```

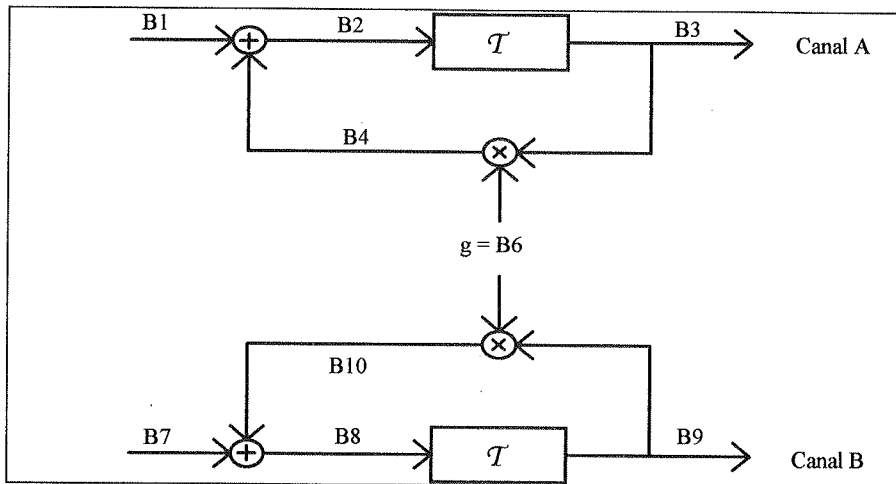


Figura 3 - Um sistema de reverberação simples

4. Considerações Finais

O Processador de Efeitos aqui descrito foi inicialmente desenvolvido em linguagem C para o sistema operacional DOS durante a disciplina "Introdução à Computação Sônica" (Lobão, Martinelli, 1992) do bacharelado em Ciência da Computação da Universidade de Brasília. Atualmente o sistema está sendo migrado para a linguagem C++ para ser utilizado no ambiente Windows.

Diversas melhorias estão sendo introduzidas, dentre as quais podemos citar:

- Suporte ao formato WAV de arquivos de áudio do ambiente Windows
- Interface gráfica para a construção de estruturas de uma forma amigável, por meio de técnicas de "drag-and-drop".
- Módulo de visualização dos sinais de entrada e de saída à medida em que estão sendo processados.
- Macros para a descrição facilitada de configurações que se repetam em uma estrutura

5. Referências

- Arcela, A. & Ramalho, G. (1991). A formal composition system based on the theory of Time-trees. *Proceedings of the ICMC, Montreal*.
- Arcela, A. (1984). As árvores de Tempos e a configuração Genética dos Intervalos Musicais. *Tese de Doutorado, PUC, Rio de Janeiro*.
- Arcela, A. (1989). A Linguagem SOM-A para Síntese Aditiva. *Anais do 1º Simpósio Brasileiro de Computação e Música, Caxambú, MG*.
- Arcela, A., (1986). Time-Trees: the inner organization of intervals. *Proceedings of the 12th International Computer Music Conference (ICMC), pp 87-89, Haia*.
- Lobao, A. S. & Martinelli, E. O. (1992). Processador de Sinais Estéreo. *Monografia da disciplina "Introdução à Computação Sônica". Departamento de Ciência da Computação, UnB, Brasília*.
- Moorer, V. F. (1988). Table Lookup Noise for Sinusoidal Digital Oscillators, *Computer Music Journal*, Vol. 1 N. 2, pp. 26-39
- Oppenheim, A. V. (ed), Blesser, B. & Kates, J. M. (1978). Digital Processing in Audio Signals. *Applications of Digital Signal Processing*. Englewood Cliffs, NJ, Prentice-Hall Inc.