

Redução da Complexidade do Max-Log-MPA em Sistemas SCMA

Luís O. Mataveli

Resumo—Para a detecção em sistemas *Sparse Code Multiple Access* (SCMA) o algoritmo ótimo é o *Maximum a Posteriori* (MAP). Embora tenha o melhor desempenho, o MAP tem alta complexidade para casos em que há grande quantidade de usuários. Sendo assim, o *Message Passing Algorithm* (MPA) vem sendo estudado como forma de detecção para esses sistemas. Mesmo sendo menor, a complexidade do MPA também é alta e pode ser reduzida. Uma maneira é usar aproximações matemáticas, como é feito no Max-Log-MPA. Tendo este algoritmo como base, neste trabalho é proposto um método que reduz a complexidade da parte iterativa do Max-Log-MPA. Isso é feito através da comparação de valores de algumas variáveis com limiares de decisão.

Palavras-Chave—SCMA, Max-Log-MPA, Redução de Complexidade.

Abstract—For detection in SCMA (*Sparse Code Multiple Access*) systems the optimal algorithm is MAP (*Maximum a Posteriori*). Although it has the best performance, MAP has high complexity for cases where there are a large numbers of users. Thus, the MPA (*Message Passing Algorithm*) has been studied as an option for detection in these systems. Even though it is smaller, MPA's complexity is also high and can be reduced. One way is to use mathematical approximations, such as is done in Max-Log-MPA. Based on this algorithm, we propose in this work a method that eliminates some calculations within the iterative part of Max-Log-MPA. This is done by comparing the values of some variables with decision thresholds.

Keywords—SCMA, Max-Log-MPA, Complexity Reduction.

I. INTRODUÇÃO

Em aplicações de IoT, Indústria 4.0, monitoramento remoto e outros tipos de comunicação *machine-to-machine* usando redes móveis como a 5G, por exemplo, há a necessidade de um número grande de dispositivos enviarem informações usando recursos limitados, como é o espectro de frequências. Assim, sistemas que conseguem operar com bom desempenho mesmo em situações de sobrecarga são de grande interesse.

Sparse Code Multiple Access (SCMA) é um método de múltiplo acesso não ortogonal que vem sendo amplamente estudado devido a sua característica de suportar sistemas em sobrecarga. Assim, mais dispositivos conseguem transmitir ao mesmo tempo usando uma menor quantidade de subportadoras [1]. Os símbolos a serem transmitidos e as subportadoras utilizadas são previamente definidas por um *codebook* e o processo de modulação e de espalhamento espectral é feito em conjunto. Os sinais de todos os usuários sofrem com desvanecimento e ruído aditivo e devem ser recuperados pelo receptor.

Luís O. Mataveli, TIC, Petrobras, Rio de Janeiro-RJ, e-mail: luismataveli@yahoo.com.br

A detecção ótima dos símbolos poderia ser feita com o algoritmo ótimo *Maximum a Posteriori* (MAP), porém sua complexidade é proibitiva em sistemas reais com grande número de usuários. Assim, para a detecção em sistemas SCMA tem sido utilizado o algoritmo MPA que consegue detectar os sinais com uma complexidade viável. Neste trabalho, será considerado o algoritmo Max-Log-MPA como base para propor um método de baixa complexidade. O Max-Log-MPA faz o uso de uma aproximação matemática do MPA onde troca cálculos mais complexos como exponenciais e logaritmos por adições e subtrações [2].

Existem alguns estudos que buscam reduzir a complexidade do MPA. Dentre eles, podemos citar [4] que utiliza limiares dinâmicos para eliminar cálculos dentro do processo iterativo de forma semelhante ao que é proposto na primeira parte deste trabalho. A limitação deste trabalho é que a complexidade não diminui após cada iteração. Em [5], uma simplificação matemática é feita para trocar operações mais complexas, como multiplicações, por adições e subtrações, a mesma ideia do algoritmo Max-Log-MPA. Em [6] é feita uma aproximação da PDF (*Probability Density Function*) do sinal recebido para eliminar operações complexas, como exponenciais. Em [7] vários procedimentos são propostos para reduzir a complexidade. Um deles é o desenho de *codebooks* para reduzir o número de projeções de símbolos o que leva a uma convergência rápida na parte iterativa do algoritmo. [8] e [9] também propuseram um método para reduzir o número de cálculos em cada iteração, enquanto em [10], [11] a abordagem foi reduzir o número de iterações. Todos esses trabalhos apresentam algumas limitações para reduzir a complexidade. De [4] a [7], a redução da complexidade é a mesma para todas as iterações e de [8] a [11] os autores buscam a redução no número de iterações, mas nas primeiras iterações a complexidade é alta.

Neste trabalho o foco é a redução da complexidade do processo iterativo, por ser o mais complexo da detecção Max-Log-MPA [3]. O método proposto se divide em duas partes e será mostrado que, em alguns casos, é possível reduzir mais de 85% a complexidade do algoritmo Max-Log-MPA já na primeira iteração e ainda mais nas iterações seguintes.

O restante deste trabalho é organizado da seguinte maneira. A Seção II mostra o algoritmo Max-Log-MPA e seus cálculos para a detecção de sinais nos sistemas SCMA. Na Seção III é apresentado um método para a redução do número de cálculos do Max-Log-MPA, baseada em duas abordagens diferentes, e na Seção IV serão mostrados os resultados de simulação com curvas de desempenho e complexidade usando o método proposto em comparação com o algoritmo Max-Log-MPA original. Enfim, a Seção V conclui este trabalho.

II. ALGORITMO MAX-LOG-MPA

A Fig. 1 mostra o diagrama de blocos do receptor de um sistema SCMA usando o Max-Log-MPA. Inicialmente, são calculadas as distâncias Euclidianas entre o sinal recebido y_k e todos os possíveis símbolos transmitidos em cada subportadora (cálculo de $f(\cdot)$). Após este cálculo, o algoritmo entra em um processo iterativo onde há troca de informação entre os nós variáveis (VN), representando os usuários, e os nós de função (FN), que representam as subportadoras do sistema. Finalmente, o algoritmo calcula as *Log-Likelihood Ratios* (LLRs) dos símbolos para tomar uma decisão sobre os bits de informação que foram transmitidos. Um exemplo de sistema SCMA muito usado na literatura, e que também será estudado neste artigo, tem 6 usuários que usam 4 subportadoras, ou seja, a sobrecarga neste caso é de 150%. Cada usuário utiliza 2 subportadoras para transmitir um dos 4 possíveis símbolos, assim, em cada subportadora há 3 usuários transmitindo simultaneamente. Como os usuários não transmitem em todas as subportadoras, há uma interferência menor e, além disso, a complexidade de recepção fica menor.

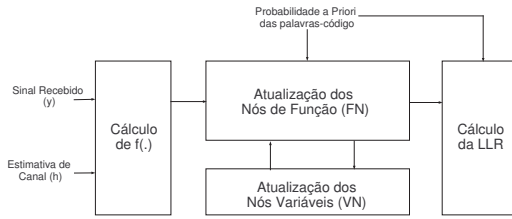


Fig. 1. Diagrama de Blocos do MPA.

O Max-Log-MPA realiza uma aproximação matemática que pode causar uma pequena perda de desempenho em relação ao MPA já que não tem o fator de correção da função Jacobiana, porém tem complexidade menor.

Na Fig. 1 podemos ver que inicialmente são feitos cálculos da distância Euclidiana entre o sinal recebido y_k e todas as combinações de possíveis símbolos transmitidos em cada subportadora e são dados por:

$$f_k(y_k, m_i, N_0, \mathbf{H}_k) = \frac{-1}{N_0} \left\| y_k - \sum_{i=1}^{d_f} [h_{k,i} \mathbf{c}_{k,i}(m_i)] \right\|^2 \quad (1)$$

onde $m_i = 1, \dots, M$ e $k = 1, \dots, K$ em que $M = 4$ é o número de *codewords* em cada *codebook*, $K = 4$ é o número de subportadoras e d_f é o número de usuários que transmitem em uma mesma subportadora, y_k é a amostra do sinal recebido na k -ésima subportadora, $\mathbf{c}_{k,i}(m_i)$ é o possível símbolo transmitido pelo i -ésimo usuário na k -ésima subportadora, N_0 é a densidade espectral de ruído e \mathbf{H}_k é a matriz do canal, onde seus elementos $h_{k,i}$ representam o canal experimentado pelo i -ésimo usuário na k -ésima subportadora.

Após este cálculo inicial, o algoritmo Max-Log-MPA entra em seu processo iterativo, onde a cada iteração há a troca de informações entre os FNs e os VNs.

Inicialmente há a atualização da informação dos VNs, ou seja, um FN g passa a informação extrínseca para um VN obtida a partir de outros VNs [4]. O cálculo dessa informação é dado por:

$$I_{g \rightarrow v_1}^t(m_1) = \max_{m_2, m_3} \left[f_k(y_k, m_1, N_0, \mathbf{H}_k) + I_{v_2 \rightarrow g}^{t-1}(m_2) + I_{v_3 \rightarrow g}^{t-1}(m_3) \right], \quad (2)$$

onde foi considerado que $d_f = 3$ usuários transmitindo na k -ésima subportadora, $m_1 = 1, \dots, M$, $m_2 = 1, \dots, M$, $m_3 = 1, \dots, M$ e t é a iteração que está sendo calculada.

Considerando os cálculos realizados em (2), pode-se escrever a complexidade de $I_{g \rightarrow v}$ em termos de adições e subtrações como:

$$C_{I_{g \rightarrow v}} = (M^{d_f} K d_f (d_f + 1) - M K d_f) N \quad (3)$$

onde N é o número de iterações realizadas pelo algoritmo Max-Log-MPA. Note que a complexidade cresce exponencialmente com o número de usuários que transmitem em cada subportadora (d_f).

O próximo passo no processo iterativo é a atualização dos FNs. Neste caso, os VNs passam informação extrínseca obtida de um FN para outros FNs. O cálculo de $I_{v \rightarrow g}$ é dado por:

$$I_{v \rightarrow g_1}^t(m_i) = A_{p_v}(m) + I_{g \rightarrow v_2}^t(m) - \max(I_{g \rightarrow v_2}^{t-1}) \quad (4)$$

onde $A_{p_v}(m)$ é a probabilidade a priori do símbolo m e o cálculo é realizado para todo $n = 1, \dots, d_v$. Em termos de adições e subtrações, a complexidade de $I_{v \rightarrow g}$ é dada por:

$$C_{I_{v \rightarrow g}} = [(M + 1) V d_v M^{d_v - 1}] N. \quad (5)$$

Depois de N iterações são calculadas as *LLRs* para obter uma suposição de quais símbolos e, conseqüentemente, quais bits que foram transmitidos por cada usuário.

A maior complexidade do algoritmo Max-Log-MPA está na parte iterativa [3], e o número de cálculos nessa parte é dado pela soma de (3) e (5). Como exemplo numérico, podemos usar os parâmetros de um sistema em que a taxa de carregamento de canal é de 150%. A complexidade para cada iteração é de $C = 3.024 + 240 = 3.264$ adições/subtrações, onde $C_{I_{g \rightarrow v}} = 3.024$ e $C_{I_{v \rightarrow g}} = 240$. O cálculo de $I_{g \rightarrow v}$ corresponde à 92,65% das operações no processo iterativo. Sendo assim, o foco deste trabalho é na redução de complexidade de $I_{g \rightarrow v}$.

III. MÉTODO PARA REDUÇÃO DA COMPLEXIDADE DO ALGORITMO MAX-LOG-MPA

O método consiste em duas partes distintas. A primeira é baseada na comparação de $f_k(y_k, m_i, N_0, \mathbf{H}_k)$, calculado em (1), com um limiar para tomar uma decisão prévia ao processo iterativo. Esta comparação reduz a complexidade a partir da primeira iteração, porém a redução é a mesma para todas as iterações. A segunda parte faz a comparação entre os valores de $I_{g \rightarrow v}$ de duas iterações seguidas e não calcula mais

o valor de $I_{g \rightarrow v}$ quando os valores são próximos. Nesta parte, o método proposto tem que calcular ao menos duas iterações completas para poder fazer a comparação, mas, por outro lado, consegue uma redução maior da complexidade a cada iteração.

A. Parte I: Comparação de $f_k(y_k, m_i, N_0, \mathbf{H}_k)$ com Limiar

O cálculo de $f_k(y_k, m_i, N_0, \mathbf{H}_k)$ pode ser visto como uma probabilidade dos símbolos m_i ($i = 1, \dots, d_f$) terem sido transmitidos pelos d_f usuários na k -ésima subportadora. Sendo assim, se o valor de $f_k(y_k, m_i, N_0, \mathbf{H}_k)$ for muito pequeno, a chance dos d_f usuários terem transmitidos os m_i símbolos é muito pequena, portanto esses símbolos serão descartados nos cálculos seguintes.

No cálculo de $I_{g \rightarrow v}$ em (2) os valores de $f_k(y_k, m_i, N_0, \mathbf{H}_k)$ são considerados para todas as combinações de m_i , mas somente o valor máximo é escolhido como o valor de $I_{g \rightarrow v}$, o que é feito pela operação \max . Para simplificar o cálculo de (2), podemos desconsiderar todos os casos em que $f_k(y_k, m_i, N_0, \mathbf{H}_k)$ for muito pequeno. Isso significa que estamos descartando os casos em que a soma dentro do operador \max provavelmente não será a de maior valor, ou seja, os casos em que provavelmente os símbolos $\hat{c}_{k,i}(m)$ não foram os transmitidos.

Para classificar $f_k(y_k, m_i, N_0, \mathbf{H}_k)$ como pequeno ou não, definimos um limiar T_f . Assim, podemos adaptar o algoritmo Max-Log-MPA e propor o seguinte método para cálculo deste:

Algoritmo 1 Parte 1: Comparação de $f_k(y_k, m_i, N_0, \mathbf{H}_k)$ com Limiar.

- 1) Calcula o valor de $f_k(y_k, m_i, N_0, \mathbf{H}_k)$ usando (1) e vai para o processo iterativo.
 - 2) Compara o valor de $f_k(y_k, m_i, N_0, \mathbf{H}_k)$ com o limiar T_f .
 - 3) Se $f_k(y_k, m_i, N_0, \mathbf{H}_k) > -T_f$, atualiza o valor de $I_{g \rightarrow v_i}^t(m_i)$ usando (2). Caso contrário, não realiza nenhum cálculo.
 - 4) Atualiza o valor de $I_{v \rightarrow g_1}^t(m_i)$ usando (4).
 - 5) Após N iterações, calcula a LLR e toma uma decisão sobre os bits transmitidos pelos usuários.
-

B. Parte II: Comparação entre valores de $I_{g \rightarrow v}$

Em (2) é mostrado que o cálculo de $I_{g \rightarrow v}$ precisa ser feito a cada iteração para cada VN. A segunda parte do método consiste em calcular os valores de $I_{g \rightarrow v}$ de duas iterações e, caso os valores sejam próximos, nas iterações seguintes $I_{g \rightarrow v}$ não precisa mais ser calculado. Para classificar os valores de $I_{g \rightarrow v}$ como próximos, ou não, é calculado o desvio padrão entre os valores de duas iterações. Caso o desvio padrão seja menor que um limiar T_I , então o valor de $I_{g \rightarrow v}$ é mantido para as iterações seguintes. Caso sejam maiores que o limiar, então $I_{g \rightarrow v}$ ainda deve ser calculado. Temos que o Max-Log-MPA, considerando a segunda parte do método, fica assim:

O método serve também como critério de parada do cálculo das iterações, o que acontece quando todos os $I_{g \rightarrow v}$ convergem.

Algoritmo 2 Parte 2: Comparação entre valores de $I_{g \rightarrow v}$.

- 1) Calcula o valor de $f_k(y_k, m_i, N_0, \mathbf{H}_k)$ usando (1) e vai para o processo iterativo.
 - 2) Atualiza o valor de $I_{g \rightarrow v_i}^t(m_i)$ usando (2) e armazena os valores das duas primeiras iterações ($t = 1, 2$).
 - 3) Para $t \geq 3$, calcula o desvio padrão entre os valores das duas iterações anteriores: $std[I_{g \rightarrow v_i}^{t-1}(m_i), I_{g \rightarrow v_i}^{t-2}(m_i)]$.
 - 4) Compara o valor do desvio padrão com o limiar T_I .
 - 5) Se $std[I_{g \rightarrow v_i}^{t-1}(m_i), I_{g \rightarrow v_i}^{t-2}(m_i)] < T_I$, então $I_{g \rightarrow v_i}^t(m_i) = I_{g \rightarrow v_i}^{t-1}(m_i)$. Caso contrário, atualiza o valor de $I_{g \rightarrow v}^t$ usando (2).
 - 6) Atualiza o valor de $I_{v \rightarrow g_1}^t(m_i)$ usando (4).
 - 7) Após N iterações, calcula a LLR e toma uma decisão sobre os bits transmitidos pelos usuários.
-

IV. SIMULAÇÕES USANDO O MÉTODO PROPOSTO

Para demonstrar o funcionamento do método proposto para redução da complexidade do algoritmo Max-Log-MPA foram feitas simulações usando o método de Monte Carlo para o uplink de um sistema SCMA. Os resultados das simulações são curvas de desempenho, complexidade e tempo de simulação. O desempenho será mostrado em termos de taxa de erro de bit (BER). A complexidade é calculada através de simulação considerando as condições mostradas nos algoritmos (1) e (2). Esses valores são normalizados com relação à complexidade calculada em (3) para o algoritmo Max-Log-MPA original. Os tempos de simulação também são normalizados em relação ao Max-Log-MPA.

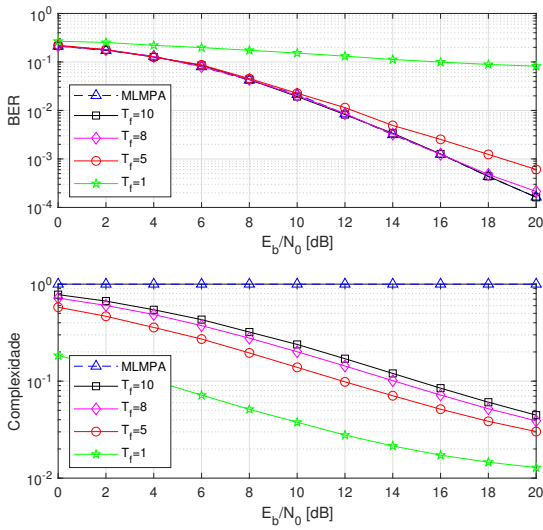
Nas simulações será considerado o número máximo de $N = 10$ iterações. Para os usuários, foi considerado o codebook apresentado em [12] e a transmissão dos símbolos se deu em um canal com desvanecimento Rayleigh sem o uso de código corretores de erro. Os resultados de simulação de cada parte do método são apresentados separadamente e depois será feita uma análise mais detalhada utilizando ambas as partes.

A. Resultados para a Parte I do Método

Usando somente a Parte I do método sabemos que a redução da complexidade não varia a cada iteração. Por outro lado, a complexidade é reduzida desde a primeira iteração. A Fig. 2 mostra o desempenho e a complexidade em função de E_b/N_0 para diferentes valores de limiar T_f . As curvas do Max-Log-MPA servirão como referência para avaliação do método proposto.

Podemos ver que à medida que o limiar diminui, menor é a complexidade, porém o seu desempenho piora. Existe um *tradeoff* entre desempenho e complexidade, e deve ser buscado um limiar que atinja a menor complexidade possível sem que haja perda de desempenho. Para os valores de E_b/N_0 mostrados na Fig. 2, o limiar $T_f = 10$ foi o que apresentou o desempenho mais próximo ao do Max-Log-MPA e com complexidade menor, variando entre 0,8, para $E_b/N_0 = 0$ dB, até 0,045, para $E_b/N_0 = 20$ dB.

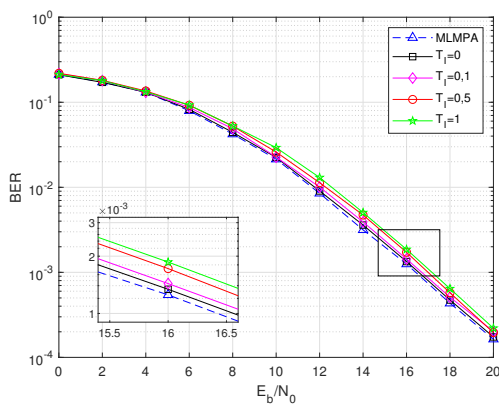
Com os parâmetros utilizados, a complexidade de $I_{g \rightarrow v}$ a cada iteração corresponde à 92,65% e o de $I_{v \rightarrow g}$ é 7,35%


 Fig. 2. Desempenho e Complexidade em Função de E_b/N_0 .

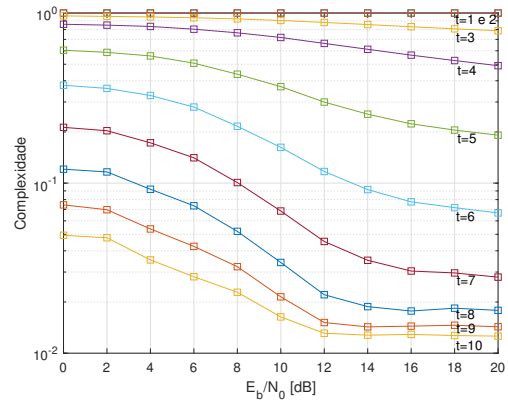
da complexidade. A Parte I do método consegue reduzir a complexidade de $I_{g \rightarrow v}$ em 92,8% para $E_b/N_0 = 16$ dB e $T_f = 8$. Isso significa reduzir a complexidade da parte iterativa do algoritmo Max-Log-MPA em 85,45%.

B. Resultados para a Parte II do Método

Para verificar o desempenho da segunda parte do método, podemos ver na Fig. 3 as curvas de BER em função de E_b/N_0 para diferentes limiares T_I . $T_I = 0$ é o caso que apresenta o melhor desempenho, porém há uma pequena perda de desempenho em comparação com o algoritmo Max-Log-MPA. Para $E_b/N_0 = 16$ dB, o desempenho para $T_I = 0$ é 0,15 dB pior. À medida que T_I aumenta, há uma maior perda de desempenho.


 Fig. 3. Desempenho em Função de E_b/N_0 .

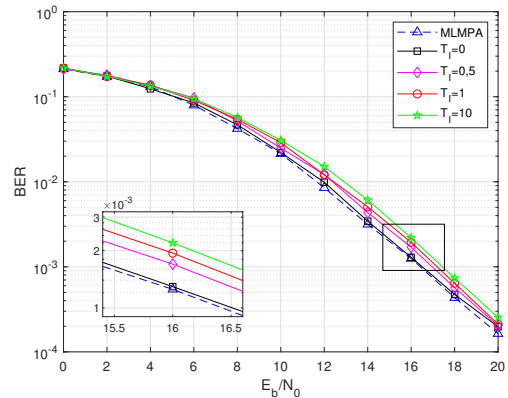
Por outro lado, quanto maior o valor do limiar T_I e da E_b/N_0 , menor é a complexidade, como mostra a Fig. 4. Isso acontece pois há uma menor distorção no sinal devido ao ruído e o algoritmo consegue fazer $I_{g \rightarrow v}$ convergir mais rápido.


 Fig. 4. Complexidade em Função de E_b/N_0 e $T_I = 0$.

C. Resultados Considerando as Duas Partes do Método

A primeira parte do método consegue uma redução na complexidade do Max-Log-MPA desde a primeira iteração, porém a redução é a mesma para todas as iterações. Já a segunda parte do método, consegue reduzir a complexidade cada vez mais a cada iteração, porém precisa calcular pelo menos duas iterações inteiras. Se aplicarmos as duas partes do método podemos reduzir a complexidade à partir da primeira iteração e reduzir ainda mais a cada iteração.

A Fig. 5 mostra curvas de BER em função E_b/N_0 para $T_f = 10$ e diferentes valores de T_I . Comparando as Figs. 3 e 5 podemos ver que há pouca diferença entre elas, já que eliminar alguns cálculos devido à classificação de $f_k(y_k, m_i, N_0, \mathbf{H}_k)$ não leva à perda de desempenho, como já mostrado na Fig. 2.


 Fig. 5. Desempenho em Função de E_b/N_0 para Diferentes Valores de T_I e $T_f = 10$.

Embora haja uma perda de desempenho de 0,1 dB em comparação com o Max-Log-MPA, para $T_f = 10$ e $T_I = 0$, a redução de complexidade é muito grande. Por exemplo, para $E_b/N_0 = 10$ dB, a complexidade normalizada nas duas primeiras iterações é de 0,24 e reduz para 0,0048 na iteração $t = 10$, como pode ser visto na Fig. 6. Esta figura mostra a complexidade e o tempo de simulação normalizados em função de E_b/N_0 para $T_f = 10$ e $T_I = 0$.

Em termos práticos, a redução do número de cálculos leva a uma redução, em igual proporção, do tempo de detecção do

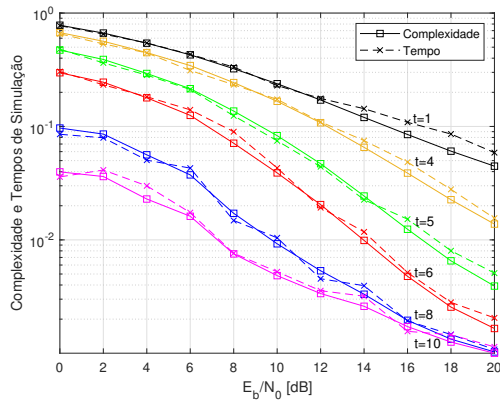


Fig. 6. Complexidade e Tempos de Simulação em Função de E_b/N_0 para $T_f = 10$ e $T_I = 0$.

algoritmo Max-Log-MPA. Através da Fig. 6 podemos ver que calcular a complexidade pelo número de operações matemáticas definido em (3) está correto e que o método proposto realmente reduz o tempo de decodificação do algoritmo Max-Log-MPA.

Para efeito de comparação, a Fig. 7 mostra a complexidade do método proposto em comparação com os apresentados nas referências citadas. Na figura foi utilizado $T_I = 0$ e T_f diferente para cada E_b/N_0 de modo a obter o desempenho do Max-Log-MPA mas com a menor complexidade possível. O método proposto neste trabalho tem complexidade menor que todos os outros citados para valores de E_b/N_0 mais altos.

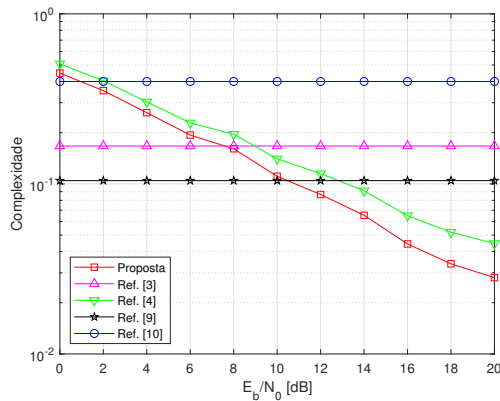


Fig. 7. Comparação da Complexidade para $N = 5$ Iterações.

V. CONCLUSÕES

Neste trabalho foi proposto um método para a redução da complexidade do algoritmo Max-Log-MPA, utilizado na decodificação de sistemas SCMA, considerando que a maior complexidade está dentro do processo iterativo e que o cálculo da variável $I_{g \rightarrow v}$ tem a complexidade predominante.

O método é dividido em duas partes, onde a primeira visa a redução da complexidade desde a primeira iteração, porém a redução se mantém a mesma para todas as iterações. A segunda parte consegue uma redução da complexidade somente a partir da terceira iteração, porém reduz cada vez mais

após cada iteração. Considerando as duas partes é possível aproveitar das qualidades de cada parte de maneira conjunta.

Através de simulações foi possível mostrar que o método proposto é simples e consegue reduzir inicialmente mais de 95% da complexidade do cálculo de $I_{g \rightarrow v}$ para valores altos de E_b/N_0 , o que representa uma redução de mais de 85% da complexidade de toda a parte iterativa do algoritmo Max-Log-MPA. Além disso, o método reduz ainda mais a complexidade a cada iteração, chegando a zero para iterações mais altas e altos valores de E_b/N_0 . Isso significa que o método funciona também como um critério de parada para o processo iterativo, uma vez que não realiza mais cálculos quando os valores de $I_{g \rightarrow v}$ convergem. Essa grande redução de complexidade foi obtida com perda de desempenho de aproximadamente 0,1 dB.

Foi mostrado também que a redução do número de cálculos levou a uma redução em igual proporção do tempo de simulação. Portanto, em sistemas práticos, os dispositivos que utilizarem esse método conseguirão reduzir o tempo para detecção dos bits e, conseqüentemente, demandarão menos energia e processamento que outro dispositivo que implementa o Max-Log-MPA original.

REFERÊNCIAS

- [1] H. Nikopour and H. Baligh, "Sparse code multiple access," 2013 IEEE 24th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC), London, 2013, pp. 332-336.
- [2] S. Zhang, X. Xu, L. Lu, Y. Wu, G. He and Y. Chen, "Sparse code multiple access: An energy efficient uplink approach for 5G wireless systems," 2014 IEEE Global Communications Conference, Austin, TX, 2014, pp. 4782-4787.
- [3] A. Bayeseh, H. Nikopour, M. Taherzadeh, H. Baligh and J. Ma, "Low Complexity Techniques for SCMA Detection". 2015 IEEE Globecom Workshops (GC Wkshps), San Diego, USA, 2015.
- [4] B. Min and J. Sun, "Message Passing Algorithm with Dynamic Thresholds in SCMA," 2019 International Conference on Communications, Information System and Computer Engineering (CISCE), Haikou, China, 2019, pp. 324-327, doi: 10.1109/CISCE.2019.00079.
- [5] R. Alizadeh and Y. Savaria, "Performance analysis of a reduced complexity SCMA decoder exploiting a low-complexity maximum-likelihood approximation," 2016 IEEE International Conference on Electronics, Circuits and Systems (ICECS), Monte Carlo, 2016, pp. 253-256.
- [6] A. Ghaffari, M. Leonardon, Y. Savaria, C. Jégo and C. Leroux, "Improving performance of SCMA MPA decoders using estimation of conditional probabilities," 2017 15th IEEE International New Circuits and Systems Conference (NEWCAS), Strasbourg, 2017, pp. 21-24.
- [7] M. Taherzadeh, H. Nikopour, A. Bayesteh and H. Baligh, "SCMA Codebook Design," 2014 IEEE 80th Vehicular Technology Conference (VTC2014-Fall), Vancouver, BC, 2014, pp. 1-5, doi: 10.1109/VTC-Fall.2014.6966170.
- [8] F. Wei and W. Chen, "Low Complexity Iterative Receiver Design for Sparse Code Multiple Access," in IEEE Transactions on Communications, vol. 65, no. 2, pp. 621-634, Feb. 2017.
- [9] K. Han, J. Hu, J. Chen and G. E. Sobelman, "A low complexity SCMA detector based on stochastic computation," 2017 IEEE 60th International Midwest Symposium on Circuits and Systems (MWSCAS), Boston, MA, 2017, pp. 783-786.
- [10] S. Hao, X. Su, J. Zeng, X. Ma and T. Lv, "A Low Complexity SCMA Detector Based on Avoiding Redundant Iterations," 2018 12th International Symposium on Medical Information and Communication Technology (ISMICT), Sydney, NSW, 2018, pp. 1-5.
- [11] Y. Du, B. Dong, Z. Chen, J. Fang, and X. Wang, "A Fast Convergence Multiuser Detection Scheme for Uplink SCMA Systems," IEEE Wireless Commun. Lett., vol. 5, no. 4, pp. 388-391, Aug. 2016.
- [12] Altera Innovate Asia website, Presentation 1st 5G Algorithm Innovate Competition-ENV1.0-SCMA, Web: <http://www.innovateasia.com/5g/en/gp2.html>.