

Método de rastreamento de subespaços dominantes: aplicação em sinais de áudio

Lício H. Bezerra

Saulo Castilho

Depto de Matemática, UFSC

88040-900, Florianópolis, SC

E-mail: licio@mtm.ufsc.br, saulo@mtm.ufsc.br

RESUMO

Neste trabalho, vamos assumir que um sinal musical $x(t)$ é expresso da seguinte forma:

$$x(t) = \left[\sum_{i=1}^{K(t)} a_i(t) \cos(\theta_i(t)) \right] + n(t),$$

em que $\theta_i(t) = \theta_{i0} + 2\pi \int_0^t f_i(s) ds$.

Consideremos amostras de um som digitalizado, que são do tipo $h_i = x((i-1)\Delta t)$, $i = 1, 2, \dots, N$, Δt fixo. Queremos analisá-las e ressampleá-las. Isto é, queremos descobrir $K(t)$ e os outros parâmetros de cada parcial acima para, a partir disso, recriar a função x .

Se o som não for complexo, por exemplo, uma melodia reproduzida por um instrumento, trechos bem pequenos do som, os quais chamamos de janelas, podem ser analisados matematicamente a partir do modelo que chamamos de senoidal com amortecimento exponencial (EDS). Nesse modelo, $K(t) \equiv k$, em que k é uma constante; para todo i , f_i é uma constante; para todo i , $a_i(t) = A_i e^{-d_i t}$, com $d_i > 0$ e, assim, o sinal pode ser reescrito na forma exponencial

$$x(t) = \sum_{i=1}^k [\alpha_i e^{-d_i t + i 2\pi f_i t} + \bar{\alpha}_i e^{-d_i t - i 2\pi f_i t}],$$

em que $\alpha_i = A_i e^{i\theta_{i0}}$. Dada uma janela de tamanho L , esses parâmetros podem ser calculados por técnicas de Álgebra Linear Computacional (ver [3], [4]), que utilizam a matriz de Hankel H , cuja primeira coluna é $[h_1 \dots h_L]$ e cuja última linha é $[h_L \dots h_{2L-1}]$. Um fato fundamental para a ressample é o cálculo do subespaço associado aos $2k$ autovalores dominantes de H . É esse espaço que queremos rastrear.

Para isso, utilizaremos o método de janelas deslizantes, calculando Q_i para cada matriz de Hankel H_i formada com os sinais da janela i . Vamos usar um método alternativo ao QR, que ortonormaliza as colunas de $C_i Q_{i-1}$ ($C_i = H_i^2$), computando

$$Q_i = C_i Q_{i-1} (Q_{i-1}^T C_i^2 Q_{i-1})^{-1/2}$$

A intenção dessa escolha é diminuir a ordem de complexidade do cálculo de Q_i . Essa computação será feita por dois métodos aproximativos, cuja ordem de complexidade é $\mathcal{O}(2kL)$: o primeiro é o algoritmo NP3, descrito em [5]; o segundo é o algoritmo OPAST ([1]). Ambos baseiam-se na aproximação da projeção $C_i Q_i \approx C_i Q_{i-1}$ que é válida quando Q_i varia suavemente em relação a i .

Uma vez computada Q_i , estimamos todos os parâmetros necessários para ressample h_i , isso para cada i , que varia de 1 a $N - 2L$.

Apresentamos, no final do trabalho, resultados de experimentos feitos com um som digitalizado de piano, com a nota A4 (lá médio) sendo tocada. É feita a comparação da performance dos dois métodos na ressample desse som, tanto em relação à precisão como em relação ao tempo de computação. Através desses resultados, vemos que o esforço computacional para ressample um som simples com fidelidade é bastante grande.

Finalizamos o trabalho com observações sobre a ressample de sons polifônicos, à qual esses dois métodos de alta resolução não se aplicam, devido às limitações das arquiteturas atuais dos computadores.

Toda a programação foi feita em MATLAB e executada em computadores pessoais.

1 Introdução

A representação matemática de sons gerados por instrumentos musicais não é única. Neste trabalho, vamos assumir que um sinal musical $x(t)$ é expresso da seguinte forma:

$$x(t) = \left[\sum_{i=1}^{K(t)} a_i(t) \cos(\theta_i(t)) \right] + n(t),$$

em que

$$\theta_i(t) = \theta_{i_0} + 2\pi \int_0^t f_i(s) ds.$$

Nessas fórmulas, os parâmetros são os seguintes:

$a_i(t)$ é a amplitude da i -ésima senóide (ou i -ésima parcial) no tempo t ;

$K(t)$ é o número de parciais, que podem variar com o tempo, por exemplo, se o som for produzido por vários instrumentos, simultaneamente;

$\theta_i(t)$ é a fase da i -ésima parcial no tempo t ;

$f_i(t)$ é a frequência da i -ésima parcial no tempo t ;

θ_{i_0} é a fase inicial da i -ésima parcial (no tempo $t = 0$);

$n(t)$ é um ruído.

Suporemos, ainda, que x é uma função quase periódica em relação ao tempo, no sentido que sua forma ondulatória é periódica do ponto de vista microscópico. Queremos dizer com isso que a função é quase idêntica em períodos de tempo adjacentes, mas que pode ser bem diferente quando considerada em períodos bem distantes. Ou seja, matematicamente falando, $K(t)$ é praticamente uma constante e, para todo i , $a_i(t)$ é quase uma constante, assim como $f_i(t)$.

O problema para o qual buscamos alguma solução é o seguinte: consideremos amostras de um som digitalizado, que são do tipo $h_i = x((i-1)\Delta t)$, $i = 1, 2, \dots, N$, Δt fixo. Queremos analisá-las e ressampleá-las. Isto é, queremos descobrir $K(t)$ e os outros parâmetros descritos acima de cada parcial para, a partir disso, recriar a função x .

Se o som não for complexo, por exemplo, uma melodia reproduzida por um instrumento, trechos bem pequenos do som, os quais chamamos de janelas, podem ser analisados matematicamente a partir do modelo que chamamos de senoidal com amortecimento exponencial (EDS). Nesse modelo, $K(t) \equiv k$, em que k

é uma constante; para todo i , f_i é uma constante; para todo i , $a_i(t) = A_i e^{-d_i t}$, com $d_i > 0$ e, assim, o sinal pode ser reescrito na forma exponencial

$$x(t) = \sum_{i=1}^k [\alpha_i e^{-d_i t + i 2\pi f_i t} + \bar{\alpha}_i e^{-d_i t - i 2\pi f_i t}],$$

em que $\alpha_i = A_i e^{i\theta_{i_0}}$. Repare que, por esse modelo, os sinais são somas de parciais transitentes, pois, para todo i , $d_i > 0$. Se o tamanho L da janela não for muito grande (significando que o computador pode operar com matrizes de ordem $L/2$), esses parâmetros podem ser calculados por técnicas de Álgebra Linear Computacional (ver [3], [4], [6]). Essas técnicas utilizam a matriz de Hankel H , cuja primeira coluna é $[h_1 \dots h_L]$ e cuja última linha é $[h_1 \dots h_{2L-1}]$.

É fácil mostrar que $H = VAV^T$, em que V é a matriz de Vandermonde, cujos polos são $(-d_1 + i 2\pi f_1)\Delta t, (-d_1 - i 2\pi f_1)\Delta t, \dots, (-d_k + i 2\pi f_k)\Delta t, (-d_k - i 2\pi f_k)\Delta t$ e A é uma matriz diagonal, em que $A(1, 1) = \alpha_1, A(2, 2) = \bar{\alpha}_1$ etc (ver [4], por exemplo). Ou seja, o posto de H é $2k$. Obviamente, o tamanho da janela deve ser tal que L seja maior que $2k$.

A matriz H é uma matriz simétrica e, por isso, as colunas da matriz de Vandermonde V são geradas pelos autovetores de H associados aos $2k$ maiores autovalores, em valor absoluto, de H . Bem, se o posto de H é $2k$, isso significaria que as colunas de V seriam geradas pelos autovetores associados aos autovalores não nulos.

Há dois problemas no transporte dessa argumentação teórica para o caso prático: um é que o som não é livre de ruídos, ou seja, o modelo funciona bem para sons com pouco ruído; outra é que os métodos numéricos de decomposição espectral não são livres de ruídos da aritmética de ponto flutuante do computador. Conclusão: é difícil saber quais autovalores computados são perturbações do autovalor zero. Assim, a estimativa do parâmetro $2k$ é essencial: se for pequena, o som ressampleado será uma aproximação pobre do som original; se for grande, pode trazer ruídos numéricos que se traduzem em chiados no som sintetizado.

Um fato fundamental para a ressampleagem é, então, o cálculo desse subespaço associado aos $2k$ autovalores dominantes. É esse espaço que queremos rastrear. Com ele, podemos construir

inúmeras matrizes cujos espectros contêm os polos de V como autovalores. Essas matrizes são chamadas de espectrais. Uma vez calculados os autovalores, resolve-se, por mínimos quadrados, o seguinte problema: $V\aleph = He_1$. Obtém-se, então, o vetor \aleph , que é formado pelas amplitudes α_i (vamos chamá-las assim). Finalmente, então, podemos recriar x , pelo menos, no tempo i daquela janela.

Quando o tamanho do sinal N é pequeno, uma janela basta para resintetizar o som total. Mas, quando N é grande, utilizamos o método de janelas deslizantes (*sliding windows*) (ver [8]). Esse método utiliza a sequência de matrizes de Hankel H_1, H_2 etc, que percorre todo o sinal (ver Figura 1, onde H_2 é a matriz entre barras duplas). Nesse caso, para cada janela i , reconstrói-se o sinal h_i como sendo a soma das amplitudes ($\Delta t = 0$). Se quisermos alterar o sinal, podemos facilmente operar com as amplitudes, mas é um pouco mais complicado se quisermos modificar, controladamente, as frequências de uma forma não linear (ver [3]).

h_1	h_2	\cdots	h_L	h_{L+1}	\cdots	h_{N-L-1}
h_2	h_3	\cdots	h_{L+1}	h_{L+2}	\cdots	h_{N-L}
\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots
h_L	h_{L+1}	\cdots	h_{2L-1}	h_{2L}	\cdots	h_N

Figura 1: Janelas deslizantes

A idéia básica dos métodos de janelas deslizantes fica clara quando se olha a versão adaptativa do método de iteração ortogonal, a partir de uma matriz real Q_0 com colunas ortonormais, $L \times 2k$:

$$Y_i = C_i Q_{i-1}$$

$$Q_i R_i = Y_i \text{ (decomposição QR de } Y_i)$$

O método costuma convergir em poucas iterações. Os sinais de música, em geral, começam com dados nulos, relativos ao silêncio. Se não, é bom fazer um preenchimento com alguns zeros no início (equivalente ao processo de *Zero Padding* de Análise de Fourier), para que os métodos se auto-orientem.

Há várias tentativas de aproximar os cálculos tanto da multiplicação $C_i Q_{i-1}$ como da decomposição QR econômica de Y_i , com vistas a

uma diminuição da complexidade computacional desse algoritmo, sem comprometer a convergência do processo (ver [8]).

Observação: referimo-nos à decomposição QR econômica de uma matriz A como sendo um processo de ortonormalização das suas colunas (numericamente, em geral, é realizada via reflexões de Householder ou rotações de Givens), em contraste com a decomposição QR, que calcula uma matriz ortogonal Q , completando-se a base ortonormal do espaço-coluna de A que foi calculada.

Aqui, usaremos um modo alternativo ao QR para ortonormalizar as colunas de $C_i Q_{i-1}$, que é computando a equação

$$Q_i = C_i Q_{i-1} (Q_{i-1}^T C_i^2 Q_{i-1})^{-1/2}. \quad (1)$$

Note que $Q_i^T Q_i = I$.

Vamos computar essa expressão por dois modos aproximativos: o primeiro é o algoritmo NP3, descrito em [5]; o segundo é o algoritmo OPAST ([1]). Ambos baseiam-se na aproximação da projeção $C_i Q_i \approx C_i Q_{i-1}$ que é válida quando Q_i varia suavemente em relação a i . Esses algoritmos surgiram como desenvolvimento da abordagem pioneira de Yang ([9]) ao problema de estimação de subespaços dominantes. Vamos utilizar, de forma inédita na literatura, as matrizes C_i como sendo H_i^2 , em que H_1, H_2, \dots é uma seqüência de matrizes de Hankel dispostas em janelas deslizantes (ver Figura 1). Notemos que o subespaço a ser rastreado é o mesmo, quando consideramos H_i^2 em vez de H_i . Esses dois algoritmos foram readaptados ao nosso caso, pois eles foram criados originalmente para matrizes de covariância C_i , cujas atualizações são de posto 1. No nosso caso, as perturbações de C_i são de posto 2. A descrição dos dois algoritmos é feita na seção a seguir.

2 Algoritmos NP3 e OPAST

Vamos supor que temos um sinal musical

$$h(1), h(2), \dots, h(N),$$

em que N é um número ímpar (se não, redefinimos $N := N - 1$). Seja L , $2k < L < (N + 1)/2$, o tamanho da janela na qual o sinal pode ser modelado como um sinal senoidal com amortecimento exponencial, isto é, como soma de $2k$ parciais complexas.

Para cada $i \geq 1$, seja $C_i = H_i^2$, em que $H_i = \text{hankel}(h_{i,1}, h_{i,2})$, $h_{i,1} = [h_{i \dots h_{i+L-1}}]$, $h_{i,2} = [h_{i+L-1 \dots h_{i+2L-2}}]$. Cálculos bem simples resultam na seguinte expressão:

$$C_{i+1} = C_i - h_{i,1}h_{i,1}^T + h_{i+1,2}h_{i+1,2}^T.$$

Vamos descrever, a seguir, os dois algoritmos de rastreamento de subespaços citados acima: NP3 e OPAST. Esses algoritmos computam subespaços dominantes das matrizes de Hankel em janelas deslizantes, que contêm as características do sinal quando ele evolue com o tempo. Partindo desses subespaços, utilizaremos uma estratégia de cálculo adaptativo da matriz espectral ([2]), que é formada segundo o método ESPRIT ([3], [7]). Esse cálculo é feito de modo exato, se a atualização dos subespaços for feita, também, de forma exata (o que não é o caso). Não vamos relatar aqui essa estratégia, que não é complexa e está bem descrita em [2]. Resultados de testes feitos com esses dois algoritmos serão descritos na próxima seção.

2.1 NP3

Seja Q_0 a matriz $L \times 2k$ tal que, para todo i , $Q_0(i, :) = e_i$, em que $Q_0(i, :)$ representa a i -ésima coluna de Q_0 e e_i , o i -ésimo vetor da base canônica de \mathbb{R}^L . Vamos calcular o lado direito da equação (1) em duas etapas. Vamos definir $Y_i = C_i Q_{i-1}$ e $Z_i = (Y_i^T Y_i)^{-1/2}$.

Primeiro, temos que

$$\begin{aligned} Y_i &= (C_{i-1} - h_{i-1,1}h_{i-1,1}^T + h_{i,2}h_{i,2}^T)Q_{i-1} = \\ &= C_{i-1}Q_{i-1} - h_{i-1,1}h_{i-1,1}^T Q_{i-1} + h_{i,2}h_{i,2}^T Q_{i-1}. \end{aligned}$$

Agora, supondo que $Q_{i-1} \approx Q_{i-2}$, temos então que

$$\begin{aligned} Y_i &\approx Y_{i-1} - h_{i-1,1}h_{i-1,1}^T Q_{i-1} + h_{i,2}h_{i,2}^T Q_{i-1} = \\ &= Y_{i-1} - r_i * s_i^T, \end{aligned}$$

em que r_i é a matriz, $L \times 2, [h_{i-1,1} \ h_{i,2}]$ e s_i é a matriz $Q_{i-1}^T * [h_{i-1,1} \ -h_{i,2}]$. Segundo, vemos que $Y_i^T Y_i$ é igual à matriz

$$Z_{i-1}^{-1}[I - a_i b_i^T - b_i a_i^T + a_i c_i a_i^T]Z_{i-1}^{-1},$$

em que $a_i = Z_{i-1} s_i$, $b_i = Z_{i-1} Y_{i-1}^T r_i$ e $c_i = r_i^T r_i$. Calculando a decomposição espectral de $[I - a_i b_i^T - b_i a_i^T + a_i c_i a_i^T]$, que é uma matriz

simétrica definida positiva gerada por uma perturbação de posto 4 da matriz identidade, a expressão acima fica igual a

$$Z_{i-1}^{-1}[I + \lambda_1 v_1 v_1^T + \dots + \lambda_4 v_4 v_4^T]Z_{i-1}^{-1},$$

em que v_1, v_2, v_3, v_4 são autovetores ortonormais associados, respectivamente, a $\lambda_1, \dots, \lambda_4$.

Logo, $Z_i \approx [I - \gamma_1 v_1 v_1^T - \dots - \gamma_4 v_4 v_4^T]Z_{i-1}$, em que $I - \gamma_1 v_1 v_1^T - \dots - \gamma_4 v_4 v_4^T = [I + \lambda_1 v_1 v_1^T + \dots + \lambda_4 v_4 v_4^T]^{-1/2}$, na qual, para todo i , $\gamma_i = 1 - 1/\sqrt{1 + \lambda_i}$.

Observação: se P é uma matriz simétrica definida positiva, $P = AA^T$, em que A é inversível. Então, $P^{-1/2} = A^{-T}U = U^T A^{-1}$, em que U é uma matriz ortogonal. Chamamos essas matrizes de **raízes assimétricas** à esquerda e à direita, respectivamente, de P . Em particular, se $P = ABA^T$, então $P^{-1/2} = A^{-T}B^{-1/2}U = U^T B^{-1/2}A^{-1}$. Dessa forma, estamos aproximando Z_i por uma raiz assimétrica de $Y_i^T Y_i$.

Sejam $q_i = [v_1 \dots v_4]$ e $p_i = [\lambda_1 v_1 \dots \lambda_4 v_4]$. Finalmente, podemos calcular Q_i , que será definido como $Y_i Z_i^T$. Substituindo Y_i e Z_i na expressão de Q_i , temos então que

$$Q_i = Q_{i-1} + E_i G_i^T,$$

em que E_i é a matriz $[W_i \ r_i]$, com $W_i = r_i * a^T * p_i - Q_{i-1} * p_i$ e G_i , a matriz $[q_i \ a_i]$.

2.2 OPAST

Esse algoritmo é fundado no fato que se

$$W_i = C_i Q_{i-1} (Q_{i-1}^T C_i Q_{i-1})^{-1},$$

então

$$Q_i = W_i (W_i^T W_i)^{-1/2}$$

é uma matriz cujas colunas são vetores ortonormais que geram $C_i Q_{i-1}$. Note que $(W_i^T W_i) =$

$$(Q_{i-1}^T C_i Q_{i-1})^{-1} (Q_{i-1}^T C_i^2 Q_{i-1}) (Q_{i-1}^T C_i Q_{i-1})^{-1}$$

Logo, $(W_i^T W_i)^{-1/2} =$

$$(Q_{i-1}^T C_i Q_{i-1}) (Q_{i-1}^T C_i^2 Q_{i-1})^{-1/2} U,$$

em que U é uma matriz ortogonal. Além disso, vemos que

$$W_i (W_i^T W_i)^{-1/2} = C_i Q_{i-1} (Q_{i-1}^T C_i^2 Q_{i-1})^{-1/2} U,$$

ou seja, a definição de Q_i em OPAST é igual a do NP3, pós-multiplicada por uma matriz ortogonal.

O desenvolvimento do OPAST é o seguinte:

1. $Y_i = C_i Q_{i-1} \approx Y_{i-1} - r_i s_i^T$. Isto é, continua como acima, com as mesmas definições de r_i e s_i ;
2. Para calcular $Z_i = (Q_{i-1}^T C_i Q_{i-1})^{-1}$, que é igual a $(Q_{i-1}^T Y_i)^{-1}$, utilizamos a fórmula de Sherman-Morrison para inversão de perturbações de matrizes inversíveis:

$$(A + uv^T)^{-1} = A^{-1} - A^{-1}u(I + v^T A^{-1}u)^{-1}v^T A^{-1}.$$

O resultado foi o seguinte:

$$Z_i \approx Z_{i-1} + q_i \gamma_i t_i^T,$$

em que $q_i = Z_{i-1} Q_{i-1}^T r_i$, $t_i = Z_{i-1} s_i$ e $\gamma_i = (I - t_i Q_{i-1}^T r_i)^{-1}$. Concluimos, então, que

$$W_i = W_{i-1} - p_i t_i^T,$$

em que $p_i = (r_i - Q_{i-1} Q_{i-1}^T r_i) \gamma_i$.

3. Substituindo as matrizes encontradas na expressão $Q_i = W_i (W_i^T W_i)^{-1/2}$, obtemos que

$$Q_i = Q_{i-1} - \hat{p}_i n_i^T,$$

em que $n_i m_i$ é a fatoração QR econômica de t_i e

$$\hat{p}_i = p_i m_i (I - F_i) + Q_{i-1} n_i F_i.$$

Nessa expressão,

$$F_i = f_i [I - (I + d_i)^{-1/2}] f_i^T,$$

em que $f_i d_i f_i^T$ é a decomposição espectral da matriz, 2×2 , $m_i (p_i^T p_i) m_i^T$.

3 Testes e comparações

Descrevemos, na seção anterior, dois algoritmos adaptativos, NP3 e OPAST, que computam uma matriz Q_i de colunas ortonormais, as quais geram $C_i Q_{i-1}$, a partir de perturbações de ordem 4 e 2, respectivamente. As complexidades computacionais desses dois algoritmos são ambas da ordem $\mathcal{O}(2kL)$, a cada passo (ver [1] e [5]).

Toda a programação dos métodos foi feita em MATLAB e os programas foram rodados em computadores pessoais comuns, mas de última geração. Vamos apresentar a seguir alguns resultados de testes realizados, que foram obtidos tomando-se um sinal que representa o som emitido pelo tocar da tecla A4 de um piano (a nota lá, no meio do piano). Esse e outros arquivos de áudio podem ser baixados em

<http://amath.colorado.edu/pub/matlab/music>

Esse arquivo *WAVE* contém ($N =$) 24931 amostras, tomadas a cada $1/22050$ s ($= \Delta t$). O gráfico de h em função do tempo pode ser visto na Figura 2.

Observação: nas figuras, o número de picos representa as $2k$ frequências que foram tomadas para descrever o som.

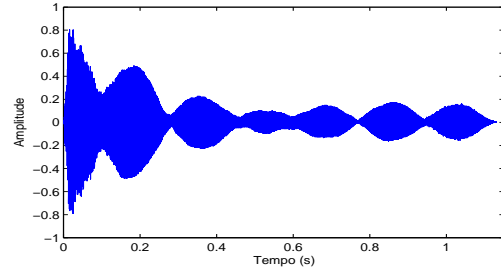


Figura 2: o som original

Com $L = 300$ e $k = 50$ ($2k = 50$), o sinal sintetizado por NP3 e por OPAST são mostrados nas Figuras 3 e 4, respectivamente.

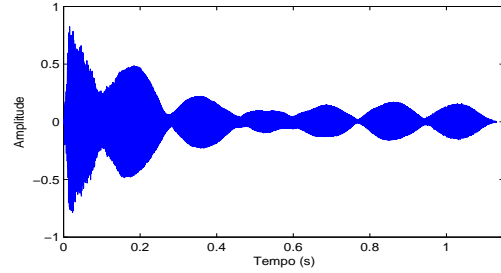


Figura 3: o som resintetizado por NP3

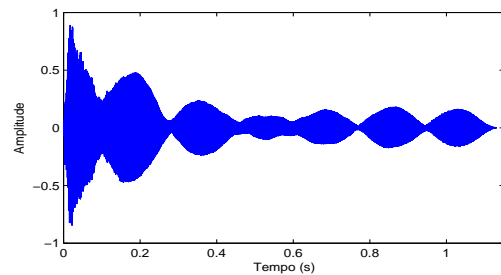


Figura 4: o som resintetizado por OPAST

Aparentemente, os sinais são quase idênticos. Porém, se tomarmos uma janela do início do processo, os sinais resintetizados estão ainda distante do original, pois o método não convergiu ainda, conforme a Figura 5.

Se tomarmos uma janela bem mais adiante, por exemplo entre 12000 e 12300, a diferença

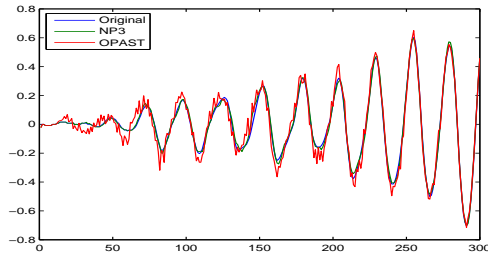


Figura 5: janela $0 \leq i \leq 300$

entre o som original e os resintetizados pelos dois algoritmos é desprezível, de acordo com o que se vê na Figura 6. Isso acontece porque o método já está convergindo.

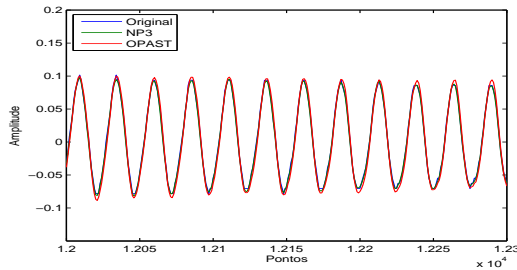


Figura 6: janela $12000 \leq i \leq 12300$

Na Figura 7, podemos ver os erros relativos, na norma infinito, dos sons sintetizados pelos dois algoritmos em relação ao som original. Quando k aumenta, o NP3 tem um desempenho ruim. Verificamos que isso foi causado pelo fato das matrizes Q_i perderem a ortogonalidade das suas colunas, quando k é grande em relação a L . A atualização dessas matrizes dependem dos vetores v_1, \dots, v_4 . Para calcular esses vetores, não utilizamos a decomposição espectral de $A = a_i b_i^T - b_i a_i^T + a_i c_i a_i^T$, que é uma matriz $2k \times 2k$. Não teria sentido calcular 250 autovetores, no caso de $2k = 250$, e tomarmos apenas 4 deles. Ao invés disso, utilizamos a decomposição espectral de uma matriz 4×4 , cujos autovalores são os mesmos de A . Em [5], eles utilizam uma matriz diferente da que usamos neste trabalho. Como os testes mostraram que a computação de Q_i era sensível ao cálculo numérico de q_i , nós refinamos o seu cálculo, para que gerasse o mínimo de erro possível. Para isso, buscamos outra matriz de ordem 4 e uma outra estratégia de construção de q_i . Ela funcionou bem para k pequeno em relação a L . Pode-se estabilizar mais esse cálculo, por exemplo, tomando a decom-

posição QR econômica com permutação, que é uma função do MATLAB. Porém, o custo computacional será maior.

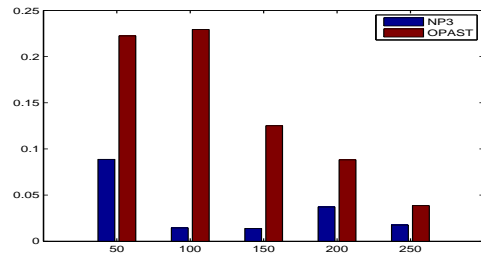


Figura 7: Erro relativo para vários valores de $2k$

4 Conclusões

Os sons monofônicos gerados por instrumentos harmônicos, como o piano e o violão, quando digitalizados, são representados por sinais cujos parâmetros podem ser estimados, eficientemente, por técnicas de Álgebra Linear Computacional, uma vez aplicados métodos de rastreamento de subespaços dominantes. Implementamos e testamos dois algoritmos que pertencem à classe de algoritmos ditos de alta resolução. Esses algoritmos ainda têm um custo computacional alto para as arquiteturas de computador existentes atualmente, o que pode ser observado na Figura 8.

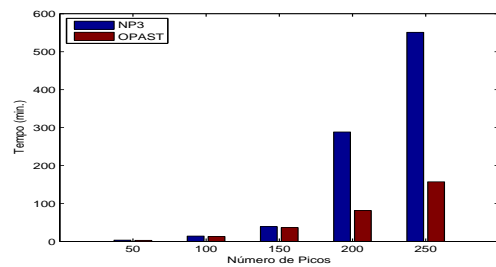


Figura 8: Tempo de processamento para valores de $2k$

Observemos que, a cada passo i , depois de achar Q_i , calculamos autovalores de uma matriz espectral $2k \times 2k$ e resolvemos um problema de mínimos quadrados para então resintetizarmos o valor de h_i . Estamos criando algoritmos que calculem h em blocos, com janelas deslizando em passos maiores. Porém, nos resultados obtidos ainda persistem ruídos relativos à descontinuidade das matrizes definidas a partir

desses passos. Estamos, também, implementando outros algoritmos adaptativos de alta resolução, cujos resultados estão sendo avaliados para realimentar o algoritmo.

Uma pergunta natural é sobre a convergência desses métodos. Os autores do OPAST e do NP3 dão uma demonstração da sua convergência global, no caso de C_i ser constante, para todo i , mais algumas outras poucas suposições ([1], [5]). Os argumentos utilizados baseiam-se na convergência de métodos de potência para cálculo de autovalores, em aritmética de ponto flutuante. Ainda não conseguimos uma prova de convergência local em aritmética exata desses métodos, mas estamos investigando isso.

Para sons polifônicos, ou para arquivos grandes de áudio, essas técnicas são pouco eficazes, no que diz respeito ao tempo. Para esses sons, há necessidade de se aliarem conceitos de psicoacústica à Matemática, para eliminar frequências que o ouvido humano pouco percebe (aliás, existem dados que indicam que o ouvido do ser humano está, cada dia que passa, menos eficiente na discriminação de sons). Na intenção de ressintetizar sons polifônicos, os autores deste trabalho implementaram, também, alguns algoritmos espectrais, baseados em Análise de Fourier, que mostraram um desempenho surpreendente na síntese de sons de música popular com instrumentos eletrônicos. Porém, quando aplicados a arquivos simples como o piano_A4.wav, não funcionaram muito bem. Os métodos espectrais são rápidos, funcionam bem quando não se exige fidelidade. Por outro lado, os métodos de alta resolução representam o som de um modo fiel ao original, mas exige grande capacidade de processamento.

Referências

- [1] K. Abed-Meraim, A. Chkeif and Y. Hua, Fast Orthogonal PAST Algorithm, *IEEE Signal Processing Letters*, 7 (2000) 60-62.
- [2] R. Badeau, G. Richard and B. David, Fast Adaptive Esprit Algorithm, *2005 IEEE 13th Workshop on Statistical Signal Processing, SSP'05*, 289-294.
- [3] R. Badeau, R. Boyer and B. David, EDS Parametric Modelling and Tracking of Audio Signals, *Proceedings of the 5th International Conference on Digital Audio Effects, DAFX-02*, 139-144.
- [4] L. H. Bezerra and F. S. V. Bazán, Eigenvalue Locations of Generalized Companion Predictor Matrices, *SIAM Journal on Matrix Analysis and Applications*, 19 (1998) 886-897.
- [5] Y. Hua, Y. Xiang, T. Chen, K. Abed-Meraim and Y. Miao, A New Look at the Power Method for Fast Subspace Tracking, *Digital Signal Processing*, 9 (1999) 297-314.
- [6] J. Laroche, The use of the matrix pencil method for the spectrum analysis of musical signals, *J. Acoust. Soc. Am*, 94 (1993) 1958-1965.
- [7] R. Roy and T. Kailath, ESPRIT - estimation of signal parameters via rotational invariance techniques, *IEEE Trans. on Acoustics, Speech, and Signal Processing*, 37 (1989) 984-995.
- [8] P. Strobach, Square hankel SVD subspace tracking algorithms, *Signal Processing*, 57 (1997) 1-18.
- [9] B. Yang, Projection Approximation Subspace Tracking, *IEEE Trans. on Signal Processing*, 43 (1995) 95-107.