

# **O MPM como Ferramenta para o Processamento de Dados Clínicos de Ressonância Magnética**

**Uma Avaliação da Viabilidade**

**João Victor Dell Agli Floriano**

**Fernando Fernandes Paiva**

**Curso:** Mestrado

**Período a que se refere:** 02/2025 a 05/2025

**Bolsa de Estudos:** CAPES

**Período de Vigência:** 01/08/2024 a 28/02/2026 (19 meses)

# 1 Resumo

O seguinte trabalho tem como objetivo avaliar a aplicabilidade do algoritmo Matrix Pencil Method (MPM) para processamento de dados clínicos de MRS. A partir de uma simulação implementada in loco baseada em parâmetros empíricos de metabólitos presentes no cérebro, foram gerados sinais sintéticos de MRS, os quais foram corrompidos de maneira artificial com ruído gaussiano de valores variados. Esses sinais corrompidos foram então pós-processados por meio do algoritmo do MPM, implementado com sucesso também de maneira local, baseando-se no trabalho de Yingbo Hua e Tapan Sakar [2], permitindo a recuperação dos parâmetros originais do sinal, o conjunto de amplitudes  $S_0$ , as frequências  $\omega$ , as fases  $\phi$ , e os tempos de decaimento  $T_2$ . O comportamento desses parâmetros foi analisado para diferentes SNR, revelando um papel importante da amplitude  $S_0$  na composição do sinal. Foi também estudado o comportamento de um pico único sob efeito de ruído, revelando um comportamento exponencial decrescente na quantidade de picos de valor relevante conforme o SNR crescia, com prevalência do pico único como maior valor dentre todos. Espera-se elucidar melhor o comportamento de sinais clínicos sob a ótica do MPM, com perspectiva de encontrar caminhos para a melhora na qualidade de sua análise.

## 2 Introdução

### 2.1 Lápis de Matrizes

O termo "lápis" (do inglês *pencil*), cunhado por Grantmacher [?] no contexto de *pencil* de matrizes, é uma ferramenta que combina matrizes quadradas linearmente a partir de um parâmetro  $\lambda$ , como descrito pela Equação 1.  $l$  é um inteiro não-negativo.

$$L(\lambda) = \sum_{i=0}^l \lambda^i A_i \quad (1)$$

No contexto de *pencil* de funções,  $f(t, \lambda)$  é *pencil* de  $g(t)$  e  $h(t)$  quando segue o formato descrito pela Equação 2.

$$f(t, \lambda) = g(t) + \lambda h(t) \quad (2)$$

O método "lápis de matrizes", do inglês *Matrix Pencil Method* (MPM) é uma técnica numérica de estimativa de parâmetros de sinais, desenvolvido originalmente por Yingbo

Hua e Tapan Sakar [3] como uma alternativa a métodos já existentes como o de Prony [1]. O mesmo consiste em modelar os sinais como uma soma de exponenciais complexas amortecidas, como na Equação 3. Partindo dessa ideia, é então aplicada uma série de etapas, que inclui a utilização de outros métodos, como Decomposição em Valores Singulares (SVD, do inglês *Singular Value Decomposition*), para estimar os parâmetros dessa função modeladora.

$$y(n) = \sum_{k=1}^M R_k e^{i(\omega_k t + \phi_k) + \alpha_k} \quad (3)$$

## 2.2 Decomposição em Valores Singulares

A Decomposição em Valores Singulares, ou SVD, é um método antigo de fatoração de matrizes, tendo sido desenvolvido no século XIX, no qual por volta de 1870 foi estabelecido para matrizes reais por Beltrami e Jordan [2]. Dada uma matriz  $M$  real, de tamanho  $n \times m$ , sua decomposição SVD terá o formato definido pela Equação 4.

$$M = U \Sigma V^T \quad (4)$$

A qual:

- $U$ : Matriz  $n \times n$  a qual suas colunas são os autovetores de  $MM^T$ , também chamados de *vetores singulares à esquerda*
- $V$ : Matriz  $m \times m$  a qual suas colunas os autovetores de  $M^T M$ , também chamados de *vetores singulares à direita*
- $\Sigma$ : Matriz  $n \times m$  a qual suas entradas diagonais são as raízes quadradas dos autovalores não nulos de  $M^T M$  ou  $MM^T$ , também chamadas de *valores singulares*.

Essas três submatrizes, por sua vez, possuem inúmeras interpretações, geralmente dependendo do contexto a qual a decomposição foi aplicada. Uma interpretação comum, no contexto o qual  $M$  é uma transformação linear, é que essas três submatrizes são uma fatoração da transformação original em três operações elementares: uma rotação no subespaço de  $V$ , um dimensionamento por  $\Sigma$ , e uma rotação no subespaço de  $U$ .

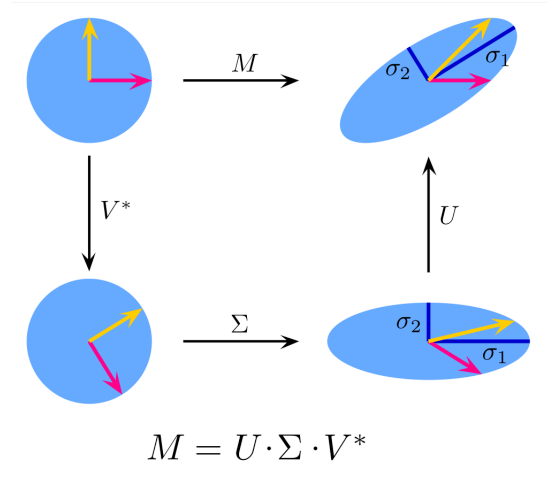


Figura 1: Sub-transformações de  $M$ .

Fonte: <https://commons.wikimedia.org/wiki/File:Singular-Value-Decomposition.svg>

No contexto deste trabalho, essa ferramenta é essencial para o algoritmo de MPM, sendo utilizado não apenas explicitamente no mesmo, como também de maneira implícita em outras etapas, como a do cálculo da matriz pseudo-inversa de Moore-Penrose, e no processo de mínimos quadrados.

## 3 Métodos

### 3.1 Implementação do MPM

A implementação do MPM em seu trabalho original é descrita originalmente de duas maneiras: a sem ruído, implementada de maneira mais simplificada; e a que leva em conta a presença de ruído, que utiliza algoritmos mais complexos, como a Decomposição em Valores Singulares (SVD, do inglês *Singular Value Decomposition*).

#### 3.1.1 Caso sem ruído

Assumindo um sinal  $x$  de tamanho  $N$ , para o caso sem ruído, define-se duas matrizes  $(N - L) \times L$ ,  $Y_1$  e  $Y_2$ , descritas pela Equação 5 e Equação 6.

$$Y_2 = \begin{bmatrix} x(1) & x(2) & \dots & x(L) \\ x(2) & x(3) & \dots & x(L+1) \\ \vdots & \vdots & & \vdots \\ x(N-L) & x(N-L+1) & \dots & x(N-1) \end{bmatrix} \quad (5)$$

$$Y_1 = \begin{bmatrix} x(0) & x(1) & \dots & x(L-1) \\ x(1) & x(2) & \dots & x(L) \\ \vdots & \vdots & & \vdots \\ x(N-L-1) & x(N-L) & \dots & x(N-2) \end{bmatrix} \quad (6)$$

Sendo  $L$  o parâmetro de *pencil*, importante para etapas posteriores. É possível escrever  $Y_1$  e  $Y_2$  como a Equação 7 e a Equação 8, sendo  $Z_1$  descrito pela Equação 9,  $Z_2$  descrito pela Equação 10,  $Z_0$  descrito pela Equação 11, e enfim  $R$ , descrito pela Equação 12. Por enquanto, o parâmetro  $M$  pode ser considerado como igual a  $N$ , porém virá a assumir valores diferentes posteriormente.

$$Y_2 = Z_1 R Z_0 Z_2 \quad (7)$$

$$Y_1 = Z_1 R Z_2 \quad (8)$$

$$Z_1 = \begin{bmatrix} 1 & 1 & \dots & 1 \\ z_1 & z_2 & \dots & z_M \\ \vdots & \vdots & & \vdots \\ z_1^{N-L-1} & z_2^{N-L-1} & \dots & z_M^{N-L-1} \end{bmatrix} \quad (9)$$

$$Z_2 = \begin{bmatrix} 1 & z_1 & \dots & z_1^{L-1} \\ 1 & z_2 & \dots & z_2^{L-1} \\ \vdots & \vdots & & \vdots \\ 1 & z_M & \dots & z_M^{L-1} \end{bmatrix} \quad (10)$$

$$Z_0 = \text{diag}(z_1, z_2, \dots, z_M) \quad (11)$$

$$R = \text{diag}(R_1, R_2, \dots, R_M) \quad (12)$$

Considerando agora o *pencil*, descrito pela Equação 13: Reescrevendo-o a partir da Equação 7 e da Equação 8, obtém-se a Equação 14, que, reorganizada, rende a Equação 15.

$$Y_2 - \lambda Y_1 \quad (13)$$

$$Y_2 - \lambda Y_1 = Z_1 R Z_0 Z_2 - \lambda Z_1 R Z_2 \quad (14)$$

$$Y_2 - \lambda Y_1 = Z_1 R (Z_0 - \lambda I) Z_2 \quad (15)$$

Escolhendo  $\lambda = z_i$  de maneira intencional, a matriz  $Z_0 - \lambda I$  é zero, fazendo com que o *pencil* seja igual a zero, transformando o problema em um problema de autovalores generalizados. Encontrando os autovalores generalizados do par  $\{Y_1, Y_2\}$ , encontra-se os polos  $z_i$ . É possível também encontrar  $z_i$  como resultado do problema de autovalores comuns da matriz  $Y_1^+ Y_2$ , como na Equação 16.  $Y_1^+$ , a matriz pseudoinversa de Moore-Penrose de  $Y_1$ , é definida pela Equação 17, no qual  $H$  denota o conjugado transposto.

$$Y_1^+ Y_2 - \lambda I = 0 \quad (16)$$

$$Y_1^+ = (Y_1^H Y_1)^{-1} Y_1^H \quad (17)$$

Encontrados os polos  $z_i$ , basta encontrar os resíduos  $R_i$  a partir de um problema de mínimos quadrados, descritos pela Equação 18.

$$\begin{bmatrix} y(0) \\ y(1) \\ \vdots \\ y(N-1) \end{bmatrix} = \begin{bmatrix} 1 & 1 & \dots & 1 \\ z_1 & z_2 & \dots & z_M \\ \vdots & \vdots & \dots & \vdots \\ z_1^{N-1} & z_2^{N-1} & \dots & z_M^{N-1} \end{bmatrix} \cdot \begin{bmatrix} R_1 \\ R_2 \\ \vdots \\ R_M \end{bmatrix} \quad (18)$$

### 3.1.2 Caso com ruído

No caso com ruído, é construída a matriz generalizada  $Y$ , descrita pela Equação 19, da qual é possível extrair  $Y_1$  e  $Y_2$  excluindo a última e a primeira coluna, respectivamente.

$$Y = \begin{bmatrix} y(0) & y(1) & y(2) & \dots & y(L) \\ y(1) & y(2) & y(3) & \dots & y(L+1) \\ \vdots & \vdots & \vdots & & \vdots \\ y(N-L-1) & y(N-L) & y(N-L+1) & \dots & y(N-1) \end{bmatrix} \quad (19)$$

Como citado anteriormente, o parâmetro  $L$  viria a ser importante para filtragem do ruído, sendo escolhido entre  $N/3$  e  $N/2$ , intervalo no qual a variância dos parâmetros  $z_i$ , por conta do ruído, apresentou-se como mínima [3].

Após a construção, aplica-se uma decomposição em Valores Singulares (SVD) nessa matriz.

$$Y = U\Sigma V^H \quad (20)$$

Encontradas as submatrizes, o número  $M$  de polos a serem calculados é escolhido, sendo eles os  $M$  valores singulares de  $\Sigma$  que mais contribuem para a matriz final. A sugestão de exclusão segue a Equação 21, sendo  $\sigma_c$  o valor singular analisado,  $\sigma_{max}$  o maior valor singular e  $p$  a ordem do número de algarismos significativos. A partir disso, é possível filtrar os autovalores, separando os mais significativos de ruído.

$$\frac{\sigma_c}{\sigma_{max}} > 10^{-p} \quad (21)$$

Encontrado o valor singular limite de filtragem, a matriz  $Y'$  é construída a partir de  $Y$  com menos valores singulares, e a partir dela,  $Y'_1$  e  $Y'_2$  são derivadas. Realizada essa etapa, basta calcular os autovalores generalizados do par  $\{Y'_2, Y'_1\}$ , ou o autovalor da matriz  $Y'^+_1 Y'_2$ , e os resíduos  $R_i$ , seguindo o procedimento descrito anteriormente, concluindo o processo do MPM. O sinal pode ser reconstruído com os polos  $z_i$  e resíduos  $R_i$  pela Equação 22.

$$y(kT_s) \approx \sum_{i=1}^M R_i z_i^k, \quad k = 0, \dots, N-1 \quad (22)$$

O algoritmo do MPM pode ser resumido, então, pelas seguintes etapas:

1. Calcular a matriz  $Y$ .
2. Calcular a SVD de  $Y$ .
3. Filtrar os valores singulares de acordo com o critério de seleção da Equação 21.
4. Reconstruir a matriz filtrada  $Y'$ .
5. Obter  $Y'_1$  e  $Y'_2$  a partir de  $Y'$ .
6. Calcular os autovalores generalizados do par  $\{Y'_2, Y'_1\}$  ou calcular os autovalores da matriz  $Y'^+_1 Y'_2$ .

7. Encontrar os resíduos por meio do problema de mínimos quadrados descrita pela Equação 18.

Essas etapas foram traduzidas em um algoritmo na linguagem *python*, implementado em uma biblioteca própria customizada. Para sua implementação, foram usadas funções já prontas da biblioteca *numpy*, como a função para o cálculo da pseudoinversa de Moore-Penrose, cálculo de autovalores e de resolução de mínimos quadrados. É importante ressaltar que, por se tratar de um algoritmo com características numéricas em múltiplas etapas, algumas adaptações e aproximações foram feitas de maneira a viabilizar seu funcionamento. Foi necessário aproximar valores de saída dos cálculos dos polos  $z_i$  e resíduos  $R_i$  para zero em casos os quais seus valores calculados eram significativamente baixos. Essa aproximação foi feita a partir de um valor limite, definido a critérios do usuário, para o qual valores abaixo ou igual ao valor limite eram truncados. Além disso, para as funções externas, foi necessário se atentar ao seus parâmetros próprios de filtro, usados em etapas intermediárias de cálculos que utilizam o SVD. Nesse último caso, foi definido como filtro de corte o valor de  $10^{-7}$ .

### 3.2 Separação de variáveis

Com o intuito de melhorar o controle e entendimento dos parâmetros calculados pelo MPM, foi feita uma correspondência entre os polos  $z_i$  e resíduos  $R_i$  com os parâmetros originais que compõe o sinal, nesse caso,  $S_{0,i}$ ,  $\phi_i$ ,  $\omega_i$  e  $T_{2,i}$ . Considerando que um sinal de Espectroscopia por Ressonância Magnética (do inglês *Magnetic Resonance Spectroscopy*, MRS) pode ser representado pela Equação 23, sua versão discreta pode ser escrita assumindo  $t = kT_s$ , sendo  $T_s$  o período de *sampling* do sinal, e  $k$  um inteiro representando o passo, como na Equação 24.

$$S = \sum_{i=1}^M S_{0,i} e^{j\omega_i t - \frac{t}{T_2}} e^{j\phi_i} \quad (23)$$

$$S = \sum_{i=1}^M S_{0,i} e^{j\phi_i} (e^{j\omega_i T_s - \frac{T_s}{T_2}})^k \quad (24)$$

Considerando que a aproximação feita por meio da Equação 22 é uma representação suficientemente correta do sinal, é possível estabelecer uma equivalência entre os parâmetros originais e calculados, a partir da Equação 25.



$$\sum_{i=1}^M S_{0,i} e^{j\phi_i} (e^{j\omega_i T_s} e^{-\frac{T_s}{T_2}})^k = \sum_{i=1}^M R_i z_i^k \quad (25)$$

Essa equivalência demonstra que enquanto os polos  $z_i$  representam  $e^{j\omega_i T_s} e^{-\frac{T_s}{T_2}}$ , os resíduos representam a multiplicação  $S_{0,i} e^{j\phi_i}$ . Considerando a representação polar de um número complexo, na Equação 26, é possível visualizar uma maneira simples de cálculo dos parâmetros originais a partir de  $z_i$  e  $R_i$ , concluindo que  $e^{-\frac{T_s}{T_2}}$  corresponde ao módulo de  $z$ ,  $\omega_i T_s$  ao argumento de  $z$ ,  $S_{0,i}$  ao módulo de  $R$ , e  $\phi_i$  ao argumento de  $R$ . A Tabela 1 reúne as equivalências resultantes.

$$z = A e^{j\theta} \quad (26)$$

$S_0$	$\phi$	$\omega$	$T_2$
$ R $	$\arg(R)$	$-\frac{1}{T_s} \arg(z)$	$-\frac{T_s}{\log( z )}$

Tabela 1: Relação entre os parâmetros calculados e originais.

Com o cálculo dos parâmetros individuais bem estabelecidos, foi implementada uma função na biblioteca própria para realização dessa tarefa. Foi feita uma adaptação dos cálculos dos argumentos de  $R$  e  $z$ , visto que a função *arctan2* da biblioteca *numpy* retornava valores no intervalo  $[-\pi, \pi)$ , sendo que o desejado era o intervalo  $[0, 2\pi)$ . Para isso, foi somado  $2\pi$  para valores negativos da função, de maneira a adequá-la às necessidades apresentadas.

### 3.3 Sinais de Controle

#### 3.3.1 Sinal de controle completo

Com o objetivo de caracterizar o comportamento do algoritmo com relação às suas variáveis  $L$  e  $p$ , foi gerado um sinal de controle de comportamento previsível e variáveis geradoras conhecidas. O sinal se trata de uma simulação de MRS elaborada a partir de um algoritmo próprio, gerada a partir de parâmetros já caracterizados de metabólitos conhecidos, e fase  $\phi = 0$ . O mesmo foi gerado com um campo de  $B_0 = 3T$ , começando em  $t_0 = 0.0$  s e terminando em  $t_n = 1.0$  s, com um passo de  $dt = \frac{1}{2048} \approx 0.0005$  s, resultando em um sinal com 2048 pontos, de maneira a se adequar ao padrão de medidas clínicas. O sinal pode ser conferido pela Figura 2.

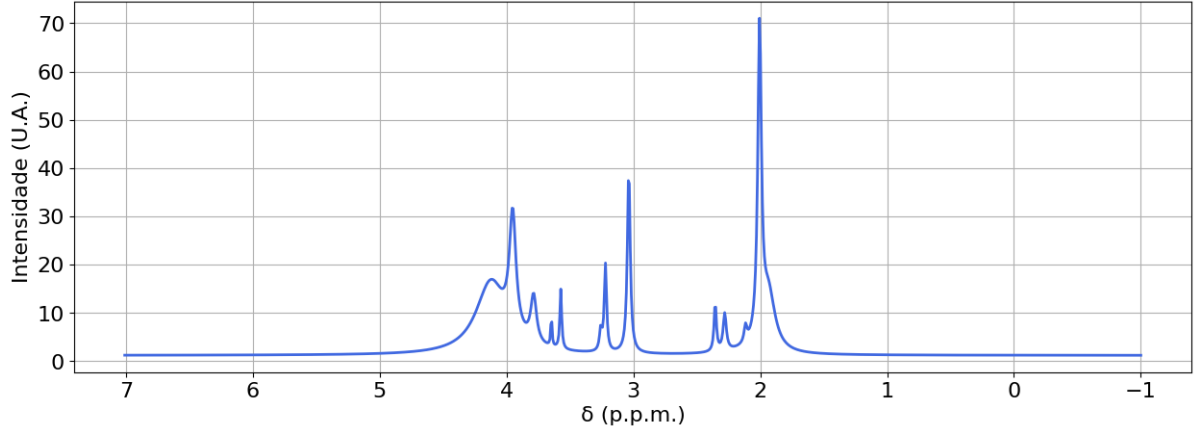


Figura 2: Sinal de controle gerado para caracterização das variáveis.

### 3.3.2 Sinal de controle de pico único

Além de averiguar a influência dos parâmetros tradicionais no processo do algoritmo, foi também verificada a necessidade de avaliar o comportamento dos parâmetros resultantes do processo,  $S_0$ ,  $\phi$ ,  $\omega$  e  $T_2$ , com relação à qualidade do sinal, traduzida pela sua relação sinal-ruído (do inglês, *Signal-to-Noise Ratio*, SNR). Para prosseguir, foi escolhido um novo sinal de controle: a simulação de MRS de um único metabólito, o N-Acetylaspartato, ou NAA, escolhido por ser o maior pico dentre os metabólitos comumente encontrados no cérebro. Essa escolha foi motivada pela simplificação apresentada, visto que um único pico facilitaria o processo de investigação de possíveis fenômenos associados à distorções causada pela presença de ruído. A simulação foi feita com as mesmas condições de contorno apresentadas na Subsubseção 3.3.1, podendo ser conferida pela Figura 3.

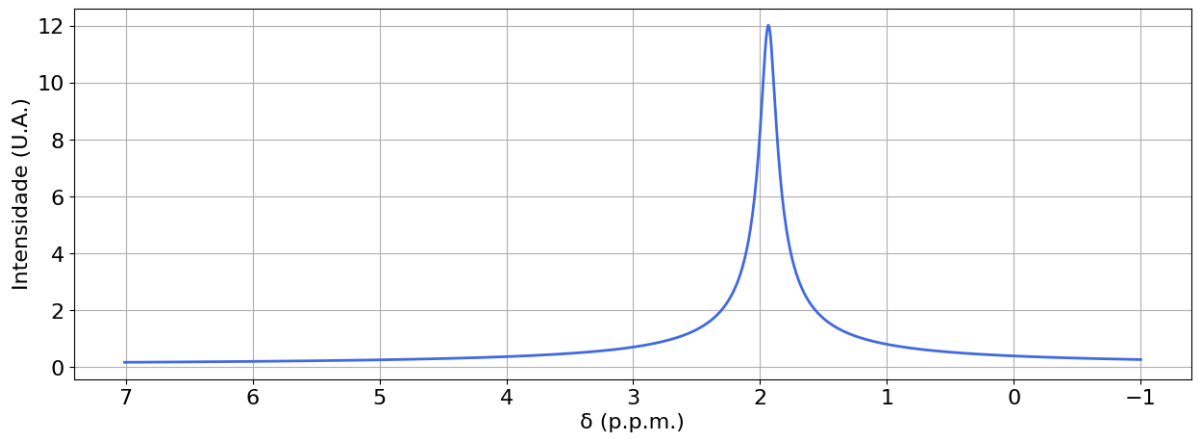


Figura 3: Sinal de controle de pico único gerado para estudo dos parâmetros resultantes.

### 3.4 Corrupção dos Sinais

A corrupção dos sinais foi realizada pela função *awgn*, da biblioteca *scikit-commpy*, uma biblioteca código aberto que reúne ferramentas para o estudo e simulação de sinais de comunicações. Apesar de se tratar de outra área aparentemente distante, seu uso foi considerado adequado pois seu funcionamento se mostrou condizente com o propósito deste projeto, considerando seu funcionamento estável e possibilidade de controle do parâmetro SNR. A função mencionada recebe como entrada o sinal e um valor de SNR, em dB, para o qual o sinal deve ter ao ser corrompido, e parte da definição estabelecida em Equação 27.

$$SNR_{linear} = \frac{P_{sinal}}{P_{ruído}} \quad (27)$$

$$SNR_{linear} = 10^{\frac{SNR}{10}} \quad (28)$$

Sendo  $P_{sinal}$  e  $P_{ruído}$  a potência do sinal e a potência do ruído, respectivamente. A função gera o sinal corrompido a partir de um desvio padrão do ruído,  $\sigma$ , calculado a partir do SNR de entrada, como definido pela Equação 30 [4]. O  $\sigma$  é calculado a partir da potência do sinal de entrada, calculado pela Equação 31, do valor de SNR linear, calculado pela Equação 28, e de uma variável  $r$  no divisor, definida pelo algoritmo como "Taxa do código FEC se usado", que aqui será omitida por ter tido valor 1.

$$\begin{aligned} SNR_{linear} &= \frac{P_{sinal}}{P_{ruído}} \\ P_{ruído} &= \frac{P_{sinal}}{SNR_{linear}} \end{aligned} \quad (29)$$

$$P_{ruído} = 2\sigma^2$$

$$\sigma^2 = \frac{P_{sinal}}{2SNR_{linear}} \quad (30)$$

$$P_{sinal} = \frac{1}{N} \sum_{min}^{max} |x(i)|^2 \quad (31)$$

A corrupção é feita com ruído gaussiano, diretamente no espectro do sinal.

## 4 Resultados

### 4.1 Implementação do MPM

O algoritmo foi implementado, assim como descrito na Subseção 3.1, em etapas. Primeiro, foi testada a capacidade do mesmo de identificar exponenciais decrescentes em condições ideais, a partir da implementação que não leva em conta ruído. A Figura 4 demonstra um exemplo, para o sinal descrito na Subsubseção 3.3.1, de decomposição e reconstrução do mesmo a partir da implementação que ignora a etapa que usa SVD, para  $L$  de 40%. O algoritmo parece ter obtido sucesso na identificação dos picos dos metabólitos.

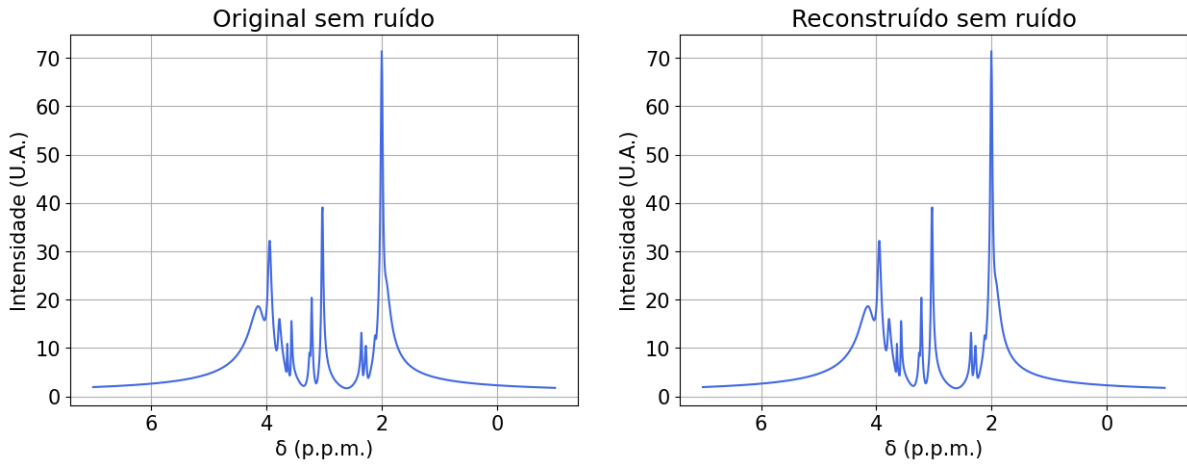


Figura 4: Sinal limpo reconstruído a partir do processo de MPM sem ruído

Realizada a etapa de filtragem em um sinal sem ruído, foi implementada então a etapa da filtragem por SVD. Para testar seu funcionamento, o sinal anteriormente usado foi corrompido com um SNR de valor  $15dB$ . A Figura 5 demonstra um exemplo da decomposição e reconstrução desse sinal corrompido, para  $L$  de 40% e limite de ruído de  $1^{-24}$ . O limite de ruído foi escolhido extremamente baixo de maneira a testar se o algoritmo continuava a funcionar após implementação da etapa de SVD.

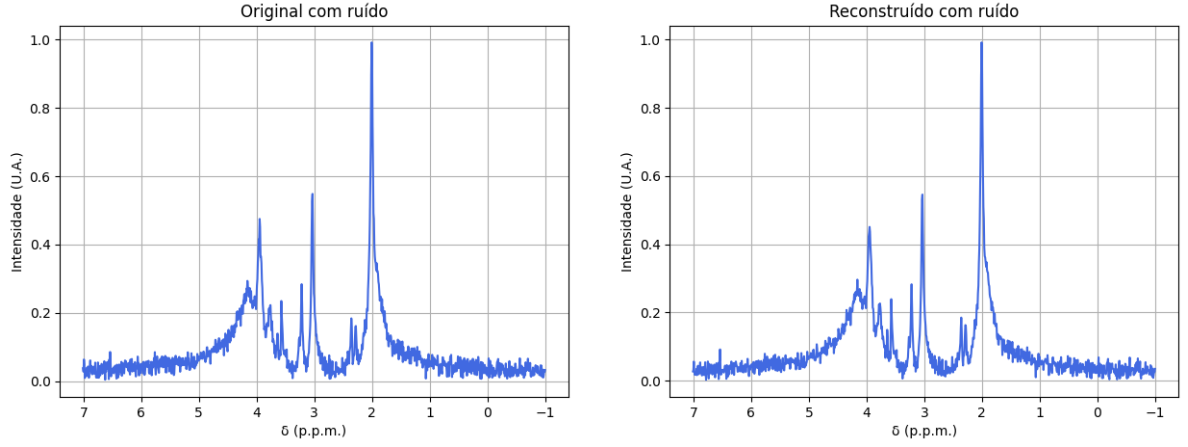


Figura 5: Sinal corrompido reconstruído a partir do processo de MPM com ruído de SNR equivalente a  $15dB$

## 4.2 Separação de variáveis

Para testar a capacidade de cálculo das variáveis  $S_0$ ,  $\phi$ ,  $\omega$  e  $T_2$  pelo programa, foi usado o sinal limpo descrito na Subsubseção 3.3.2. Esse sinal foi processado pelo MPM no domínio do tempo, e a partir dos polos e resíduos calculados, teve suas variáveis resultantes calculadas. Todas essas variáveis, por sua vez, foram ordenadas em ordem crescente a partir da amplitude  $S_0$ .

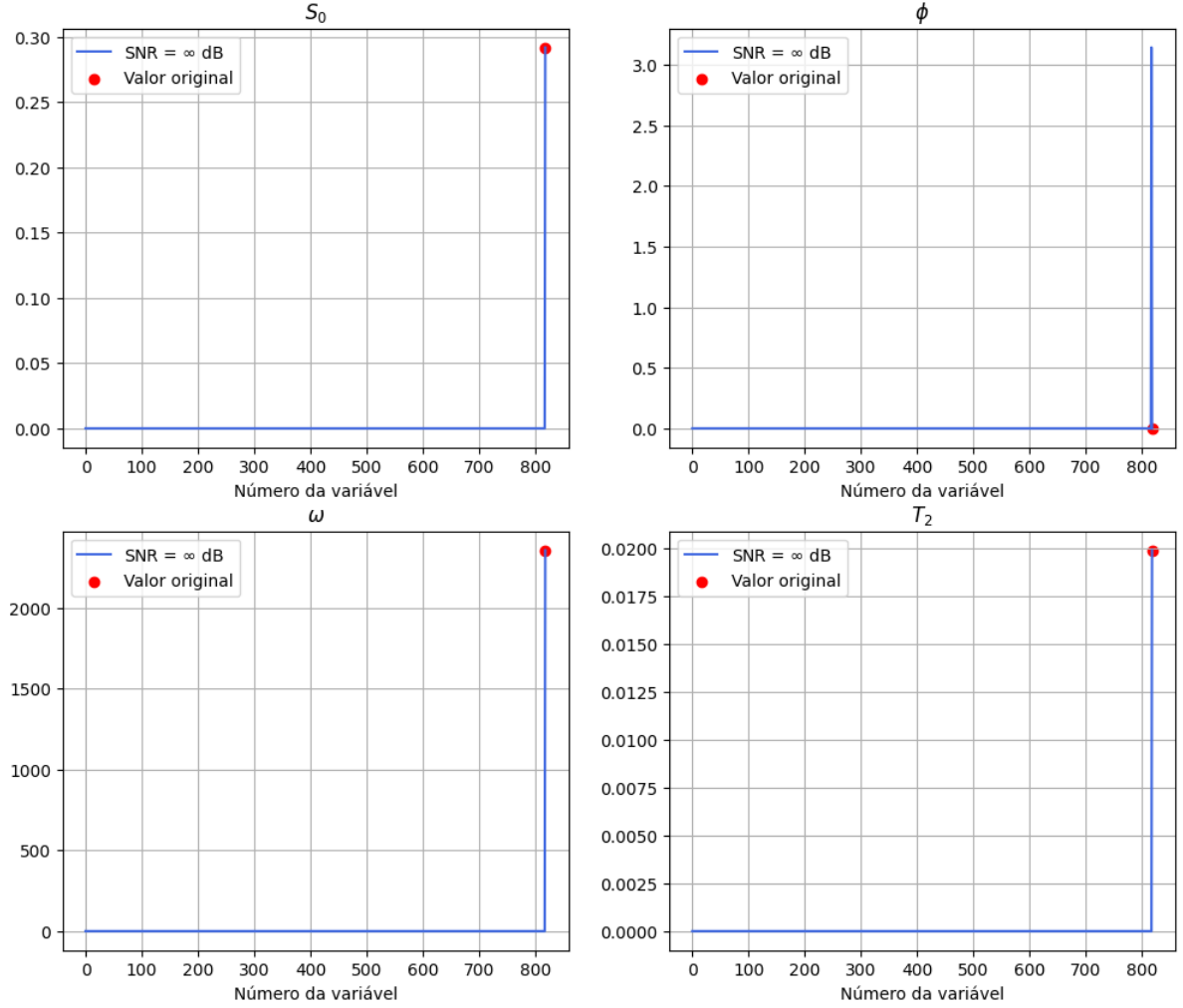


Figura 6: Valor das variáveis para o sinal limpo

Como pode ser verificado pela Figura 6, o cálculo foi realizado com sucesso, no qual todas as variáveis esperadas foram encontradas. É importante comentar que, por motivos de origem numérica, uma fase errônea de aproximadamente  $\pi$  foi calculada, porém na reconstrução do sinal original, essa fase multiplica uma amplitude  $S_0$  de valor 0, fazendo-a desaparecer.

### 4.3 Testagem de $L$ sem ruído.

Com a simulação e o algoritmo implementados com sucesso, foi possível enfim iniciar o estudo de seu comportamento na aplicação em sinais para diferentes parâmetros e processos, como inicialmente planejado. Como primeira etapa, foi testado para o algoritmo implementado como as variáveis resultantes do processo se comportavam com relação à variação de  $L$ , no sinal anteriormente apresentado, livre de ruído. Para isso, o algoritmo foi rodado com  $L$

variando de  $L_0 = 20\%$  do  $N$  a  $L_N = 80\%$  do  $N$ , limites além de  $N/3$  e  $N/2$  previamente estabelecidos, com o objetivo de verificar se havia alguma flutuação significativa nos valores das variáveis resultantes. Para cada  $L$ , variado nesse intervalo com um passo de  $5\%$  do  $N$  foram feitas 10 médias. O parâmetro  $p$  foi fixado em 15, e a comparação foi feita por meio da Raiz do Erro Quadrático Médio (do inglês *Root-mean-squared Error*, RMSE), podendo ser conferida pela Figura 7.

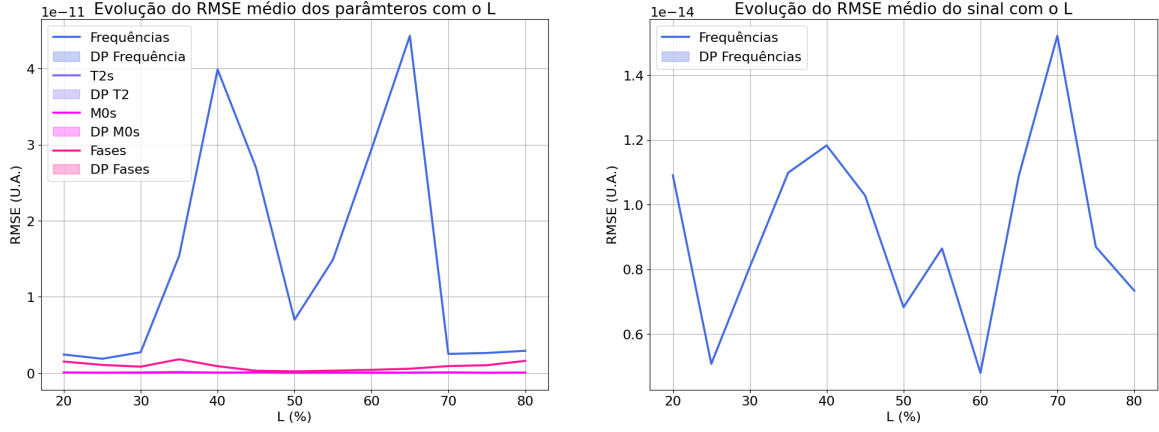


Figura 7: Resultados para a variação de  $L$

Pela Figura 7, é possível concluir que a escolha do  $L$  parece não impactar a capacidade de cálculo das variáveis originais do sinal limpo de ruído, visto que as flutuações de valor do RMSE médio são praticamente zero.

#### 4.4 Testagem de SVD sem ruído.

Também foi testado, para o mesmo sinal, como as variáveis resultantes se comportavam com relação ao outro parâmetro do MPM, nesse caso, a ordem da variável de corte do SVD.  $p$  foi variado entre  $p_0 = 3$  e  $p_N = 15$ , com um passo de uma unidade, e  $L = 40\%$ . Para cada  $p$ , foram feitas também 10 médias.  $p_0$  foi escolhido como 3 pois constatou-se que a partir de  $p = 2$  os picos principais do sinal eram filtrados, impossibilitando a análise por meio do RMSE.

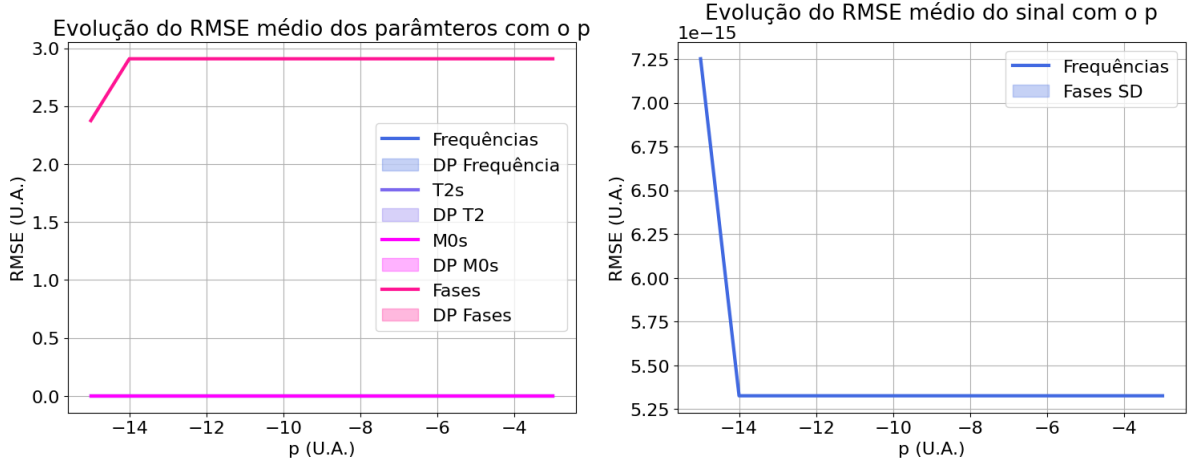


Figura 8: Resultados para a variação de  $p$

A priori, o evidente erro com relação às fases pode chamar a atenção, visto que a fase de entrada do sinal de controle era  $\phi = 0$ . Constatou-se porém, após verificar manualmente os valores de fases gerados, que sua origem reside em pequenos erros numéricos que se propagam na função  $\arctan2$ , provocando um resultado de saída de aproximadamente  $2\pi$  para fases calculadas, equivalente à fase esperada. Esclarecido esse importante detalhe, é possível concluir que, novamente para o sinal limpo, a variável  $p$  não demonstra nenhuma influência significativa na capacidade de cálculo das variáveis originais.

#### 4.5 Relação entre as variáveis e o ruído

Caracterizou-se o comportamento das variáveis anteriormente mencionadas com relação ao ruído, a simulação de pico único da Subsubseção 3.3.2 foi corrompida com ruídos de diferentes níveis, variando o SNR resultante de 1.0 a 100.0.



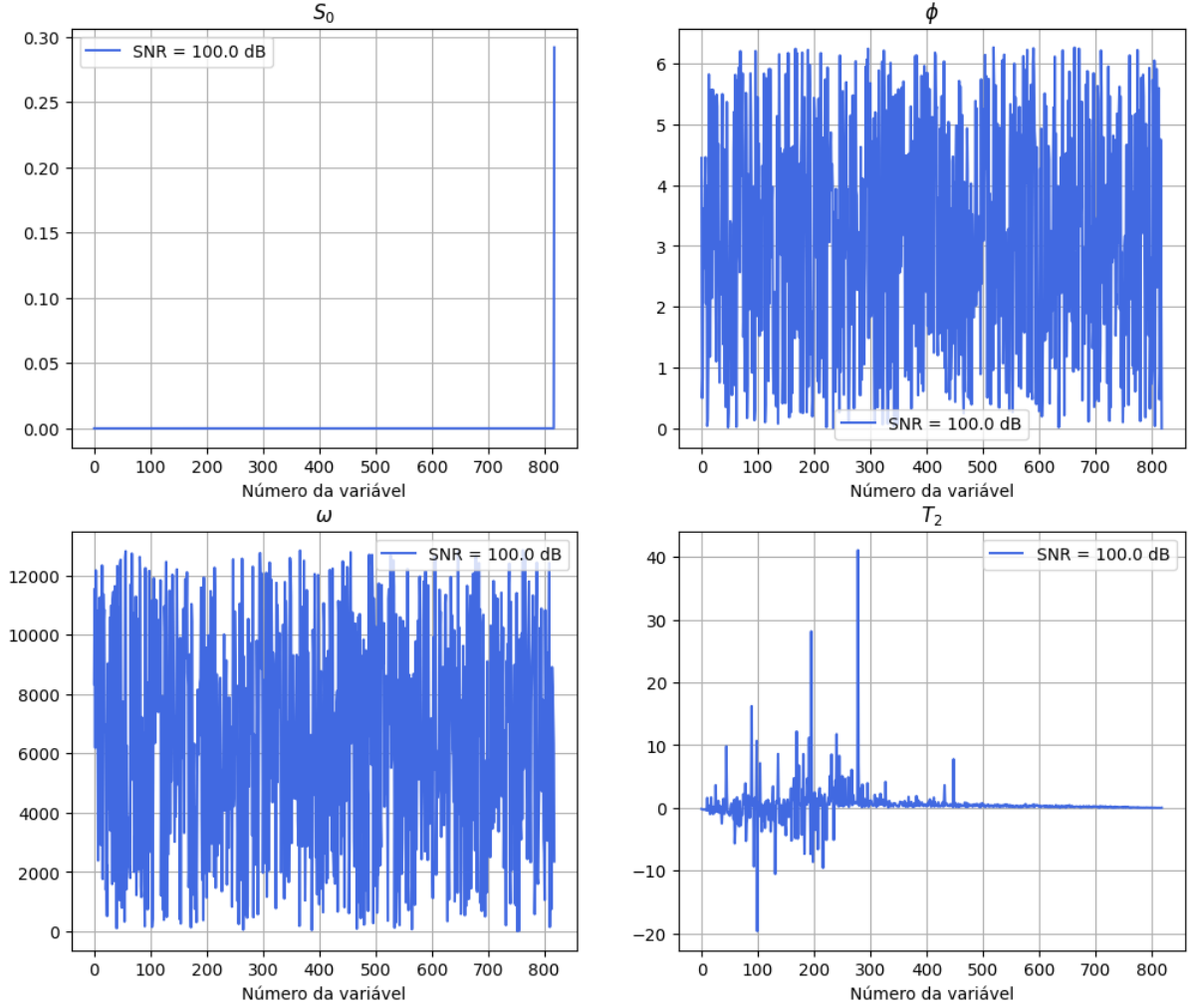


Figura 9: Valor das variáveis para o segundo SNR, de valor 100

Esse fenômeno de variáveis de valores "exóticos", presente no cálculo da fase em sinal limpo na Figura 6, se mostrou comum ao longo dos outros processos de cálculo, tornando necessária a sua análise no contexto do algoritmo. Como pode ser verificado pela Figura 9, para um valor de SNR mais baixo, nesse caso 100, o cálculo das frequência, tempos de decaimento e das fases demonstrou o mesmo comportamento anteriormente descrito, com valores oscilando de maneira ruidosa e pouco previsível, e amplitudes significativas. Assim como descrito no caso de SNR infinito, a priori esse fenômeno pode se apresentar como obstáculo à capacidade de cálculo do MPM, porém novamente, quando analisada sob o contexto das outras variáveis, a maioria dos valores, quando multiplicados pela amplitude  $S_0$ , se anula.

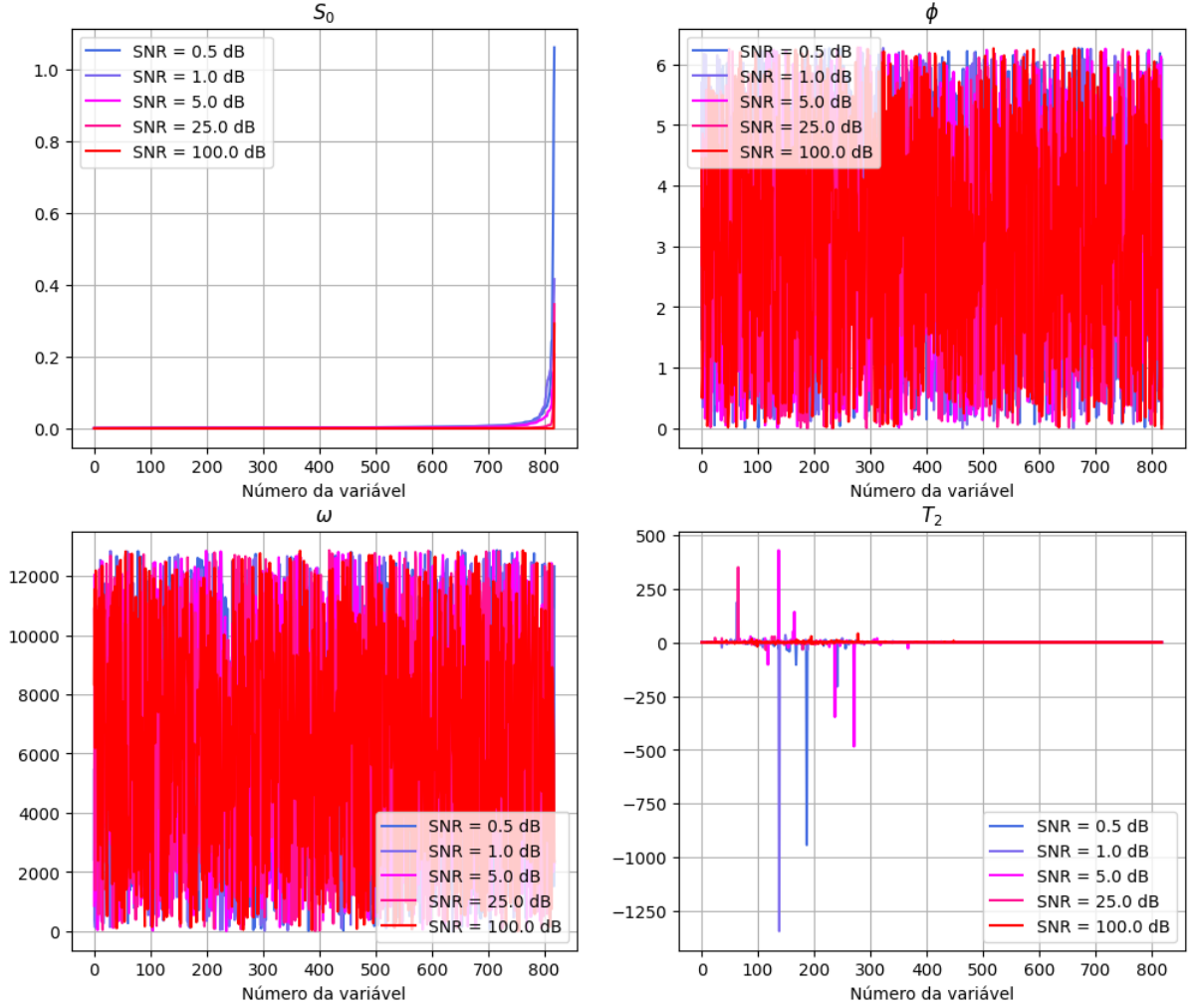


Figura 10: Valor das variáveis para o segundo SNR, de valor 100

A variável  $S_0$  aqui se mostra dominante para a construção do sinal final. Como verificado pela Figura 10, conforme o SNR decresce, partindo da ordenação crescente de amplitudes, a distribuição de valores de  $S_0$  não nulos aumenta. Esse aumento tanto em quantidade quanto em amplitude confere ao sinal final uma quantidade maior de frequências aparentemente aleatórias, compondo assim o ruído característico da corrupção originalmente aplicada.

## 4.6 Relação entre picos e ruído

Outra relação explorada por meio do MPM foi o comportamento das frequências do sinal original após serem corrompidas. Para isso, foi usado o sinal implementado em Subsubseção 3.3.2, de pico único, de maneira a simplificar a análise. Esse sinal foi corrompido com ruídos de SNRs resultantes no intervalo de 1.0 a 100.0, com passos não lineares. Cada

sinal corrompido era desconstruído pelo MPM e tinha suas variáveis calculadas. Foi realizada então uma filtragem a partir do valor de seus picos  $S_0$ , na qual se calculava o maior pico, e a partir dele eram escolhidos os  $n$  picos com valor maior ou igual a 5% do maior pico. Esse procedimento foi realizado 50 vezes para cada SNR, para análise estatística.

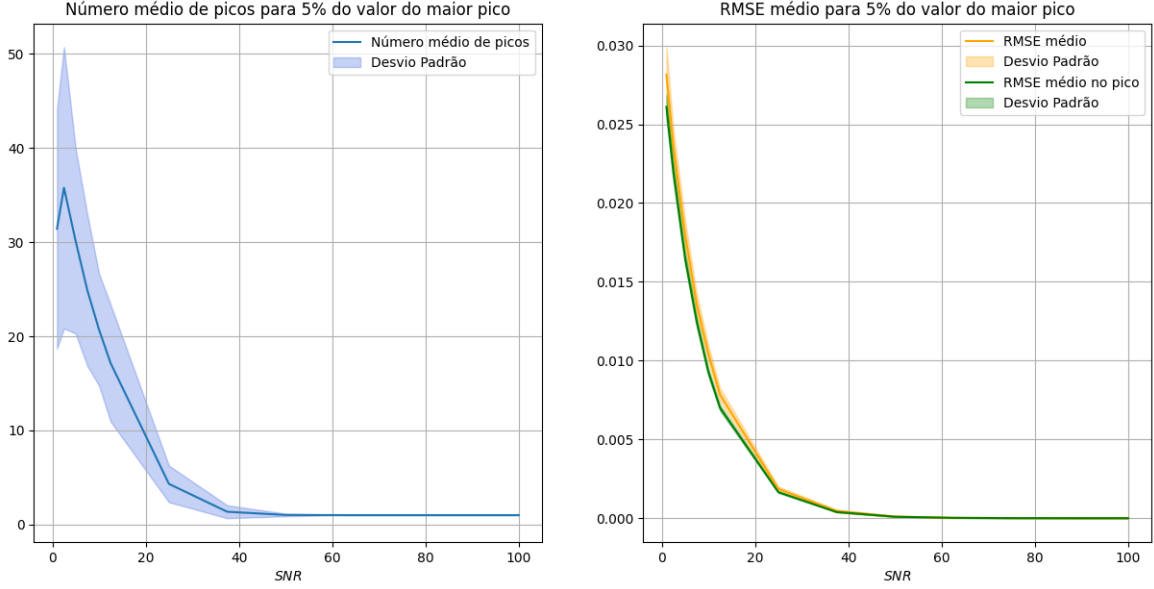


Figura 11: Comportamento dos picos com a progressão do SNR.

É possível verificar pela Figura 11, dentro de um certo intervalo, um comportamento de decrescimento na quantidade de picos identificados pelo processo descrito anteriormente. Apesar da grande dispersão do número médio de picos para SNRs baixos. Além disso, foi calculado o RMSE entre o sinal total original e o RMSE ao redor do pico principal, e como também pode ser constatado pela Figura 11, mesmo para os SNRs mais baixos, se escolher-se reconstruir o sinal apenas com os picos pelo critério especificado, o erro da reconstrução será baixo, diminuindo de maneira exponencial conforme o valor do SNR do sinal aumenta.

As figuras 12, 14 e 16 demonstram alguns exemplos de níveis de SNR, a filtragem com picos pelo critério anteriormente especificado, e o resíduo, aqui definido como a subtração do sinal original pela filtragem. Analisando-as em conjunto com as figuras 13, 15 e 17, corrobora-se a percepção sobre a diminuição no número de picos com o SNR, permitindo uma análise além: é possível perceber que apesar da qualidade do sinal piorar significativamente, o maior pico, correspondente ao do NAA, se mantém.

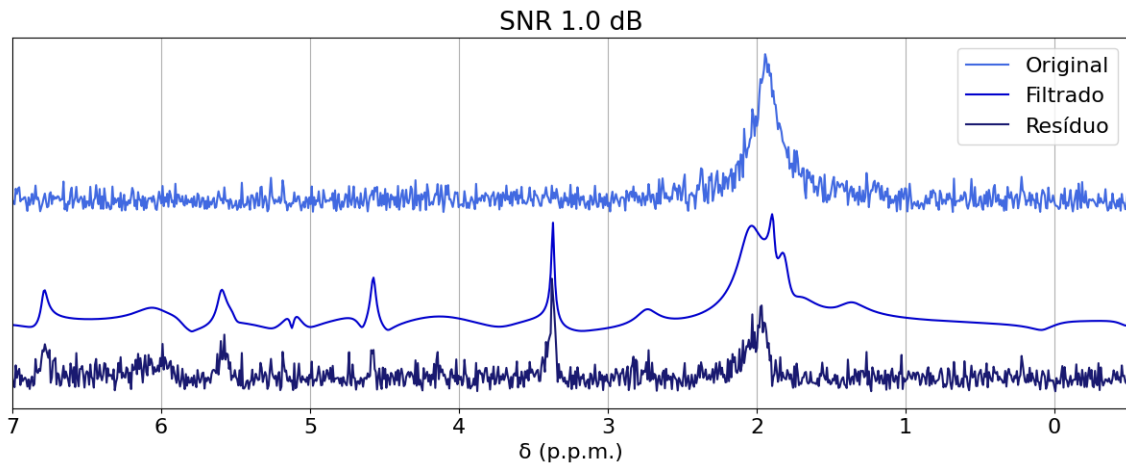


Figura 12: Exemplo de sinal composto apenas com picos pelo critério estabelecido para 1dB

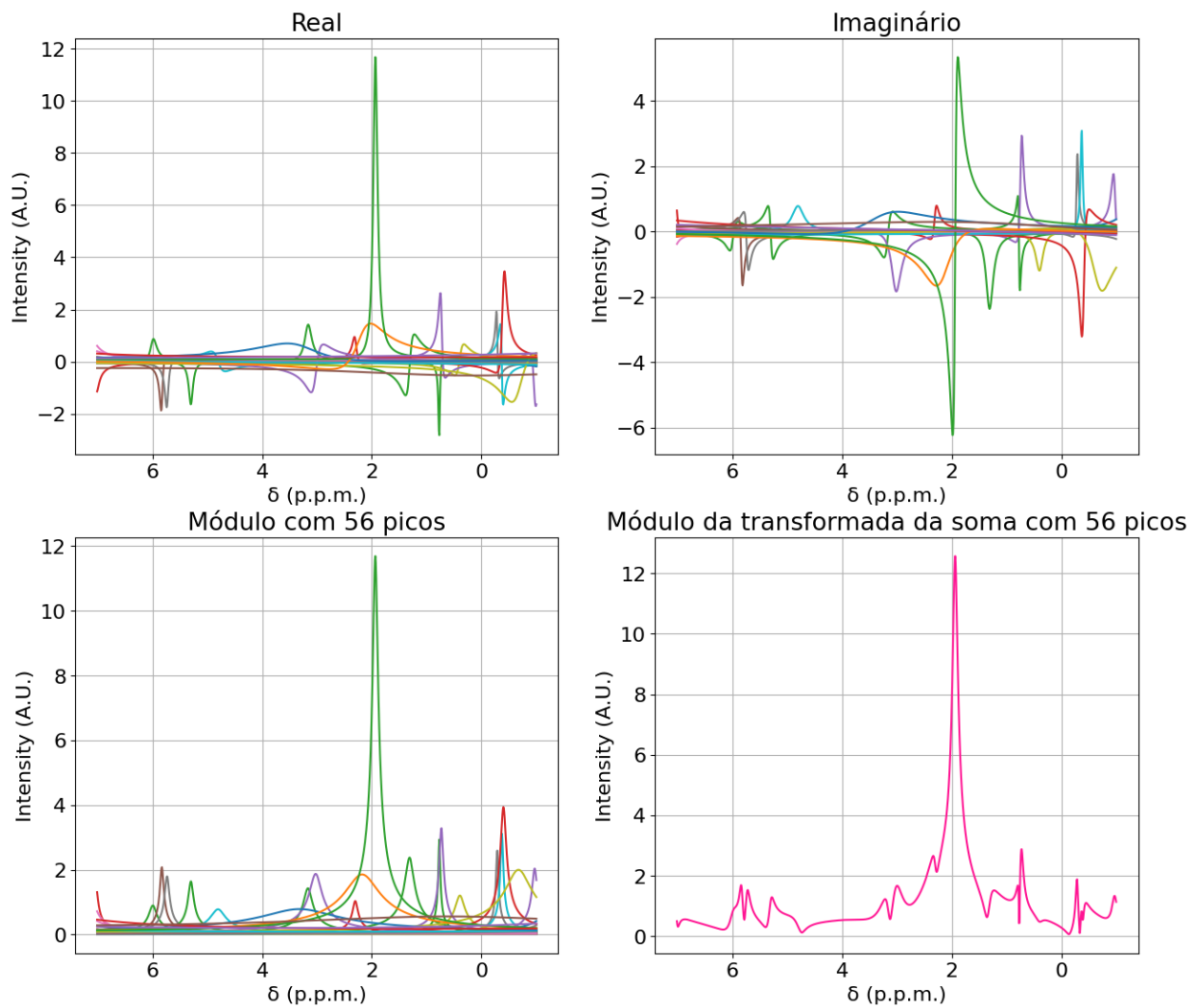


Figura 13: Picos selecionados pelo critério estabelecido para 1dB

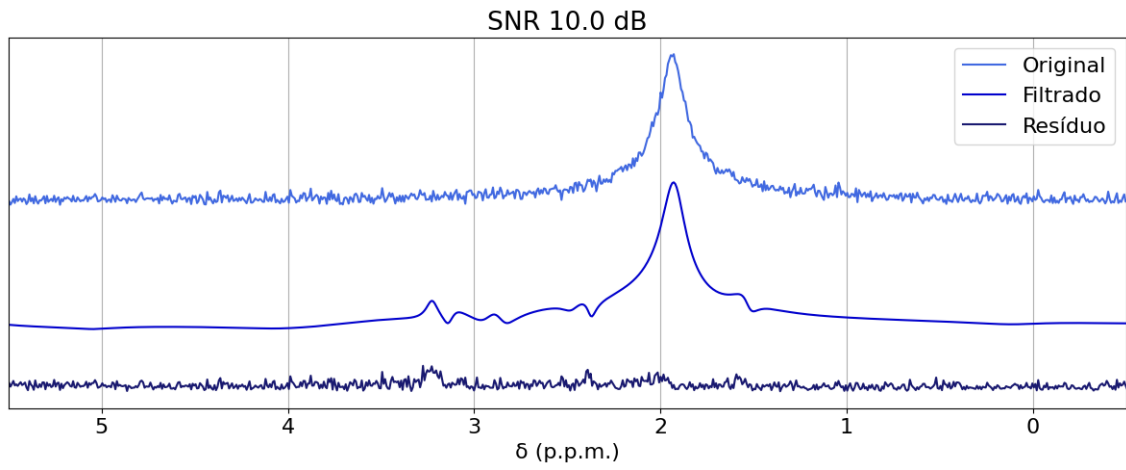


Figura 14: Exemplo de sinal composto apenas com picos pelo critério estabelecido para 10dB

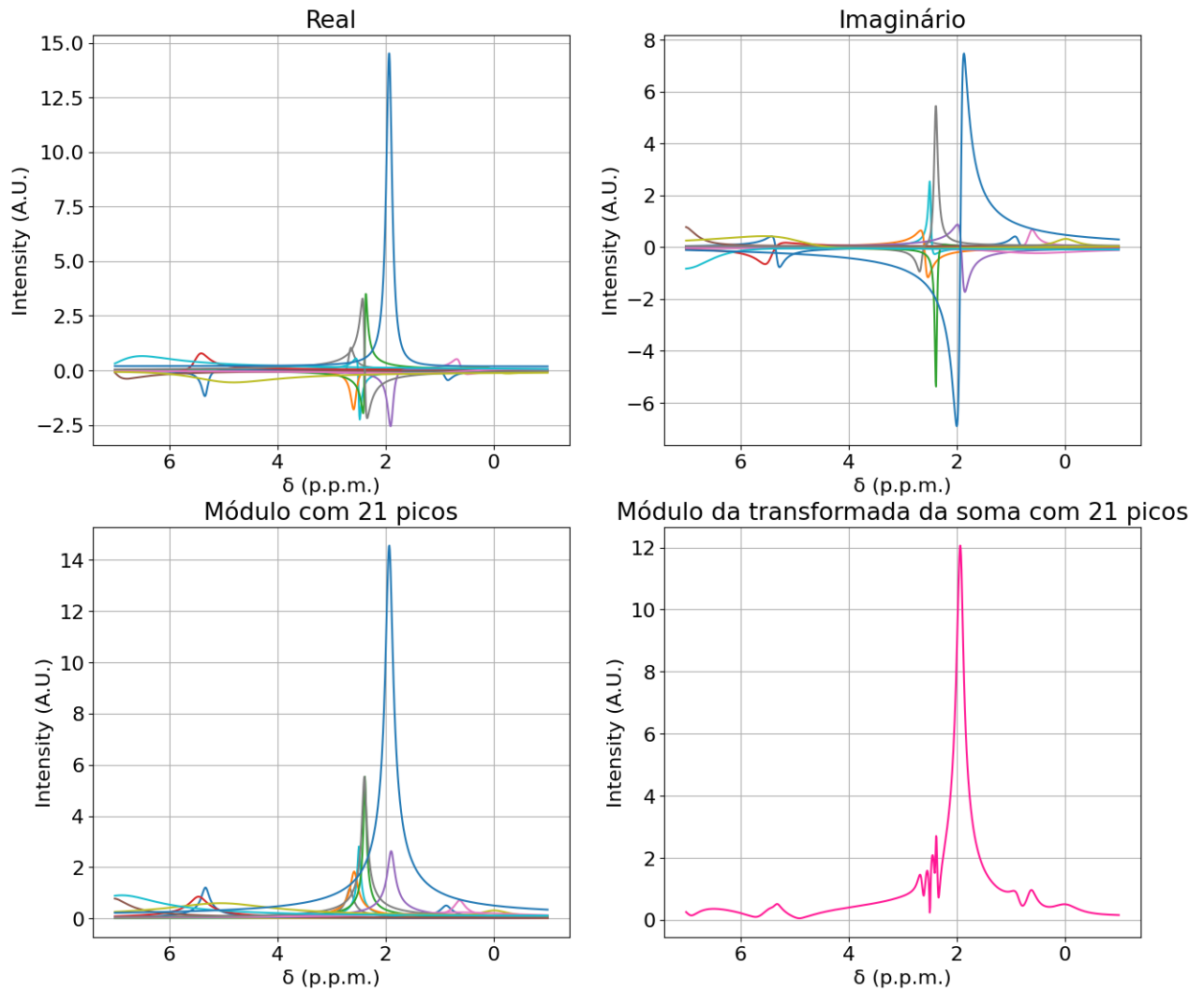


Figura 15: Picos selecionados pelo critério estabelecido para 10dB

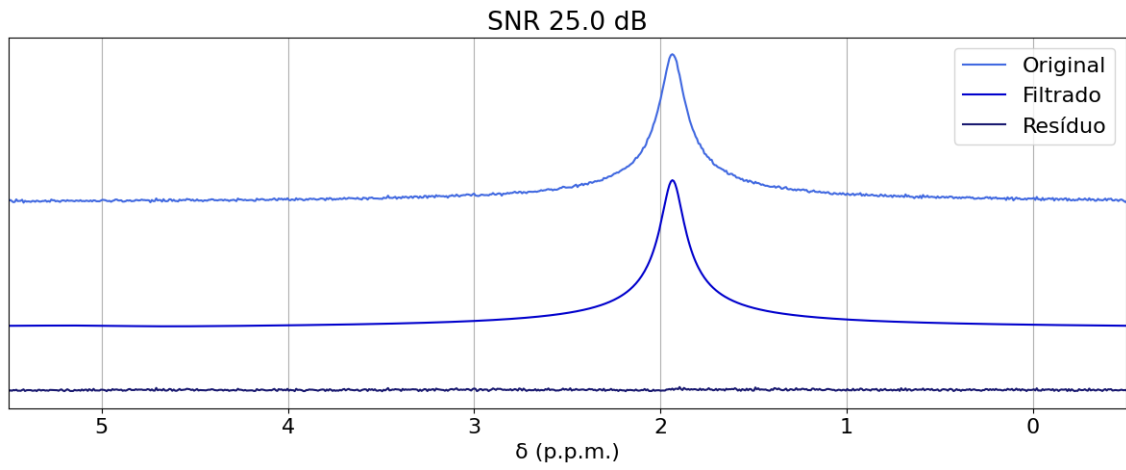


Figura 16: Exemplo de sinal composto apenas com picos pelo critério estabelecido para 25dB

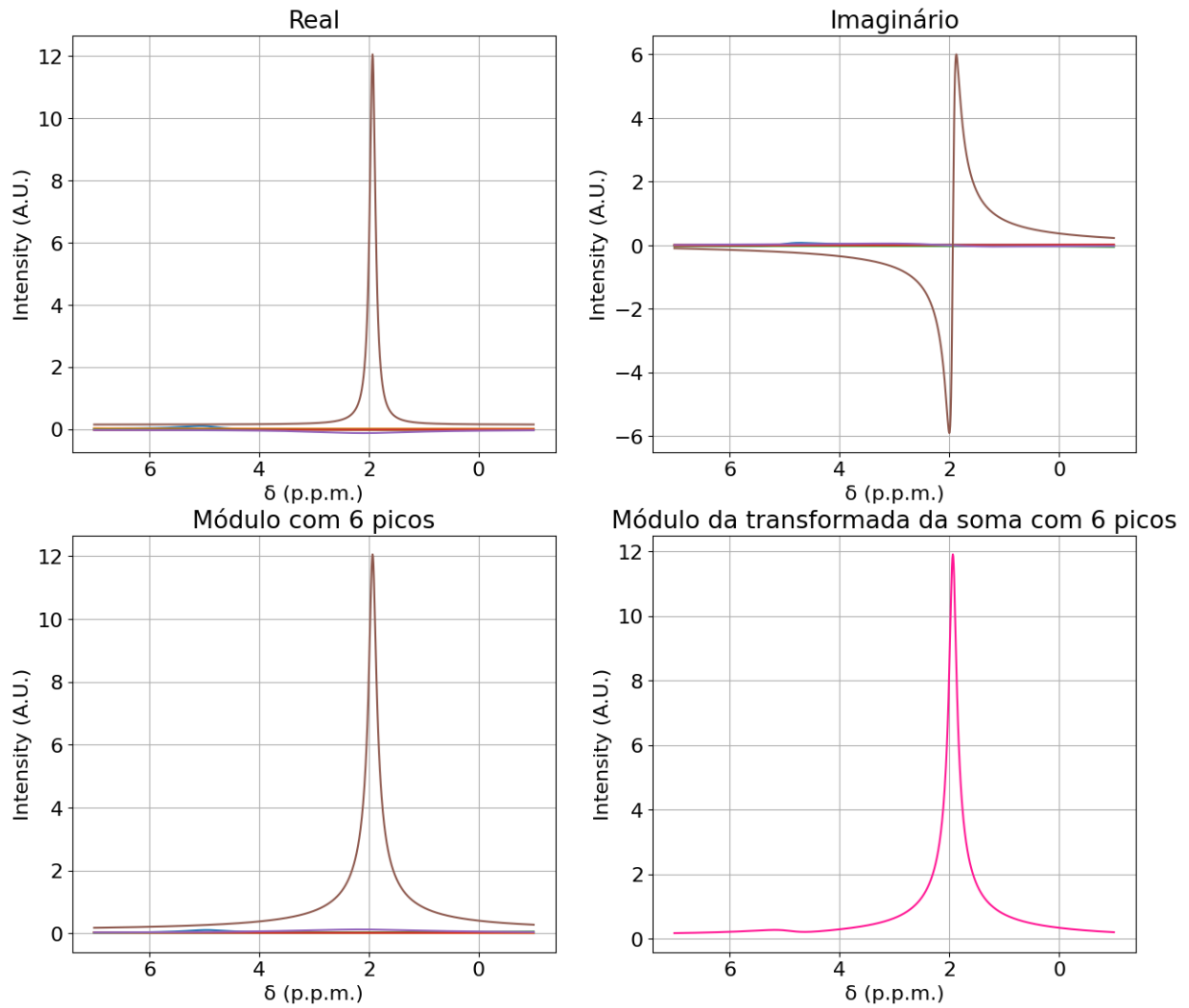


Figura 17: Picos selecionados pelo critério estabelecido para 25dB

## 5 Conclusão

Considerando sua capacidade de extração de informações de sinais pós-aquisição, o MPM se mostra uma interessante ferramenta para o estudo de sinais de MRS. Sua capacidade de separar os parâmetros individuais do sinal abre novos caminhos para o entendimento dos fenômenos na área, com a perspectiva de mitigar os efeitos associados à corrupção por ruído em sinais clínicos, possibilitando uma análise de maior qualidade. No tocante aos testes realizados, o parâmetro de amplitude,  $S_0$ , desempenha importante papel de filtro de parâmetros, demonstrando capacidade de seleção e modelagem do formato do sinal final. Com relação ao comportamento dos picos, foi possível verificar que, apesar do número de picos de valor significativo aumentar, o maior pico ainda se manteve, mesmo que com frequência distorcida. Como próxima etapa, pretende-se investigar o comportamento dos parâmetros do sinal com a sobreposição de dois picos, aumentando, assim, o potencial de aplicabilidade do método.

## Referências

- [1] J.F. Hauer, C.J. Demeure, and L.L. Scharf. Initial results in prony analysis of power system response signals. *IEEE Transactions on Power Systems*, 5(1):80–89, 1990.
- [2] Y.C. LIANG, H.P. LEE, S.P. LIM, W.Z. LIN, K.H. LEE, and C.G. WU. Proper orthogonal decomposition and its applications—part i: Theory. *Journal of Sound and Vibration*, 252(3):527–544, 2002.
- [3] T.K. Sarkar and O. Pereira. Using the matrix pencil method to estimate the parameters of a sum of complex exponentials. *IEEE Antennas and Propagation Magazine*, 37(1):48–55, 1995.
- [4] Mathuranathan Viswanathan. *Digital Modulations using Python*. Independently Published, December 2019.