

# I/O e manipulação de áudio

Prof. Regis Rossi A. Faria



# I/O e Manipulação de Áudio

- Leitura, manipulação e reprodução de som
- Envelopes e enjanelamento
- Modulações e convolução
- Sonorização estéreo e multicanal
- Laboratório de I/O e manipulação de áudio

# Entrada de áudio

- Leitura de arquivos de som
  - importação das amostras de um arquivo sonoro, sinal é armazenado em uma variável no programa
- Exemplos:
  - Pd
    - `readsf~` (*read a soundfile*)
    - `soundfiler` (*read and write soundfiles to arrays*)
    - Ver exemplos no patch `cson-io.pd` mostrando o armazenamento do áudio lido em um array
- Exemplos adicionais:
  - python (colab)

# Envelopamento

- Conceito mostrado no patch `cson-io.pd`
  - sinal  $e(t)$  que controla o nível (amplitude) de um sinal de áudio  $s(t)$
  - o envelope  $e(t)$  especifica uma envoltória para os limites da amplitude máxima do sinal de áudio ao longo do tempo
  - o envelope pode ter várias geometrias, podendo ser formado pela junção de segmentos de reta ou curvas
  - no Pd pode-se esculpir o formato do envelope em um *array* gráfico de tamanho N amostras
  - o envelope mais comum é uma onda quadrada, em que a parte alta tem nível 1 e a parte de baixo, 0

# Envelopamento

- Um exemplo de envoltória ou envelope clássico em computação musical é o ADSR (*attack, decay, sustain, release*)
- ADSR é um envelope que modela a amplitude no tempo imitando a evolução da amplitude de uma nota musical emitida por um instrumento
- O envelope ADSR tem 4 seções:
  - uma porção de ataque (início da emissão, tipicamente entre 10 e 50ms)
  - uma porção de leve e rápida queda na amplitude (*decay*)
  - uma porção mais longa de amplitude sustentada, correspondente à maior parte da duração da nota (*sustain*)
  - uma porção de redução na amplitude até zero, correspondente ao encerramento da nota (*release*)
- Exemplo: ver objeto D02.adsr.pd

# Enjanelamento

- Para processar um sinal de áudio operamos não sobre toda a sua duração de uma só vez, mas sobre porções consecutivas retiradas do sinal, uma de cada vez
- Para isto coletamos um *frame* contendo N amostras do sinal, processamos esta fatia, e então obtemos do sinal a próxima fatia a ser trabalhada
- O processo de coleta de uma porção de cada vez é chamado enjanelamento do sinal, e corresponde a multiplicar o sinal  $s(t)$  por um envelope  $w(t)$  contendo N amostras, denominado janela (*window*)

# Modulações

- Multiplicando sinais de áudio
  - Modulação em anel (*ring modulation*): ver exemplo no Pd  
E02.ring.modulation.pd
- Modelagem de onda (*waveshaping*)
  - Pode-se construir várias ondas realizando-se a leitura de tabelas de waves (*wavetables*) com varreduras dos índices de acordo com algum algoritmo (ex: loops de partes do sinal, sequenciamento da leitura de segmentos do sinal, etc)
  - Pode-se também construir ondas modulando-se a sua frequência e fase, o que leva a ondas de forma (e timbre) diferentes
  - Ver seção Waveshaping no livro texto de Miller Puckette (Theory of electronic music), cap. 5 (modulação)

# Modulações

- Modulações clássicas:
  - AM (modulação de amplitude)
  - FM (modulação de frequência)



# Convolução

- Teorema da convolução
- Exemplo de aplicação

# Convolução

- A convolução, representada por uma integral, ocupa lugar de destaque no estudo de sistemas dinâmicos lineares
- Com ela é possível descrever a resposta de um sistema quando excitado por qualquer tipo de sinal de entrada  $F(t)$  <sup>(1)</sup>. Para isto é necessário se conhecer a função de resposta do sistema ao impulso (IRF)  $h(t)$ .

(1) para condições iniciais nulas (ex: num sistema mecânico quando as condições iniciais de deslocamento e velocidade são nulas,  $x(0) = 0$  e  $\dot{x}(0) = 0$ , respectivamente)

# Convolução

- Para sistemas causais a integral de convolução pode ser escrita como

$$x(t) = \int_0^{+\infty} F(\tau)h(t - \tau)d\tau = F(t) * h(t)$$

- onde o símbolo \* representa a operação de convolução entre sinais
- Sabendo-se a  $h(t)$  do sistema, podemos calcular a resposta do mesmo à função  $F(t)$

# Convolução

- No caso de sistemas digitais, a força e a resposta impulsiva são escritas como as sequências  $h[n]$  e  $F[n]$ , e a integral discreta será

$$x[n] = \sum_{k=0}^N h[n-k]F[k] = h[n] * F[n]$$

- sendo  $N = N_F + N_h - 1$  o número de amostras contidas no sinal discreto  $x[n]$ , onde  $N_F$  é o número de amostras no sinal  $F[n]$  e  $N_h$  o número de amostras da IRF discreta  $h[n]$

# Convolução

- A operação pode ser vista como um processo de produção ou síntese de um sinal  $f(n)$  gerado pela convolução de dois sinais  $x(n)$  e  $y(n)$

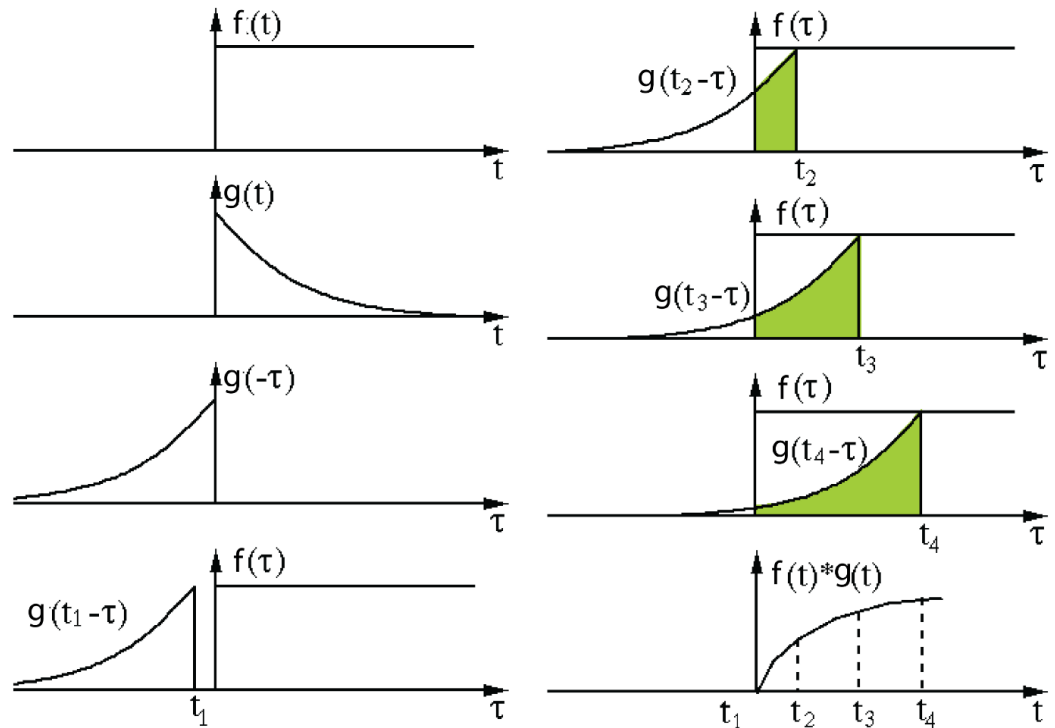
$$f(n) = x(n) * y(n) = \sum_{m=0}^{N-1} x(m)y(n - m)$$

- Cada amostra de  $x(n)$  é multiplicada por uma cópia deslocada no tempo de todas as amostras de  $y(n)$  e os resultados são somados, isto é,  $f(n)$  é uma somatória de deslocamentos escalados de cópias de  $y(n)$

# Convolução

- Interpretação gráfica

- Ver também:  
<https://en.wikipedia.org/wiki/Convolution>



# Convolução

- Uma propriedade desta operação é que no domínio da frequência ela é reescrita como

$$\begin{aligned}\text{DFT} [x(n) * y(n)] &= \text{DFT} [x(n)] \text{DFT} [y(n)] \\ &= X(n)Y(n)\end{aligned}$$

- e reciprocamente

$$\begin{aligned}\text{DFT} [x(n)y(n)] &= \text{DFT} [x(n)] * \text{DFT} [y(n)] \\ &= X(n) * Y(n)\end{aligned}$$

onde  $\text{DFT} [x[n]]$  é a transformada de Fourier discreta de  $x[n]$

# Convolução

- Exemplo de aplicação: auralização com a resposta impulsiva  $IR(t)$ 
  - Considere  $IR(t)$  o sinal que registra a resposta acústica de uma sala a um sinal impulsivo (como um transiente intenso, aproximado por um bater de palmas na sala)
  - Este  $IR(t)$  consiste numa assinatura acústica única da sala, obtida no ponto onde foi gravado
  - Considere agora um sinal de áudio  $s(t)$  qualquer
  - O sinal  $s(t) * IR(t)$  = resposta acústica que a sala terá ao ser tocada nela o sinal de áudio  $s(t)$



# Saída de áudio

- Output (exportação) de áudio
  - Exemplo Pd:
    - pode ser exportado pela saída dac~ (conversor digital analógico, que envia o sinal para o dispositivo de saída de som, chipset de áudio e alto-falantes)
    - pode ser gravado em arquivo usando-se por exemplo o objeto writesf~ (escreve em arquivo de som)
    - a reprodução de um array contendo um som consiste em mandar para a saída o valor de array(i) onde o índice i é controlado em um loop (playback convencional, sequenciando os índices de i=0 até N) ou por meio de uma variável y(n) onde i = y(n)
    - Ver cap. 2 *Wavetables and samplers* (livro texto Miller Puckette)

# Saída de áudio

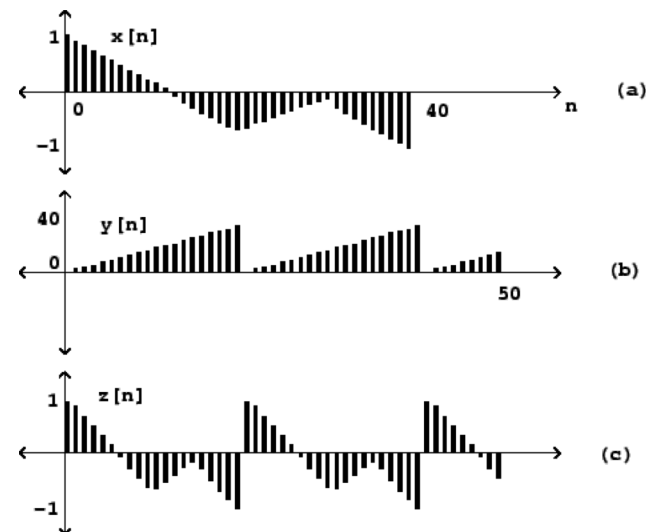
- *Wavetable lookup*

$x[n]$  é um sinal sonoro com  $N=40$  amostras e amplitudes variando entre -1 e +1

$y[n]$  é um sinal “rampa” (obtido com objeto `phasor~` no Pd) com período  $T=20$  amostras e amplitudes variando de 0 a 40

$z[n] = x[y[n]]$  é um sinal produzido varrendo-se  $x[n]$  com um índice  $i = y[n]$

Como  $y[n] = 0..40$ , então podemos acessar todas as  $N=40$  amostras de  $x[n]$



$y[n]$  controla a reprodução de  $x[n]$  → usamos os valores de  $y[n]$  como índices para buscar os valores do sinal  $x[n]$  armazenado em uma tabela (*array*)

*Ver exemplo B03.tabread4.pd no Pd (wavetable oscillators) mostrando a produção do índice  $y[n]$  usando o objeto oscilador `phasor~` que lê circularmente um array com certa frequência de leitura (que será a frequência fundamental do som ouvido)*

# Saída de áudio

- Reprodução sonora:
  - estéreo e multicanal
    - diversos tipos de *rigs* (configurações de alto-falantes) e número de canais
    - ver slides de técnicas e sistemas para reprodução sonora

# Leituras

- Reprodução (saída de áudio): 3.3.1 - A simple command-line audio file player (Audio Programming book)
  - Excerto pdf disponível no edisciplinas
  - Programas do livro em [https://github.com/meandavejustice/audio\\_programming\\_book](https://github.com/meandavejustice/audio_programming_book)
- Entrada e saída de áudio: Cap. 2 Wavetables and samplers (M. Puckette, Theory and Technique of electronic music)
  - Online em <http://msp.ucsd.edu/techniques/latest/book-html/node26.html>
- Modulação: Cap. 5 Modulation (M. Puckette, Theory and technique of electronic music)
  - Online em <http://msp.ucsd.edu/techniques/latest/book-html/node75.html>

eof