

Notação paramétrica por inteiros: timbre, tom, ultrassom

MODALIDADE: COMUNICAÇÃO EM SIMPÓSIO

SIMPÓSIO: ST-13. Música, Cultura Digital e Epistemologias da Escuta: caminhos entrecruzados na produção de sons para um novo tempo

Peter Pereira Marques
UFBA
peterpm@ufba.br

Resumo. Apresento a especificação de uma notação, não necessariamente sonora/musical, baseada em números inteiros. Demonstro então algumas aplicações possíveis para tal abordagem notacional, dito Timbre, Tom, e Ultrassom. Quanto ao Timbre, pode-se utilizar diversas linhas de voz para controlar os parâmetros de um sintetizador. Uma literal transferência das "linhas de automação" para dentro da notação documental. Quanto ao Tom, o chamado microtonalismo racional, foi implementado utilizando duas linhas, onde a melodia é definida como uma sequencias de frações. Quanto ao Ultrassom, surge na notação microtonal uma necessidade de notacionar frequências por extenso, e, ao estender a matemática, obteve-se uma notação programática formal, que permitiu tal implementação, o que habilita escrever virtualmente qualquer numero real, sendo portanto, capaz de notacionar emissões ultrassônicas. Discute-se o uso da notação em aplicações não-sonoras, com enfoque na parametrização rítmica de elementos/objetos.

Palavras-chave. Síntese modal, Música computacional, Música experimental, Microtonalismo racional, Música ultrassônica.

Title. Integer Based Parametric Notation: Timbre, Tone, Ultrasound

Abstract. I present the specification of a notation, not necessarily sound/musical, based on integers. I then demonstrate some possible applications for such a notational approach, namely Timbre, Tone, and Ultrasound. As for Timbre, you can use several voice lines to control the parameters of a synthesizer. A literal transfer of "automation lines" into documental notation. As for tone, the so-called rational microtonalism was implemented using two lines, where the melody is defined as a sequence of fractions. As for Ultrasound, in the microtonal notation there was a need to notate frequencies in full, and, by extending mathematics, a formal programmatic notation was obtained, which allowed such implementation, which enables writing virtually any real number, therefore being capable of notating ultrasonic emissions. The use of notation in non-sound applications is discussed, focusing on the rhythmic parametrization of elements/objects.

Keywords. Modal Synthesis, Computational Music, Experimental Music, Rational Microtonalism, Ultrasonic Music.

Introdução

Pluripotência, a competência quase divina de não restritividade operacional, e música, ou melhor, composição, a arte de utilizar campos de escolhas de maneira intencional.

Agora, como utilizar um campo de escolhas que, não existe por completo? A potencialidade criativa torna-se restrita, contra-intuitivamente fortalecendo a criatividade de maneira indireta, através do engenho. Porém, aqui, buscamos uma forma diretamente imediata de liberdade.

A Síntese Modal, o Microtonalismo Racional e o Som Ultrassônico, integrados pelos paradigmas computacionais arranjados, podem oferecer um vislumbre de tal *afordance* integral.

Implementação

Utilizando software totalmente de código aberto, foi possível implementar a primeira competência pluripotente necessária, o paralelismo potencialmente infinito.

Softwares

O arranjo funciona de tal forma que temos 3 softwares interagindo entre si, Musescore, Plugdata e Godot Engine. Vamos as suas funcionalidades específicas.

Musescore

Com uma partitura totalmente customizada, com arquivo modificado via editor de texto externo, que com um simples processo formal, pode ser expandida para um numero N de instrumentos simultâneos. Isso permite construir uma partitura com quantas vozes forem necessárias para implementação de processos, sem que seja necessário executar os N*passos de criação e modificação das configurações do instrumento dentro do musescore, via cliques na interface gráfica.

Plugdata

Uma versão do puredata convertido em VST, que pode ser utilizado como instrumento virtual, permitindo que cada uma das vozes tenha uma instancia individual do puredata. Isso é utilizado primariamente para construir uma rede computacional de inputs.

Sendo assim, cada faixa da partitura tem uma conexão persistente com um servidor local, isolada. O que permite tratar cada linha da partitura como equivalente a uma das conexões de rede, automaticamente.

Godot Engine

Motor com linguagem de programação similar ao python, e capacidade de integração com quaisquer linguagens externas via módulos nativamente, que funciona como servidor bidirecional. Cada faixa da partitura, através do plugdata, conecta-se ao servidor godot, e envia seus pares nota/velocidade, e recebe de volta valores arbitrários. Como testes, foram implementadas duas peças microtonais e teste de geração de tom ultrassônico, onde a lógica de afinação é totalmente contida no godot, e a síntese no plugdata.

Arquitetura

Sendo assim, temos uma *stack* que nos permite notacionar qualquer número de vozes paralelas, as processar individualmente, e utilizar quaisquer paradigmas algorítmicos, retornando seja qual o resultado pretendido. Vamos então conjecturar um sintetizador modal controlado totalmente pela partitura.

Aplicações

Como visto, temos 3 softwares interagindo entre si:

Musescore, que oferece N vozes para as parametrizações;

PlugData, que permite conexão a rede e as sínteses sonoras;

Godot Engine, usa a conexão bidirecional para a lógica/cálculos, operacionalizando.

Timbre como elemento notacional

Aliar-se-á uma implementação de um sintetizador a partitura em si, buscando integrar peça e síntese, no próprio diálogo musical.

A síntese modal, que modela sistemas acústicos, cresce em potencial junto com a quantidade de modos (“ressonâncias discretizadas”), porém, podemos encontrar diversas abordagens que permitem reduzir o número de modos e manter alta performance tímbrica, simultaneamente, via otimizações.

Uma destas otimizações consiste em utilizar um sistema de partículas. Este sistema funciona com duas abstrações principais, “Massa” e “Elo”. Massa é uma representação de um

ponto no espaço, que pode mover-se ao reagir a impulsos, computando sua posição, e o Elo é o que une duas ou mais Massas, com uma rigidez e atenuação especificáveis.

A interação entre constructos arbitrários utilizando essas duas abstrações, gera movimentos de alta complexidade emergente, que podem ser quantizados e sonorizados, gerando uma síntese de modelagem física (Henry, 2004).

Outra abordagem, mais focada na otimização numérica da cardinalidade do conjunto de modos, modela um sistema preditivo do comportamento de sistemas modais, utilizando um procedimento de transformação matemática aplicada ao conjunto original, gerando um computacionalmente mais eficiente, ou seja, um numero menor de modos que aproxima-se do comportamento do conjunto original, obtendo menor complexidade algorítmica.

Pode-se também modelar sinais e ir aproximando-se, gerando então ondas com periodicidade, que podem ser utilizadas de maneira a gerar timbres, de forma análoga.

É interessante observar os parâmetros dos cálculos, que nos oferece uma contextualização de cada fator envolvido no processo, que poderiam ser controlados pela partitura. Temos:

Atenuação; Rigidez; Massa; Deslocamento e Velocidade; Vetor força instantâneo; Vetor força da conexão entre massas; Coordenada do enésimo modo e sua velocidade; Solução da posição e velocidade de um determinado modo, baseando-se em suas interconexões e forças externas; Frequência natural(ressonante) do sistema; etc (Yee; Tsuei. 1990).

Podemos exemplificar utilizando como base o sintetizador “ep-mk1” de Moreno, pois é implementado em PD, via síntese modal, e código aberto. Cada faixa de automação, pode ser predefinida, buscando o menor numero viável, que possa gerar timbragens com diversas coloraturas, operando as variáveis do sintetizador, através da notação rítmica e musical da própria partitura.

Tal parametrização, seria habilitada pela rede Puredata/Godot, onde cada faixa da partitura Muscore opera como faixa de automação, controlando individualmente um dos diversos parâmetros do sintetizador, em tempo real, durante a execução musical, juntamente com as informações de tom/melodia.

Microtonalismo Racional

Ao utilizar da notação por inteiros, podemos notacionar os números racionais, e obter uma tonia baseada em razões numéricas, ao contrario de uma baseada na divisão igualitária da oitava.

Também, apresento uma formulação que permite aproximar-se da oitava (e outros intervalos), para esse tipo de melodia.

Por exemplo, o ciclo de quintas: $3/2$ $3/2$ $3/2$ $3/2$. Ou até mesmo o ciclo de quartas em marcha harmônica: $4/3$ $2/3$ $4/3$ $2/3$.

Com isso, podemos compor melodias cujos intervalos são sempre puros, tanto ascendentemente quanto descendentemente, de forma notacional.

Como a divisão por zero é indefinida, podemos definir uma operação que a defina. Sendo assim, uma sequencia de frações com denominador zero, servem como entrada arbitraria de dígitos. Por exemplo, para iniciar uma melodia com 440 Hz, seguida de uma quinta acima, devemos notar: $4/0$ $4/0$ $0/0$ $1/1$ $3/2$. Podemos notacionar até mesmo a virgula, ao omitirmos o numerador: $\emptyset/0$. Sendo assim, 256,402, tonando = $2/0$ $5/0$ $6/0$ $\emptyset/0$ $4/0$ $0/0$ $2/0$ $1/1$.

Podemos escrever então uma escala? Certamente. Para isso precisamos dos intervalos em questão. As escalas tradicionais são compostas de segundas menores e maiores.

A segunda maior seria o intervalo que eleva uma frequência duas partes de doze, exponencialmente, tal como definida pela divisão igualitária.

Sendo assim, temos as formulas $2^{(1 \div 12)} \approx 1.059463094$ para a segunda menor e $2^{(2 \div 12)} \approx 1.122462048$ para a segunda maior.

Como as escalas diatônicas são formadas por combinações de 5 tons e 2 semitons, como por exemplo, TTsTTTs, podemos obter o valor da resultante da oitava de todas as combinações com a formula $((2^{(2 \div 12)})^5) \times ((2^{(1 \div 12)})^2) = 2$.

Para que seja notacionado em razão numérica, precisamos nos aproximar dos intervalos exponenciais usando a serie harmônica, onde os harmônicos 17 e 18, representados pelas fracções $17/16$ e $9/8$ são os representantes da segunda menor e maior, respectivamente.

Portanto, pela serie harmônica temos $((9 \div 8)^5) \times ((17 \div 16)^2) \approx 2,0343257$.

Podemos nos perguntar, quais todas as aproximações possíveis? Podemos plotar tal mapa através da serie de funções:

Para 1 tipo de intervalo, $(A/B)^N \approx 2$

Para até 2 tipos de intervalos, $((A/B)^N) * ((C/D)^O) \approx 2$

Para até 3 tipos de intervalos, $((A/B)^N)^*((C/D)^O)^*((E/F)^P) \approx 2$

Então, generalizando para até N tipos de intervalo temos:

$((d_1/n_1)^{p_1})^*((d_2/n_2)^{p_2})...((d_N/n_N)^{p_N}) \approx 2$ com d, n, p inteiros positivos.

Desenhar essa função, ordena todas as melodias por sua aproximação a oitava, incluindo as exatas, da única melodia de duas notas que satisfaz, o próprio salto de oitava, ate melodias com enésimas notas. Também poderíamos utilizar outros valores além de 2(oitava acima). Por exemplo, oitava abaixo seria 1/2.

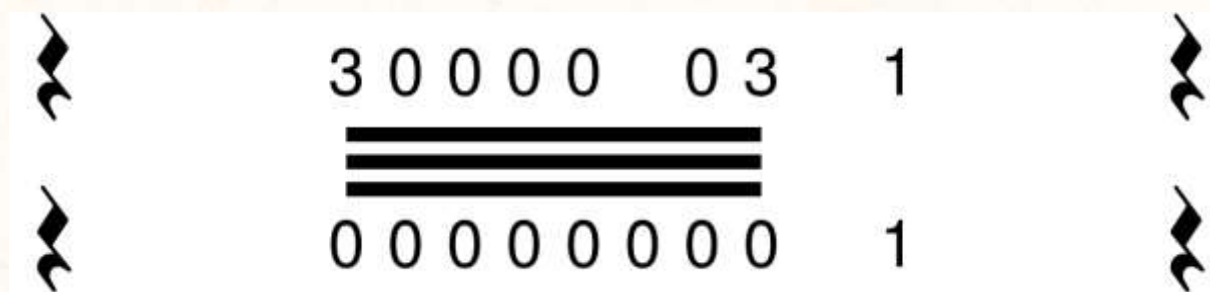
Frequências ultrassônicas arbitrárias

Como vimos na notação microtonal, podemos especificar uma frequência diretamente, usando notas na partitura. Sendo assim, é possível escrever frequências ultrassônicas acima de 20 000 Hz, que denominamos som ultrassônico (Yamashita et. al., 2008, p. 260), cujas notas não existem no escopo MIDI/Partitura tradicionalmente.

Por exemplo, a frequência 30,03 kHz, tonando, é escrita com a melodia de frações 3/0 0/0 0/0 0/0 0/0 0/0 0/0 3/0 1/1.

Abaixo, temos sua notação personalizada em musescore:

Figura 1 – 30 000,03 Hz notacionado.

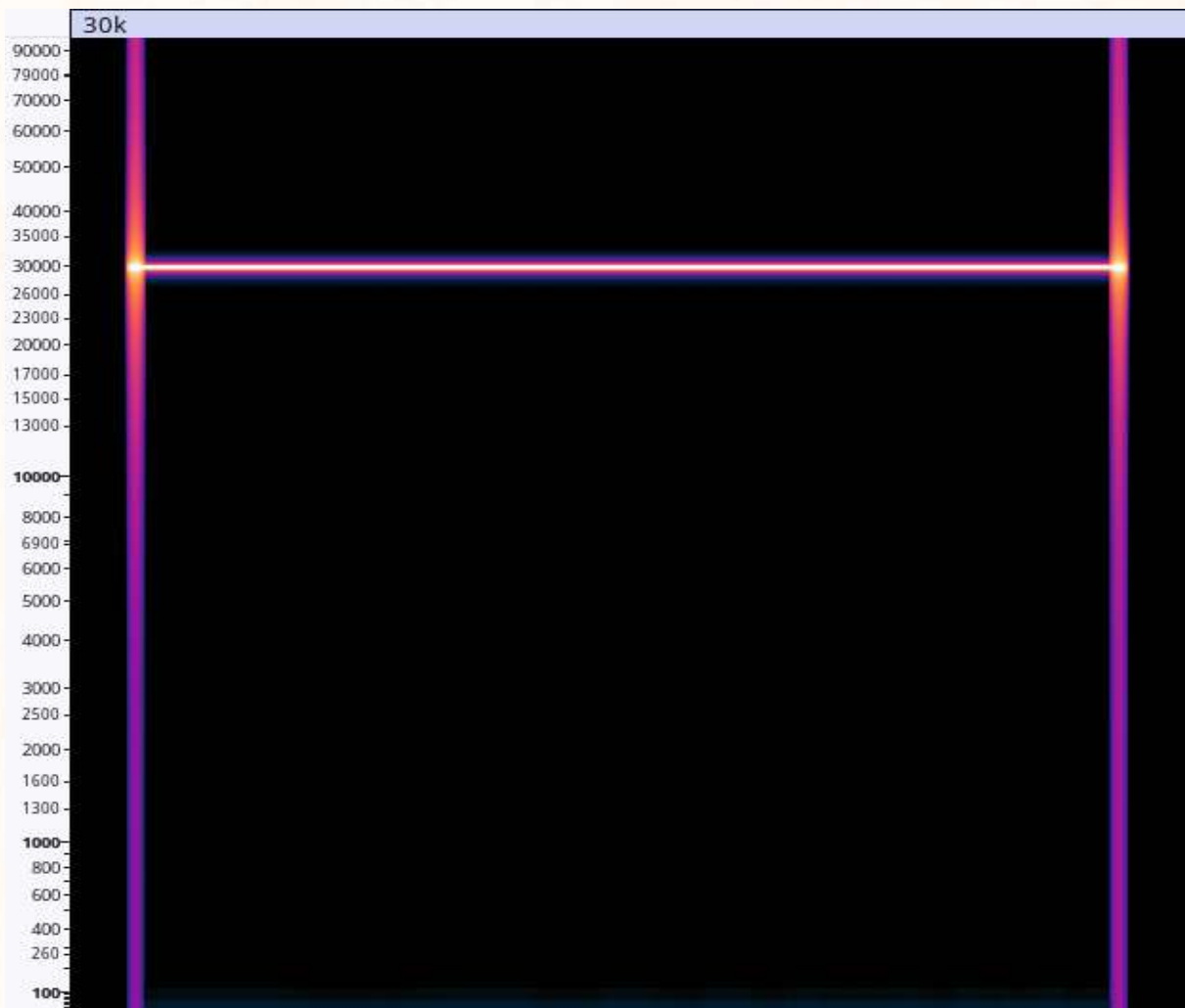


Fonte: Elaboração do próprio autor.

Deve-se atentar a configuração de amostragem de todos os programas utilizados, para que estejam acima do limite de Nyquist para o ultrassom em questão, algo que diversos sistemas de reprodução e gravação cuja amostragem abrange o registro ultrassônico, tido por alguns como relevante na qualidade geral do áudio (Melchior, 2019, p. 247) permitem configurar. Neste caso configurado para 96 kHz, maior ou igual ao dobro da frequência em questão sendo um uso prático do “áudio com maior resolução” descrito por Melchior.

Podemos então observar no espectrograma que temos a confirmação de um fonograma ultrassônico:

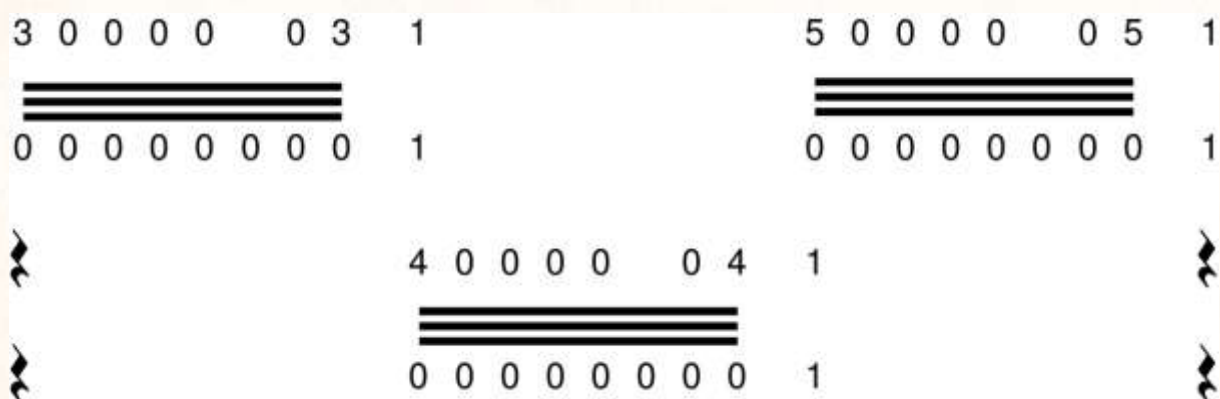
Figura 2 – Espectrograma de um tom em 30 000 Hz.



Fonte: Elaboração do próprio autor.

Ao dobrar o numero de vozes, podemos intercalar melodias, emulando um legato.

Figura 3 – Melodia ultrassônica legato.



Fonte: Elaboração do próprio autor.

Parametrização de elementos não sonoros

Por estarmos trabalhando com um conjunto de softwares de código aberto, pode-se estender a funcionalidade da notação para outros domínios, sem muita fricção.

Diversas aplicações podem ser previstas:

- Robótica
- Computação Gráfica
- Impressão 3D/CNC
- Automação

Postula-se que em geral, processos podendo ser descritos como um conjunto finito de parâmetros discretizáveis, com estado inicial, final, e uma sequencia temporal de inter estados poderiam ser perfeitamente descritos pela notação apresentada, dado uma quantidade suficiente de vozes, compassos e precisão rítmica e numérica.

Pluripotência necessita extensiva formulação formal, porem, o esboço geral do escopo da notação é suficiente, visto que além da simples notação numérica, podemos associar símbolos arbitrários a cada numero inteiro, sendo tais símbolos outros números, ícones, letras ou palavras, etc.

Com isso, mantendo a notação basal, baseada apenas em números inteiros, podemos expandir para domínios arbitrários, mas obter uma notação unificada, com um único rudimento, a notação rítmica de números inteiros em vozes paralelas.

Conclusões

Passeamos por uma quantidade de objetos distintos, mas relacionados. Uma coletânea de diversos sistemas discretos, com objetivos próprios. O Muscore não foi feito para notacionar ultrassons. O Pure data não é cliente de rede, embora tenha a capacidade. E o Godot foi feito para criar jogos, nativamente. Porem, todos são programas de criação, fundamentalmente. Sua união resultou em uma sinergia entouragica, que outros conjuntos de programas muito mais complexidade exigiria.

A natureza de código aberto não é um fator central, mas fundamental para a expansão e validação da notação apresentada. De imediato, é possível reimplementar os rudimentos apresentados, visto que as competências de cada software são bem delimitadas, e

também sendo possível derivar uma suíte integrada, em uma única implementação monolítica, quando necessário.

Quanto ao sistema notacional implementado, e suas aplicações e potenciais aplicações, deve-se continuar instigando novas vertentes que poderiam ser abarcadas. Certamente diversas funcionalidades podem ser expandidas baseando-se na sua natureza numérica, e, com um rigoroso formalismo matemático posteriormente, todo um campo universal de aplicações pode ser validado.

Das aplicações já implementadas, tal como o fonograma ultrassônico com a entrada arbitrária de frequências, e também a notação da fração microtonal, já obtemos um vislumbre prático das capacidades de tal abordagem.

Referências

HENRY, C. Physical modeling for pure data (pmpd) and Real time interaction with an audio Synthesis. *In: Journées d'informatique musicale*. Paris, França, 2004

MELCHIOR, V. R. High Resolution Áudio: A History and Perspective. *In: Journal of the Áudio Engineering Society*. Estados Unidos, 2019, v. 67, n. 5, p. 246-257.

MORENO, M. Analysis and resynthesis of real instruments using Pure Data and the MKMR library. *In: Revista Vórtex*. Curitiba, v.9, n.2, p. 1-12, 2021

YAMASHITA, A. *et. al.* Assessment of ability to discriminate frequency of bone-conducted ultrasound by mismatch fields. *In: Neuroscience Letters*. Países Baixos, 2008, v. 438, p. 260–262.

YEE, E; TSUEI, Y. Transient Response by Component Modal Synthesis Method. *In: Journal of Vibration and Acoustics*. Estados Unidos, Vol. 112/21, 1990