

# Segundo Trabalho Maior: Modelagem de Instrumentos por Modulação em Frequência

Data de entrega: 14/11/2014 até 23:55 pelo PACA

---

Este trabalho consiste em 3 partes:

- Leitura do artigo clássico de John Chowning intitulado “The Synthesis of Complex Audio Spectra by Means of Frequency Modulation” (1973), disponível em [https://ccrma.stanford.edu/sites/default/files/user/jc/fm\\_synthesis\\_paper.pdf](https://ccrma.stanford.edu/sites/default/files/user/jc/fm_synthesis_paper.pdf)
- Implementação dos exemplos mencionados no artigo, conforme as instruções específicas da seção 1 desse enunciado.
- Aplicação dos instrumentos em um trabalho musical, conforme as instruções específicas da seção 2 desse enunciado.

## 1 Implementação dos instrumentos

As 5 primeiras páginas do artigo servirão de introdução ao trabalho e como revisão do que vimos em aula sobre a técnica de síntese FM. A partir da página 6 o artigo começa a descrever os instrumentos propostos na linguagem Music V, e que nós adaptaremos para o Pure Data. Todas as nossas implementações em Pure Data serão abstrações com inlets de sinais (`[inlet~]`) para cada parâmetro de controle. Você pode reutilizar as abstrações mais genéricas para a criação de abstrações mais específicas (considerando que todos os patches estarão em um mesmo diretório).

Nos instrumentos do texto há sempre 3 parâmetros iniciais (P1, P2 e P3) que correspondem ao início do som do instrumento, o identificador do instrumento e a duração do evento produzido, que são utilizados em uma partitura na linguagem Music V. Estas informações serão controladas no Pure Data por estruturas de controle do próprio patch, e fora dos instrumentos, conforme descrito na seção 2.

Os instrumentos que devemos implementar são estes:

**FM simples:** implementado como `FMsimples~.pd`. Instrumento representado na figura 9 (do artigo), e que possui 4 parâmetros de controle (amplitude, frequência portadora, frequência de modulação e amplitude de modulação).

**Envelope ADSR linear:** implementado como `ADSRlinear~.pd`. Instrumento tratado como “unidade geradora” (u.g.4) que corresponde a um oscilador por consulta a tabela com um perfil indicado na figura 11, será substituído na nossa implementação por um gerador de envelope flexível como aquele que implementamos em aula, e deve aceitar sete entradas: disparo (inicia a produção) do sinal, tempo de ataque, amplitude de pico, tempo de decaimento inicial, amplitude de sustentação, tempo de sustentação e tempo de relaxamento final.

**Envelope ADSR exponencial:** implementado como `ADSRexponencial~.pd`. Pequena variação do instrumento anterior para nossas experiências, deve aceitar um parâmetro adicional  $\beta$  que modifica os trechos lineares da curva ADSR transformando as amplitudes lineares  $a$  em  $\phi(a) = \frac{1-e^{\beta a}}{1-e^{\beta}}$ . Atenção: as amplitudes de pico  $a_p$  e de sustentação  $a_s$  devem ser transformadas na geração das rampas para garantir que os valores produzidos pela curva são realmente  $a_p$  e  $a_s$  (e não  $\phi(a_p)$  e  $\phi(a_s)$ ), conforme os valores dos inlets.

**FM com envelope para o índice de modulação:** implementado como `FMdinamica~.pd`. Instrumento representado na figura 10, e que possui 5 parâmetros de controle (amplitude, frequência portadora, frequência de modulação, índice de modulação  $I_1$  e índice de modulação  $I_2$ ). Neste instrumento os valores do índice instantâneo são obtidos indiretamente a partir da curva de amplitude, adaptada linearmente para produzir valores de índice entre  $I_1$  e  $I_2$ .

**Instrumento (genérico) de metal:** implementado como `FMmetal~.pd`. Instrumento referido no texto como “brass-like tones”, consiste em usar a mesma frequência para a portadora e moduladora, e variar o índice entre 0 e 5 acompanhando o perfil da envoltória dinâmica. Deve receber 3 entradas (sinais): amplitude, frequência fundamental (instantâneas) e um parâmetro que chamaremos de *detune*, que é um pequeno valor somado à frequência de modulação, conforme o comentário do parágrafo anterior à figura 13. Conforme a frequência fundamental e a forma do envelope dinâmico deve ser possível reconstruir sons parecidos com trompetes, trombones e tubas, e efeitos como vibrato, glissando, etc.

**Envelope ASR:** implementado como `ASR~.pd`. Ilustrado na figura 12 e utilizado nos instrumentos de sopro (Woodwind-like tones). Utiliza rampas exponenciais e deve aceitar as entradas disparo, tempo de ataque, amplitude de sustentação, tempo de sustentação, tempo de relaxamento final e  $\beta$ .

**Tubo de órgão** implementado como `FMorgao~.pd`. Este é o primeiro exemplo da série de Woodwind-like tones, com parâmetros  $\frac{c}{m} = 3$  e índice de modulação variando entre 0 e 2. Deve aceitar 3 entradas: amplitude, frequência fundamental e *detune*. Atenção neste e em outros exemplos: a frequência fundamental neste exemplo é a da moduladora, não a da portadora; para cada caso deve-se traduzir o termo frequência fundamental para o oscilador correto.

**Fagote** implementado como `FMfagote~.pd`. É o segundo exemplo da série de sopros (bassoon-like tones). Deve aceitar 3 entradas: amplitude, frequência fundamental e *detune*.

**Clarinete** implementado como `FMclarinete~.pd`. É o terceiro exemplo da série de sopros (clarinet-like tones). Deve aceitar 3 entradas: amplitude, frequência fundamental e *detune*.

**Envelope exponencial:** implementado como `Rexponencial~.pd`. É o envelope ilustrado na figura 14. Deve receber como entradas o disparo, a duração e o parâmetro  $\beta$  da curva exponencial.

**Sino:** implementado como `FMsino~.pd`. Este é o primeiro exemplo da série Percussive sounds, com parâmetros  $\frac{c}{m} = \frac{\sqrt{2}}{2}$  e índice de modulação variando entre 0 e 10. Deve aceitar 2 entradas: amplitude e frequência da portadora.

**Envelope de tambor:** implementado como `ARtambor~.pd`. Ilustrado na figura 15, pode ser obtido por um envelope AR exponencial. Entradas: disparo, amplitude inicial, tempo de ataque, amplitude de pico, tempo de relaxamento e  $\beta$ .

**Tambor:** implementado como `FMtambor~.pd`. Este é o segundo exemplo da série Percussive sounds. Deve aceitar 2 entradas: amplitude e frequência da portadora.

**Bloco de madeira:** implementado como `FMblocodemadeira~.pd`. Este é o terceiro exemplo da série Percussive sounds. Deve aceitar 2 entradas: amplitude e frequência da portadora. Note que o índice usa um envelope separado e bem mais curto, do tipo produzido por `Rexponencial~.pd`.

**FM com três portadoras:** implementado como `FMtresportadoras~.pd`. Esta é uma pequena variação do exemplo da figura 17, em que são somadas 3 FMs com portadoras independentes que utilizam a mesma moduladora. De implementação análoga ao instrumento `FMdinamica~.pd`, deve aceitar as entradas: amplitude, frequências portadoras 1, 2 e 3, frequência de modulação, índice de modulação  $I_1$  e índice de modulação  $I_2$ . Uma sugestão de utilização desse instrumento é usar a moduladora como frequência fundamental e as portadoras como regiões formantes, para gerar sons de vogais a partir das tabelas de formantes da tabela 3-10 (página 292) do Moore.

## 2 Aplicação musical

Nessa parte do trabalho cada um utilizará os instrumentos implementados para produzir uma peça-patch que dure entre 3 e 5 minutos.

O objetivo central não é criar peças originais ou bonitas, mas explorar as possibilidades de timbres da síntese FM.

Os trabalhos podem ser totalmente inventados/compostos ou reproduzirem peças já existentes.

É necessário que a peça-patch produza os eventos/parâmetros e disparos a partir de um comando do usuário, ou seja, não é para enviar gravações, mas um patch (chamado `peça<NUMEROUSP>.pd`) com um botão do tipo “para tocar aperte aqui”.

Sequências complicadas de valores e parâmetros podem ser armazenadas em arquivos externos e lidas pelo patch (por exemplo usando `[coll]` para ler arquivos texto ou `[seq]/[midiparse]` para ler arquivos MIDI).

## Bom Trabalho!