

UMA REDE DE CODIFICAÇÃO/DECODIFICAÇÃO PARA CANAIS NÃO-COERENTES

Juliana F. Pedrosa e Jaime Portugheis

DECOM – FEEC – UNICAMP, CP. 6101, 13083-970 Campinas, SP - Brasil

e-mail: <jpedrosa, jaime>@decom.fee.unicamp.br

RESUMO

A decodificação iterativa para a combinação de modulação diferencial, entrelaçamento e codificação convolucional demonstrou melhorar o desempenho de sistemas de transmissão digital tanto em canais coerentes como não-coerentes. Este trabalho analisa a substituição do codificador convolucional por um codificador turbo, criando, assim, um cenário de decodificação que propicia a combinação de duas formas diferentes de iteração: uma devido ao próprio processo iterativo, e outra inerente à decodificação turbo. Uma modificação do algoritmo de detecção MAP para canais não-coerentes, utilizado na rede de decodificação, foi sugerida. Uma descrição detalhada da rede de decodificação, que combina as duas formas de iteração, é também desenvolvida. Resultados de simulação indicam que determinadas combinações das duas formas de iteração utilizadas da rede de decodificação possuem um compromisso melhor entre desempenho e complexidade de implementação.

1. INTRODUÇÃO

Hoeher e Lodge propuseram em [1] uma rede de decodificação iterativa para a concatenação serial de um codificador convolucional e um modulador diferencial PSK (DPSK) em *canais coerentes*. Neste trabalho, nós substituímos o codificador convolucional por um codificador turbo, criando, assim, um cenário de decodificação diferente daquele anterior, que propicia a utilização e combinação de duas formas diferentes de iteração: uma devido ao próprio processo iterativo, e outra inerente à decodificação turbo. O trabalho analisa como essas formas de iteração devem ser combinadas de maneira que possam favorecer o desempenho do sistema. Até onde vai o nosso conhecimento, esta é a primeira vez que uma combinação de duas formas diferentes de iteração na rede de decodificação é proposta e analisada. Além disto, pelo fato de termos considerado canais *não-coerentes*, houve a necessidade de sugerir uma modificação do algoritmo de detecção MAP (*Máximo a Posteriori*) utilizado na rede de decodificação.

Muito recentemente ocorreu uma explosão de interesse no projeto de sistemas concatenados de modulação diferencial com decodificação iterativa (veja [5] e suas inúmeras referências). Talvez uma das razões para isto resida no fato de que os mesmos tenham demonstrado melhor desempenho mesmo em canais coerentes quando comparados a sistemas sem modulação

diferencial de mesma complexidade [6]. Algumas modificações do algoritmo de detecção MAP para uso em canais não-coerentes foram propostas na literatura, entretanto, acreditamos que a modificação descrita neste artigo seja distinta destas propostas pelo fato de a termos sugerido de maneira independente. Uma manipulação das equações dos algoritmos poderia ser realizada como tentativa de se provar uma equivalência entre as propostas.

O artigo irá também explicar de maneira detalhada a rede de decodificação considerada, o que em geral não é feito na literatura. Vale mencionar o artigo tutorial [7] onde o termo rede de codificação/decodificação foi pela primeira vez utilizado.

O artigo está organizado da seguinte maneira: Na seção 2 é feita uma descrição da rede proposta através dos blocos constituintes. Em seguida, é analisada uma modificação no algoritmo de detecção MAP utilizado de forma a permitir o uso do demodulador proposto. Na seção 4, curvas de desempenho são obtidas permitindo, portanto, a análise completa da rede de codificação/decodificação e, por fim, são feitos comentários com algumas sugestões.

2. DESCRIÇÃO DA REDE

A rede completa de codificação/decodificação é mostrada na Fig. 5. Dois codificadores convolucionais recursivos de taxa $\frac{1}{2}$ e memória $M = 4$, de polinômios geradores 31 e 17, são concatenados em paralelo através de um entrelaçador (interleaver) interno formando um código turbo de taxa $\frac{1}{4}$. O entrelaçamento é gerado através de uma permutação pseudo-aleatória e o comprimento escolhido foi de 1000 bits de informação. Os bits codificados são entrelaçados pelo entrelaçador externo (pseudo-aleatório) de comprimento 1000 símbolos (onde cada símbolo corresponde a dois bits codificados) e são mapeados numa constelação 4-PSK, através do conversor, assumindo um dos quatro valores uniformemente distribuídos $\beta_m = 2\pi m/4$; $m = 0, \dots, 3$. Estes símbolos entrelaçados passam pelo modulador diferencial e são enviados através do canal.

Os símbolos recebidos do canal serão processados através de dois estágios: a) o demodulador diferencial b) o decodificador turbo.

Na recepção, ambos os estágios utilizam o algoritmo BCJR [2], algoritmo de detecção MAP, para a decodificação. Apenas, no demodulador, este algoritmo foi modificado no intuito de permitir uma recepção não-coerente. A idéia de se usar iteração, no sistema, tornou-se possível graças à natureza das entradas necessárias ao funcionamento do algoritmo: uma correspondendo à informação vinda do canal, e outra, dita

*Esta pesquisa teve o apoio parcial da CAPES.

informação *a priori*, responsável diretamente pela realimentação do processo. Desse modo, estando o algoritmo presente nos dois estágios, os resultados obtidos em um puderam ser utilizados como informação *a priori* no outro e vice-versa, concretizando, assim, o processo iterativo.

A rede foi desenhada como se fosse formada por duas ramificações, cada qual partindo da saída de um dos codificadores convolucionais. Esse arranjo foi apenas uma forma encontrada de facilitar a visualização do processo e, até mesmo, a programação e pode ser pensada dessa forma já que os símbolos codificados são gerados de maneira independente em cada codificador. Portanto, tudo se passa como se houvesse apenas um ramo de transmissão e recepção, ou seja, um modulador e um demodulador 4-DPSK.

2.1 Demodulador diferencial

O demodulador diferencial está associado ao modulador 4-DPSK, que pode ser pensado como um codificador de memória $M = 4$ representado por uma treliça com quatro estados. Ele irá processar os símbolos recebidos, $y_{1,i}$ e $y_{2,i}$, liberando probabilidades *a posteriori* (APP's) referentes a cada um dos símbolos 4-DPSK. As APP's fornecem estimações dos símbolos codificados (\hat{x}_1 e \hat{x}_2) relativas ao (super-) canal de entrada e saída discretas visto pelo decodificador turbo. Como informação *a priori* do algoritmo BCJR, são utilizadas quatro entradas, uma para cada símbolo 4-DPSK, alimentadas com dados vindos do decodificador turbo, exceto para a primeira utilização do demodulador, quando a cada uma das entradas é associada a probabilidade $1/4$ (inicialização).

2.2 Decodificador turbo

A decodificação turbo é realizada segundo o modelo descrito em [3]. No entanto, devido à mudança no cenário de decodificação – que agora compreende a ação conjunta de decodificador e demodulador num processo iterativo – novas variáveis são introduzidas de forma a se adequar ao contexto do receptor em questão. Os decodificadores componentes, como já mencionado, também fazem uso do algoritmo BCJR, mas na sua forma original. Cada decodificador componente utiliza, portanto, uma treliça binária referente ao codificador convolucional associado e que, neste caso, são idênticas pelo fato dos codificadores também serem idênticos.

A decodificação turbo é próxima da ótima, apesar de que cada algoritmo BCJR proporciona uma decodificação ótima. Isto acontece porque a regra de decodificação empregada utiliza as duas observações do canal, y_1 e y_2 , separadamente, isto é, cada decodificador calcula as APP's observando o y referente ao codificador associado. Isto é feito para evitar a complexidade embutida na regra ótima de observação conjunta.

2.3 Canal não-coerente

A transmissão dos sinais 4-DPSK se dá através do canal com desvanecimento *Rayleigh*. O i -ésimo sinal recebido na saída do filtro casado pode ser escrito como:

$$r_i = h_i s_i e^{j\theta} + n_i = y_{1i} + jy_{2i} \quad (1)$$

onde h_i é o ruído multiplicativo *Rayleigh*, n_i é uma amostra do ruído gaussiano branco complexo de média nula e variância $N_0/2$ e θ é a fase aleatória introduzida pelo canal que, na ausência de um modelo, foi assumida como sendo uniformemente distribuída no intervalo $(0, 2\pi)$. As amostras de ruído h_i foram consideradas independentes entre si e de n_i . Considerou-se também o conhecimento delas no receptor. Um fato a ser observado é que a modulação 4-DPSK necessita que a fase aleatória permaneça aproximadamente constante durante um intervalo de pelo menos dois símbolos. Porém, devido à própria estrutura de treliça usada no demodulador em questão, que trabalha com o conceito de “janela deslizante” (seção 3), foi imprescindível a utilização de uma fase aleatória “constante” em todo o bloco de símbolos codificados.

2.4 O super-canal

As entradas \hat{x}_1 e \hat{x}_2 dos decodificadores não poderiam ser alimentadas com dados vindo do canal, r , visto que há o bloco demodulador entre eles e não faria sentido algum a utilização desta informação após a demodulação. Assim, uma idéia foi simular um canal visto pelo decodificador turbo, denominado super-canal, mostrado na Fig. 1. Antes do sistema entrar em funcionamento, o super-canal foi simulado da seguinte maneira: Cada símbolo DPSK foi transmitido pelo canal um certo número de vezes e as probabilidades de transição de símbolo foram estimadas através das suas frequências relativas.

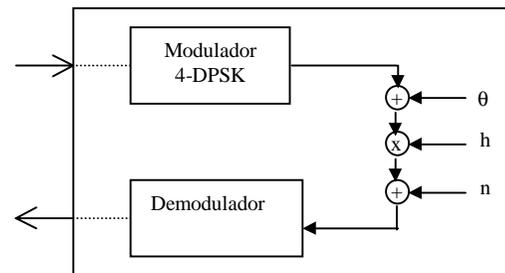


Figura 1. Supercanal visto pelo decodificador turbo. θ , h e n representam, respectivamente, a fase aleatória, o ruído multiplicativo *Rayleigh* e o ruído AWGN do canal não-coerente.

A cada simulação, a relação sinal-ruído (SNR) foi sendo variada e tabelas de probabilidades de transição foram sendo obtidas. Dessa forma, quando a rede foi posteriormente simulada, à medida que o algoritmo BCJR, nos decodificadores, requeria informação de canal, a tabela correspondente à SNR utilizada era acionada e a probabilidade de transição para os símbolos x_i e \hat{x}_i era obtida.

2.5 Informação *a priori*

Outro problema encontrado foi como alimentar as entradas relativas à informação *a priori*, no 1º decodificador (chave na Fig. 5), utilizando resultados provenientes do demodulador, uma vez que o primeiro necessita de informações binárias e o segundo apenas fornece informação relativa a símbolos 4-DPSK. A idéia foi fazer uso do fato dos codificadores serem

sistemáticos, ou seja, a entrada u equivale à própria saída $x1i$, e a entrada $ulinha$ (u entrelaçada), à saída $x2i$.

Dessa forma, com o auxílio da Tab. 1, a relação direta entre os bits de informação (u) e os símbolos codificados ($x\omega i$, $x\omega p$) é facilmente percebida:

- Um bit de informação 0 (-1) corresponde ao símbolo codificado π ou $\pi/2$;
- Um bit de informação 1 corresponde ao símbolo codificado 0 ou $3\pi/2$.

Bits codificados		Símbolo 4-DPSK
$x\omega i$	$x\omega p$	
0	0	π
0	1	$\pi/2$
1	0	$3\pi/2$
1	1	0

Tabela 1. Associação de símbolos utilizada na modulação. No texto, “ ω ” = 1, 2 (representação simbólica da saída de um dos codificadores convolucionais)

Recordando que o decodificador turbo tem como objetivo a decodificação dos bits de informação u , a convenção adotada é a de que a decisão final será tomada em relação ao bit 0. Portanto, a entrada relativa à informação $a priori$, em ambos os decodificadores turbo são assumidas como sendo:

$$eta_int = P[u = 0 | \hat{u}] = 1 - P[u = 1 | \hat{u}] \quad (2)$$

onde \hat{u} é a estimação do bit de informação.

Para o 1° decodificador, essa informação $a priori$ é obtida utilizando a tabela descrita acima. Portanto, as APP's obtidas do 1° demodulador relativas aos símbolos π e $\pi/2$ são combinadas devidamente de forma a resultar em:

$$eta_int = eta_pi + eta_pi/2 \quad (3)$$

A informação $a priori$ para o 2° decodificador vem da saída do primeiro.

2.6 Realimentação externa

É sabido que um princípio fundamental da decodificação iterativa é não realimentar um decodificador com informação que tenha sido originada dele próprio. Esta recomendação é literalmente seguida no processamento interno do turbo, onde, a cada realimentação de um decodificador para o outro, a saída produzida pelo “realimentador” é subtraída da informação $a priori$ introduzida nele (informação esta que provem justamente do decodificador que está para ser realimentado). Essa extração de informação redundante é denominada informação extrínseca.

Esse princípio se aplica bem ao turbo porque, mesmo que os dois decodificadores presentes estejam associados a codificadores distintos, eles fornecem informação sobre a mesma variável: os bits de informação u , que são os mesmos na entrada de ambos os codificadores a menos de um entrelaçamento para o segundo. Entretanto, no caso da realimentação externa demodulador/decodificador, o fato de se trabalhar simultaneamente com símbolos e bits gera algumas

dificuldades. O problema é que, dentro do decodificador turbo, não há como “passar” informação extrínseca relativa a símbolos, de um decodificador para o outro, já que os símbolos associados a um decodificador em nada se assemelham àqueles associados ao outro pois foram gerados independentemente nos dois codificadores convolucionais. Dessa forma, conclui-se que não há como extrair informação extrínseca de símbolo do segundo decodificador, o que compromete a passagem deste tipo de informação na realimentação externa (saída L_3 da Fig. 5).

Esses aspectos foram testados e, como previsto, não resultaram em ganho para o sistema tendo sido, portanto, descartados. A única exceção ocorre na realimentação do demodulador 1 para o 1° decodificador pois, neste caso, as saídas relativas aos símbolos, deste demodulador, que são combinadas para originar a informação $a priori$ para o dito decodificador, refletem-se mais diretamente na saída desse último e, então, é gerada uma informação extrínseca que será convertida em informação $a priori$ para ser utilizada no próximo acionamento do decodificador turbo, ou seja, na próxima iteração interna.

2.7 Descrição da rede de decodificação

Visando facilitar as operações realizadas, os decodificadores 1 e 2 e também o demodulador fornecem, por fim, como saída, razões log -likelihood (RLL). Para os bits de informação u , as probabilidades $a posteriori$ são convertidas tal qual em [3] para:

$$L = \log \frac{P[u = 1 | y]}{P[u = 0 | y]} \quad (4)$$

onde a probabilidade do bit 0 é tida como referência. Para os símbolos codificados, nas saídas dos decodificadores e na saída do demodulador, a fórmula foi adaptada de modo que a APP correspondente ao símbolo $3\pi/2$ fosse utilizada como referência:

$$L_X = \log \frac{P[X | y]}{P[X_{ref} = 3\pi/2 | y]} \quad (5)$$

onde X equivale aos símbolos 0, $\pi/2$ ou π e y é a observação do canal.

Durante as iterações internas do decodificador turbo, as operações usuais utilizando variáveis binárias são utilizadas. Entretanto é importante ressaltar que a realimentação externa obedece a seqüência de origem dos bits e/ou símbolos: o 1° (2°) decodificador está associado ao 1° (2°) codificador e, portanto, além de decodificar os bits codificados por este último, age no intuito de refinar as informações referentes aos símbolos diretamente associados a esses bits, ou seja, também se refere ao 1° (2°) demodulador.

Em síntese, a rede de decodificação funciona da seguinte maneira: a informação ruidosa do canal alimenta uma das entradas dos demoduladores. A outra, relativa à informação $a priori$, é alimentada tal qual sugerido em (2.1) uma vez que ainda não há informação proveniente dos decodificadores. Os blocos demoduladores liberam as APP's relativas aos símbolos que são convertidas em RLL's. Na interface entre os dois estágios, as RLL's são desentrelaçadas, pelo *deinterleaver* externo, originando \tilde{L}_{2-x} de maneira similar à eq. (5), apenas

utilizando a estimação dos símbolos codificados, \hat{x} , ao invés da observação do canal y . Estas novas razões sofrem uma transformação inversa e dão origem a uma espécie de informação *a priori* de símbolo,

$$eta_X = P[X | \hat{x}] = e^{L-x} \cdot eta_Xref \quad (6)$$

que são então combinadas, como na eq. (3), para formar a informação *a priori* de bit para o 1º decodificador. A outra entrada, referente à informação de canal é alimentada com uma das entradas da tabela do super-canal. O início das iterações internas ocorre neste momento, quando o turbo é acionado. O 1º decodificador, por sua vez, libera a RLL relativa a bit, $L_{2,u}$, que é então subtraída da RLL de bit inicial, $\tilde{L}_{2,u}$ obtida de eta_int através da eq. (4), formando, portanto, a informação extrínseca que, desentrelaçada pelo *deinterleaver* interno, dá origem à informação *a priori*, eta_int , para o 2º decodificador. Após o processamento, o dito decodificador libera a RLL $L_{3,u}$, que, subtraída de $\tilde{L}_{3,u}$, dá origem a $\tilde{L}_{2,u}$ que forma novamente a informação *a priori*, eta_int , para o 1º decodificador de forma a refinar os valores anteriormente obtidos. Eis então a primeira iteração interna. Tendo concluído a quantidade de iterações internas previamente estabelecidas, o turbo libera as RLL's $L_{2,X}$ e $L_{3,X}$, relativa aos símbolos, de forma a dar continuidade ao processo iterativo. Estas razões são, dessa forma, entrelaçadas pelo *interleaver* externo, e transformadas em informação *a priori*, eta_X , para os demoduladores, como na eq. (6). Completou-se, então, uma iteração externa. Este processo é repetido, e as informações vão se refinando à medida que passam de um estágio para o outro, até que seja findado o número previamente estabelecido de iterações externas. Neste momento, a demodulação/decodificação é finalizada e o 2º decodificador do turbo fornece as probabilidades *a posteriori* dos bits de informação u , que são desentrelaçadas pelo *deinterleaver* interno, e tem-se a decisão final, em relação a um limiar pré-estabelecido.

Uma observação a ser feita é que, na interface entre os dois estágios, as RLL's $L_{2,X}$ entrelaçadas, $\tilde{L}_{1,x}$, não contribuem para formar a informação extrínseca, $L_{1,x} - \tilde{L}_{1,x}$, uma vez que ainda não há informação vinda do turbo já que, nesse momento, ele ainda não foi utilizado. Esta informação só será relevante a partir da segunda iteração externa.

3. ALGORITMO BCJR MODIFICADO

O algoritmo utilizado no bloco demodulador é o algoritmo BCJR introduzido em [2]. Apenas, foi necessária uma pequena modificação de forma que pudesse ser levada em consideração a fase aleatória do canal, θ . Neste novo cenário, a equação sugerida para o cálculo das probabilidades de transição de canal (utilizadas na obtenção das matrizes gama) não se aplicava ao caso atual e teve que ser substituída por outra encontrada na literatura que trata justamente da detecção de símbolos DPSK (eq.9, [4]), com uma pequena adaptação para que se considerasse, também, o desvanecimento *Rayleigh*.

A equação é dada por:

$$p(r|h,s) = \frac{1}{(2\pi\sigma_0^2)^L} \exp\left[-\frac{1}{2\sigma_0^2} \sum_{l=1}^L (|r_l|^2 + |h_l s_l|^2)\right] \times I_0\left(\frac{1}{\sigma_0^2} \sum_{l=1}^L r_l h_l s_l^*\right) \approx I_0\left(\frac{1}{\sigma_0^2} \sum_{l=1}^L r_l h_l s_l^*\right) \quad (7)$$

onde r e s são, respectivamente, os vetores recebido e transmitido pelo canal, σ_0^2 é a variância do AWGN, $No/2$, e I_0 é a função de *Bessel* modificada de ordem 0 do primeiro tipo. O termo multiplicativo pode ser desconsiderado já que, para o MPSK, $|s_l|^2$ é constante para todas as fases resultando em informação irrelevante à decodificação. Em nossos cálculos, foi utilizado $L = 2$, que representa a necessidade, na decodificação diferencial, de espera de 2 símbolos tal qual foi feito na codificação. Portanto, as probabilidades de transição de canal passaram a ser calculadas a cada 2 símbolos e não mais a cada símbolo apenas, como era feito no desenvolvimento do algoritmo original. Esse fato resultou na necessidade de uma pequena modificação no algoritmo (eq. 9, [2]) justamente nas probabilidades de transição de canal:

$$\gamma_i(m', m) = \sum_{X_1, X_2} P\{X_t = X_2, X_{t-1} = X_1 | S_{t-1} = m', S_t = m\} P\{S_t = m | S_{t-1} = m'\} P\{Y_t, Y_{t-1} | X_t = X_2, X_{t-1} = X_1\} = q_t(X_1, X_2 | m', m) p_t(m', m) R(Y_1, Y_2 | X_1, X_2) \quad (8)$$

onde $R(\cdot)$ é dado, agora, pela eq. (7) com $y_i = r_i$ e $x_i = s_i$, $i = 1, 2$. Isto se reflete no processamento da treliça, por isso, passou-se a utilizar o conceito de “janela deslizante” de comprimento dois, ou seja, os cálculos não seriam mais feitos a cada estágio de tempo da treliça e sim a cada dois estágios. No próximo cálculo, a janela seria deslocada de um estágio fazendo com que sempre houvesse sobreposição de um , mantendo, assim, a conformidade com o codificador diferencial do modulador, que exigia que os dois símbolos codificados fossem adjacentes tornando-os estreitamente ligados entre si, e propiciando uma demodulação correta. A Fig. 2 é uma tentativa de visualização desse conceito, onde a variável y representa os símbolos provenientes do canal.

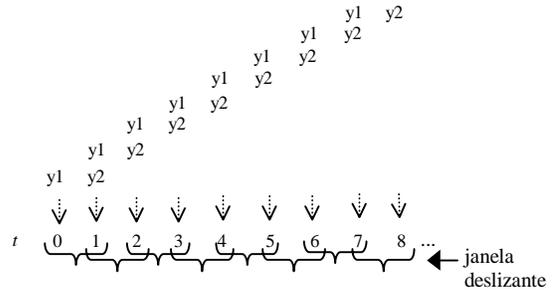


Figura 2. Representação simbólica da utilização da janela deslizante na treliça. y_1 e y_2 são, respectivamente, o símbolo anterior e o atual.

Logo, a cada cálculo de α (ou β), como em [2], o algoritmo continuará se deslocando por um estágio da treliça, apenas terá que observar dois símbolos, o atual e o anterior. No próximo deslocamento, o que era “anterior” passa a ser “atual” e os

cálculos em t ainda continuarão dependendo daquelas efetuados em $t-1$. O processo continua até que toda a treliça tenha sido “visitada” pelo algoritmo, tanto em um sentido quanto no outro, tal qual era feito no processo original.

Na decodificação turbo, o algoritmo BCJR utilizado é exatamente o original [2], apenas foi programado para fornecer, também, probabilidades *a posteriori* dos símbolos no momento da realimentação externa.

4. RESULTADOS DE SIMULAÇÃO

As Fig. 3 e 4 mostram resultados de simulação para a probabilidade de erro de bit versus E_b/N_o . Nas figuras, $sExI$ representa a simulação com E iterações externas e I iterações internas, onde uma iteração significa uma passagem pelo demodulador ou decodificador turbo. Na Fig. 3, o desempenho da modulação 4-DPSK, com demodulação MAP (utilizando o algoritmo BCJR), foi também incluído e demonstra ser ligeiramente superior ao desempenho com demodulação diferencial convencional (não mostrado na figura). Para $E = 1$, quatro iterações internas são suficientes para atingir o melhor desempenho. Mantendo a mesma complexidade de decodificação, fizemos $E = I = 2$ e observamos uma degradação de desempenho de aproximadamente 0,6 dB para $P_b \approx 10^{-4}$. Fizemos, então, $E=2, I=4$ e um pequeno ganho adicional (relativo a $E = 1, I = 4$) foi conseguido. Esta combinação de iterações mostrou-se ser a mais eficiente dentre as testadas.

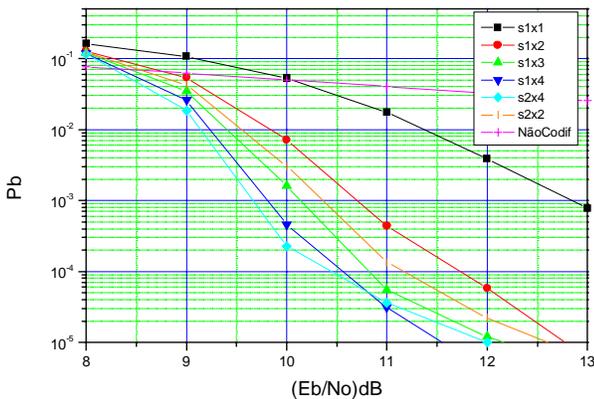


Figura 3. Curvas de desempenho para várias combinações de iterações externas e internas.

A Fig.4 mostra um exemplo do efeito da realimentação externa no desempenho da rede para $I = 2$. Nota-se que o aumento de E não contribui de maneira significativa sendo, portanto, a realimentação interna o ponto de maior importância (como observado na Fig.3). Este fato pode ter relação com a dificuldade encontrada, descrita na seção 2.6, na passagem da informação extrínseca para o demodulador .

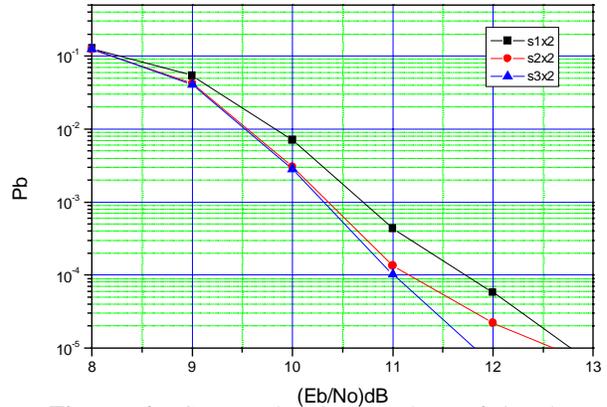


Figura 4. Curvas de desempenho enfatizando a realimentação externa.

5. COMENTÁRIOS FINAIS

Considerando um mesmo modelo de canal (não-coerente, desvanecimento Rayleigh) e complexidade de decodificação similar, não foi possível comparar os valores de E_b/N_o mínimos, atingidos pela rede de decodificação proposta, com resultados da literatura para um sistema com mesma eficiência espectral. Tal comparação ficaria para um artigo posterior.

É bem provável que modificações na passagem da informação extrínseca do decodificador para o demodulador (realimentação externa) levem a uma substancial melhoria no desempenho da rede para um número total de iterações fixo. Estudos vêm sendo desenvolvidos neste sentido.

6. REFERÊNCIAS

- [1] P. Hoeher e J. Lodge, ““Turbo DPSK”: Iterative Differential PSK Demodulation and Channel Decoding”, IEEE Transactions on Communications, vol. 47, No. 6, pp. 837-843, Junho 1999.
- [2] L. R. Bahl, J. Cocke, F. Jelinek e J. Raviv, “Optimal Decoding of Linear Codes for Minimizing Symbol Error Rate”, IEEE Transactions on Information Theory, pp. 284-287, Março 1974.
- [3] D. Divsalar e F. Pollara, “Turbo Codes for Deep-Space Communications”, TDA Progress Report 42-120, JPL, Fevereiro 1995.
- [4] D. Divsalar e M. Simon, “Multiple-Symbol Differential Detection of MPSK”, IEEE Transactions on Communications, vol. 38, No. 3, pp. 300-308, Março 1990.
- [5] G. Covalope, G. Ferrari e R. Raheli, “Noncoherent Iterative (Turbo) Decoding”, IEEE Transactions on Communications, vol. 48, No. 9, pp. 1488-1498, Setembro 2000.
- [6] I. Marsland e P. Mathiopoulos, “On the Performance of Iterative Noncoherent Detection of Coded M-PSK Signals”, IEEE Transactions on Communications, vol. 48, No. 4, pp. 588-596, Abril 2000.
- [7] S. Benedetto, G. Montorsi, D. Divsalar e F. Pollara, “Soft-Input Soft-Output Modules for the Construction and Distributed Iterative Decoding of Code Networks”, European Transactions on Telecommunications, vol. 9, No. 2, Março-Abril 1998.

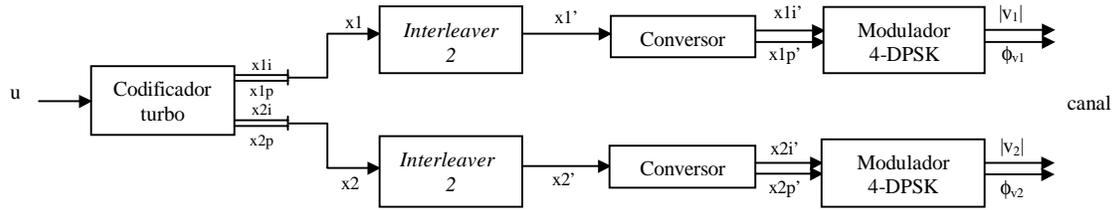


Figura 5a. Arranjo do transmissor

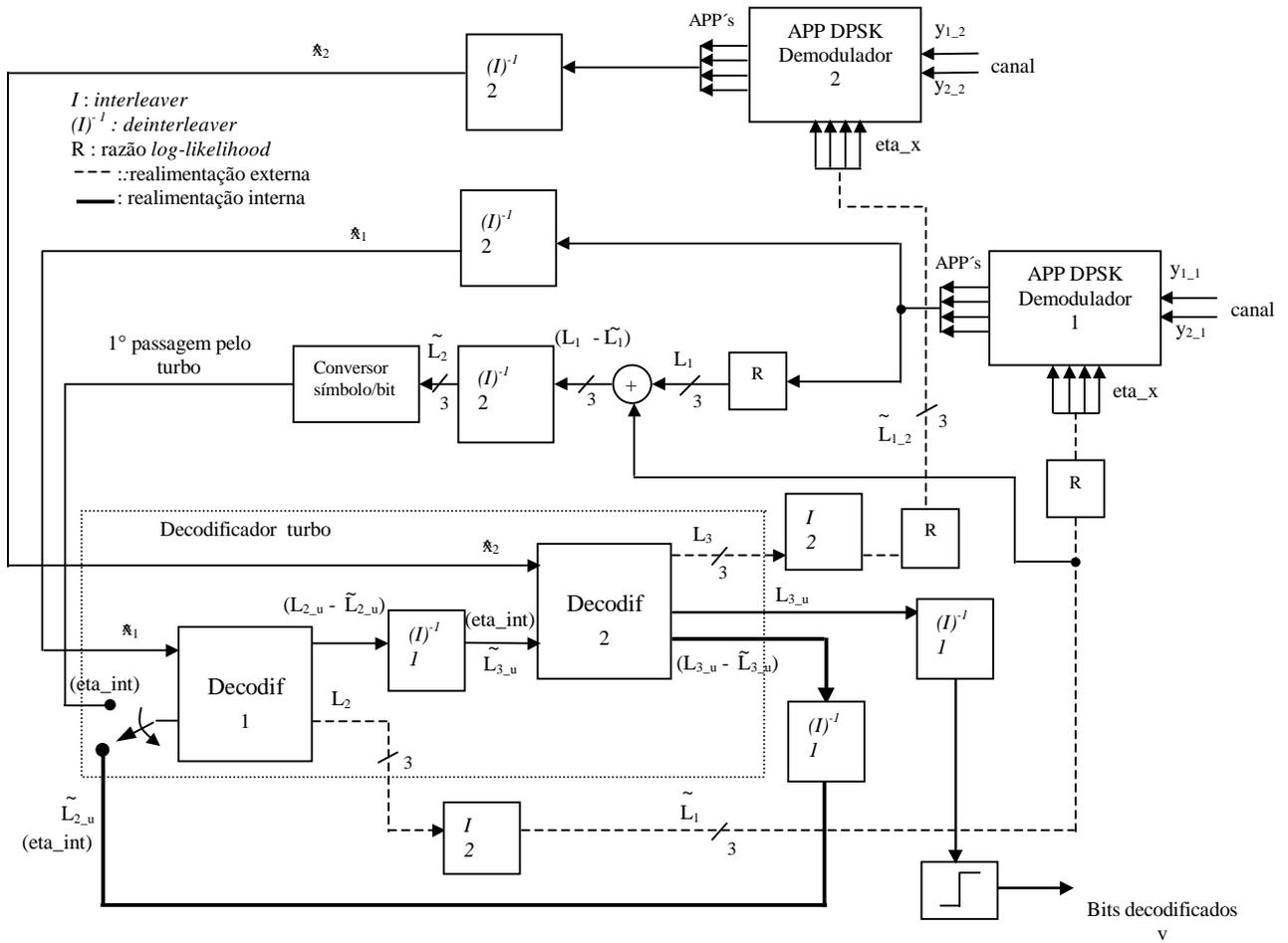


Figura 5b . Arranjo do receptor