

## 4

### Resultados

Neste capítulo encontram-se os resultados mais relevantes obtidos ao longo do estudo, bem como as suas análises. Todos os gráficos apresentados a seguir são resultados de simulações feitas em programa desenvolvido em linguagem Matlab, com o intuito de avaliar o desempenho das diversas variações sobre os códigos LT abordadas nesta tese.

Os resultados estão divididos em duas partes principais. Primeiramente, na Seção 4.1, estão os resultados obtidos com as variações dos códigos LT bidimensionais, propostos em [Paiba08]. Em seguida, na Seção 4.2, são mostrados os resultados obtidos com as variações propostas a partir dos códigos LT convencionais, ou seja, aplicados em blocos (vetores) e não em matrizes. Nesta parte, estão inclusos os resultados dos códigos LT Sistemáticos Alongados (LTSA), que são a principal contribuição científica desta tese.

Todas as simulações foram obtidas para canal BEC, sendo os resultados uma média de 500 transmissões (simulações) realizadas para cada ponto apresentado nos gráficos. Os blocos de informação têm tamanho  $k$  símbolos nos casos unidimensionais e, nos casos bidimensionais, os símbolos são agrupados em uma matriz LT, conforme explicado na Seção 2.5. Por simplificação, um símbolo foi composto por apenas um *bit*. Os parâmetros utilizados nas simulações foram:  $c = 0.03$ ,  $\rho = 0.1$ , conforme obtidos em [Paiba08]. O valor de  $\gamma$  depende do tamanho de  $k$ . A Tabela 4.1 mostra os valores de  $\gamma$  utilizados nesta tese, de acordo com os diversos valores de  $k$  presentes nos gráficos das seções a seguir. Tais valores são obtidos obedecendo-se a  $d_{max} = \frac{\gamma^k}{R}$ , como visto na Subseção 2.3.3.

#### 4.1

##### Códigos LT Bidimensionais

Nesta seção, estão apresentados – e comentados – os principais resultados das simulações referentes às variações dos códigos LT bidimensionais [Paiba08] estudadas.

Tabela 4.1: Valores de  $\gamma$  em função de  $k$ .

$k$	$\gamma$
125	2
167	2
250	3
500	5
1000	8
2000	13
3000	16
4000	20
5000	22
10000	34

Em [Paiba08], utilizou-se, para os casos bidimensionais, uma matriz LT de dimensões  $(1000 \times 1000)$  *bits* e esta matriz foi comparada ao método tradicional (em uma dimensão) com 1000 blocos de 1000 *bits*. Os resultados apresentados na Figura 4.1 à Figura 4.4 a seguir, foram obtidos através de simulações de códigos LT bidimensionais formados por um código  $LT_H$  sistemático (SLT) e por um código  $LT_V$  convencional, ou seja, não-sistemático (nSLT). Esta configuração foi proposta e estudada em [Paiba08] e será referenciada por SLT-nSLT — esta nomenclatura se refere, de forma padrão, a aplicação da codificação primeiramente às linhas da matriz ( $LT_H$ ) e, a seguir, a aplicação da codificação às colunas ( $LT_V$ ). É importante deixar claro que, em todas as simulações a seguir, os códigos SLT têm a distribuição de graus Sóliton Robusta Melhorada Truncada (DSRMT) e os nSLT têm a distribuição Sóliton Robusta Melhorada (DSRM).

Como objetivo de analisar o desempenho de outras configurações da matriz LT, além da  $(1000 \times 1000)$  de [Paiba08], a Figura 4.1 mostra a taxa de apagamento de símbolo (TAS) do sistema em função da qualidade do canal  $(1 - P_a)$ , para diversas matrizes LT. Foi utilizado um *overhead* total fixo de valor  $\epsilon = 0.30$ , resultante dos *overheads*  $\epsilon_H = 0.05$  e  $\epsilon_V = 0.24$ . É importante salientar que o cálculo da TAS implica que o decodificador opera com o *overhead* inicial total fixo, ou seja, não há transmissão de símbolos adicionais.

As configurações das quatro diferentes matrizes LT comparadas no gráfico da Figura 4.1 foram escolhidas de forma que o número de símbolos — *bits* — de cada uma delas seja, aproximadamente, o mesmo. Desta forma, a comparação torna-se justa. Percebe-se que, à medida em que o número de linhas aumenta

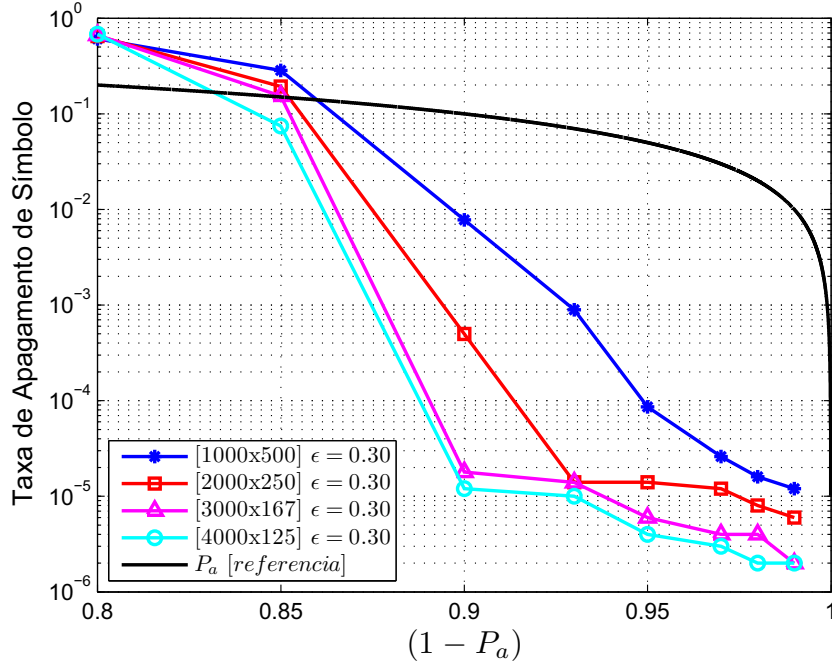


Figura 4.1: Comparação da TAS entre esquemas SLT-nSLT para diversas matrizes LT, com  $\epsilon_H = 0.05$  e  $\epsilon_V = 0.24$ .

e, conseqüentemente, o número de colunas diminui, o desempenho torna-se melhor (a TAS diminui). O aumento do número de linhas faz com que o código  $LT_V$  atue em um maior bloco de informações, melhorando seu desempenho. Por outro lado, a diminuição do número de colunas faz com que o código  $LT_H$  atue sobre um bloco de informações mais curto, diminuindo seu desempenho.

A Figura 4.2 apresenta as mesmas curvas da Figura 4.1, com a diferença de que o número de linhas e colunas das matrizes LT foram invertidos em relação à Figura 4.1. Nota-se uma considerável piora no desempenho dos esquemas na Figura 4.2. Além disso, a inversão do número de linhas e colunas acarretou também uma inversão no desempenho dos esquemas. Na Figura 4.1, o esquema com melhor desempenho é o da matriz  $(4000 \times 125)$  e o com pior é o da matriz  $(1000 \times 500)$ . Já na Figura 4.2, a matriz que apresenta menor TAS é a  $(500 \times 1000)$  e a de maior TAS é a  $(125 \times 4000)$ .

Analisando os resultados das Figuras 4.1 e 4.2 nota-se um primeiro indício de que a codificação horizontal nos esquemas bidimensionais, da forma que estão sendo abordados, não é eficiente. Em ambas as figuras, os esquemas de melhor desempenho foram aqueles cujo número de linhas — tamanho do bloco vertical, no qual o  $LT_V$  atua — é o maior possível e o número de colunas — tamanho do bloco horizontal, no qual o  $LT_H$  atua — é o menor possível.

Observando-se o caso particular das matrizes LT [4000x125] na Figura 4.1 e (125 × 4000) na Figura 4.2, percebe-se que: (a) quando o tamanho do bloco vertical é bastante grande ( $k_V = 4000$ , na Figura 4.1), o desempenho do esquema é bom e, por outro lado, (b) quando o tamanho do bloco horizontal é bastante grande ( $k_H = 4000$ , na Figura 4.2), o desempenho do esquema é ruim.

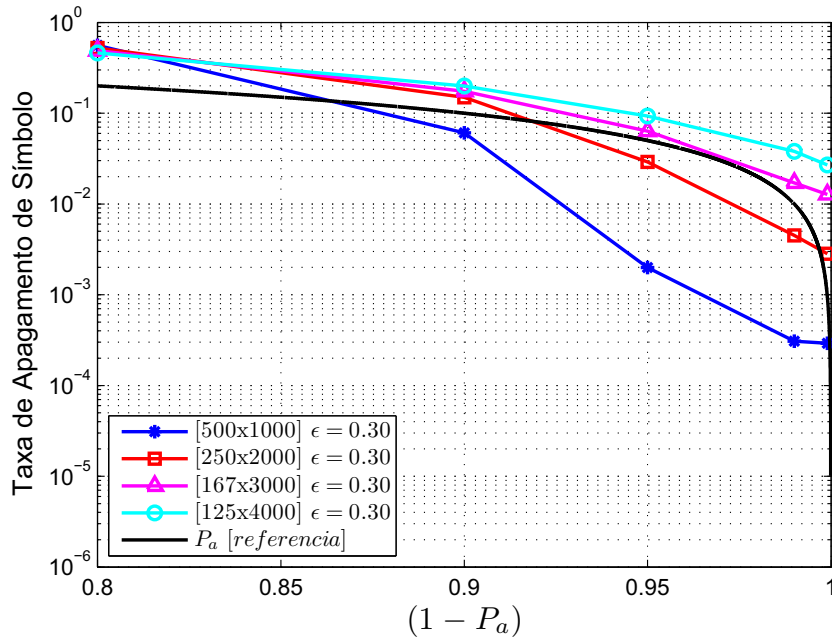


Figura 4.2: Comparação da TAS entre esquemas SLT-nSLT para diversas matrizes LT, com  $\epsilon_H = 0.05$  e  $\epsilon_V = 0.24$ .

Nas Figuras 4.1 e 4.2 analisou-se diferentes configurações de matrizes LT, porém todas com os mesmos valores de *overhead* ( $\epsilon_H = 5\%$  e  $\epsilon_V = 24\%$ ). A Figura 4.3 mostra o *overhead* total requerido ( $1 + \epsilon_{REQ}$ ) para uma correta decodificação da informação em diversas configurações dos *overheads*  $\epsilon_H$  e  $\epsilon_V$ . A matriz LT [1000x500] é utilizada. Observa-se que o *overhead* praticamente não se altera para valores de  $P_a \leq 10^{-2}$ . Isto se deve ao fato de que, nesses casos, o canal apresenta boa qualidade e não há a necessidade de muitos símbolos adicionais para que a decodificação tenha sucesso. O valor de ( $1 + \epsilon_{REQ}$ ) nesses casos é só um pouco maior que o valor inicial. A partir de  $P_a = 10^{-2}$ , a pior qualidade do canal requer que mais símbolos adicionais sejam transmitidos, fazendo aumentar assim o *overhead* requerido ( $1 + \epsilon_{REQ}$ ). Entretanto, é perceptível nas curvas cheias, onde o *overhead* horizontal inicial é fixado em  $\epsilon_H = 3\%$  e o  $\epsilon_V$  inicial varia entre 0%, 10% e 20%, que o valor de ( $1 + \epsilon_{REQ}$ ) tem pouca variação para  $P_a = 10\%$  e  $P_a = 20\%$ . Ou seja, mesmo aumentando o  $\epsilon_V$

inicial em 20%, observa-se apenas uma pequena variação ( $< 5\%$ ) em  $(1 + \epsilon_{REQ})$ . Conclui-se então que, neste caso, não é necessário um alto valor inicial de  $\epsilon_V$  para uma decodificação satisfatória.

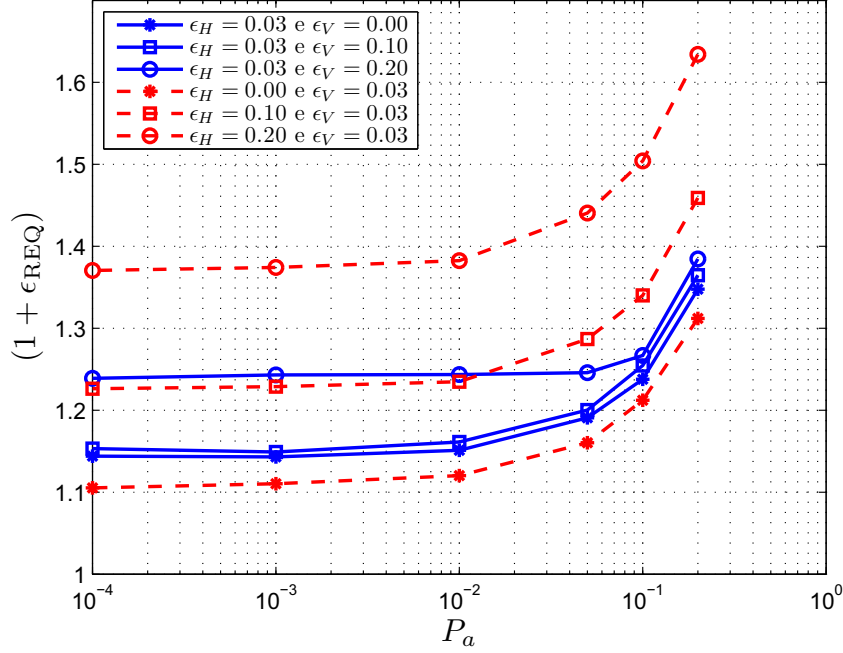


Figura 4.3: Comparação do *overhead* total requerido entre esquemas SLT-nSLT com diversas combinações de  $\epsilon_H$  e  $\epsilon_V$ . Matriz LT:  $(1000 \times 500)$ .

Na outra situação mostrada na Figura 4.3, o valor inicial do *overhead* vertical é fixado em  $\epsilon_V = 3\%$  e o  $\epsilon_H$  inicial varia entre 0%, 10% e 20%. Ao contrário do caso anterior, nesta situação há uma variação de mais de 30% no valor de  $(1 + \epsilon_{REQ})$ , quando se aumenta o  $\epsilon_H$  inicial em 20%. A conclusão deste fato é que o aumento do  $\epsilon_H$  não traz benefícios ao código bidimensional, chegando até a prejudicar seu desempenho.

Na Seção 3.1, abordou-se a obtenção das distribuições de graus para tamanhos de blocos diferentes de  $k$ . Ou seja, obtém-se a distribuição para blocos de tamanho  $(x \cdot k)$ , com  $x$  inteiro maior que um, e a utiliza-se em blocos de informações de tamanho  $k$ . A Figura 4.4 traz alguns resultados a esse respeito, onde foi utilizada a configuração SLT-nSLT com matriz LT  $[1000 \times 1000]$  e as DG's foram obtidas para valores de  $k = 1000, 2000, 5000$  e  $10000$ .

As curvas com distribuições de graus modificadas para  $k = 5000$  e  $10000$  têm a taxa de apagamento de símbolo bastante elevada em relação à curva com a DG apropriada (para  $k = 1000$ , visto que as matrizes LT utilizadas foram  $(1000 \times 1000)$ ). Já a curva com DG para  $k = 2000$  apresenta uma leve melhora

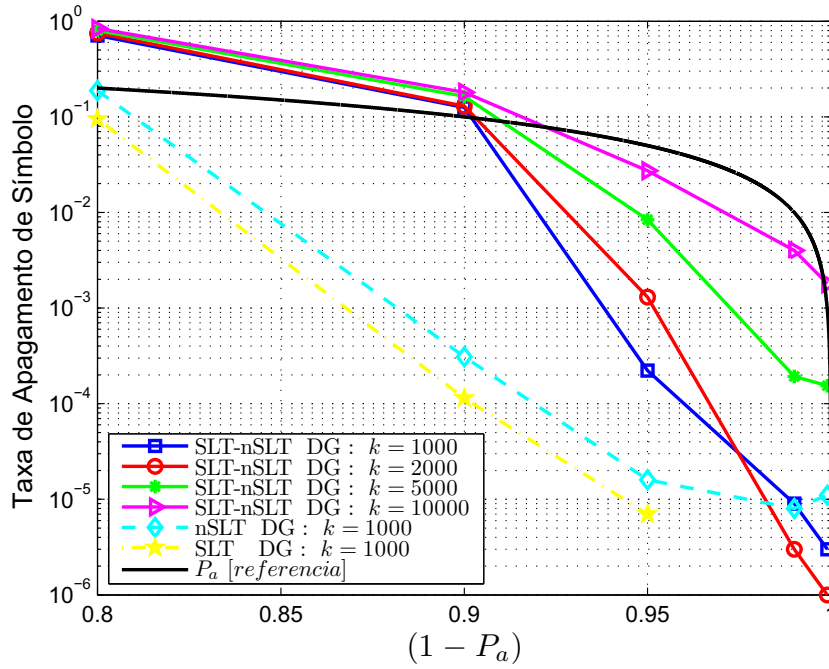


Figura 4.4: Comparação da TAS entre os esquemas nSLT, SLT e SLT-nSLT com diversas variações das distribuições de graus (DG) para  $\epsilon = 0.30$ , sendo  $\epsilon_H = 0.05$  e  $\epsilon_V = 0.24$ . Matrizes LT:  $(1000 \times 1000)$ , SLT-nSLT e  $(1 \times 1000)$ , nSLT e SLT.

no desempenho para valores de  $P_a \leq 2\%$ , ou seja, apenas em situações de ótima qualidade do canal. Na mesma figura há também as curvas SLT e nSLT com tamanhos de blocos  $k = 1000$  e utilizando DG para esse  $k$ . Observa-se que as duas curvas apresentam desempenhos bem melhores que as curvas dos esquemas bidimensionais.

Na Figura 4.5 é mostrada uma comparação entre a configuração SLT-nSLT [Paiba08] e outra composta por dois códigos LT sistemáticos (SLT-SLT).

A Figura 4.5, assim como as Figuras 4.1, 4.2 e 4.4, apresenta uma curva da probabilidade de apagamento do canal ( $P_a$ ), como referencial. Na Figura 4.5, observa-se que, tanto no caso onde  $\epsilon_H = 5\%$  e  $\epsilon_V = 14\%$  (ou  $\epsilon = 20\%$ ), quanto no caso onde  $\epsilon_H = 5\%$  e  $\epsilon_V = 24\%$  (ou  $\epsilon = 30\%$ ), a configuração proposta, formada por dois códigos sistemáticos (SLT-SLT), apresenta menores valores de taxa de apagamento se símbolo que a versão SLT-nSLT para qualquer valor de  $P_a$ . Tem-se assim, uma melhora no desempenho em relação ao esquema proposto em [Paiba08], além de uma significativa diminuição da complexidade dos processos de codificação e decodificação, devido ao uso de um código LT sistemático no lugar de um não-sistemático. Outro fato relevante

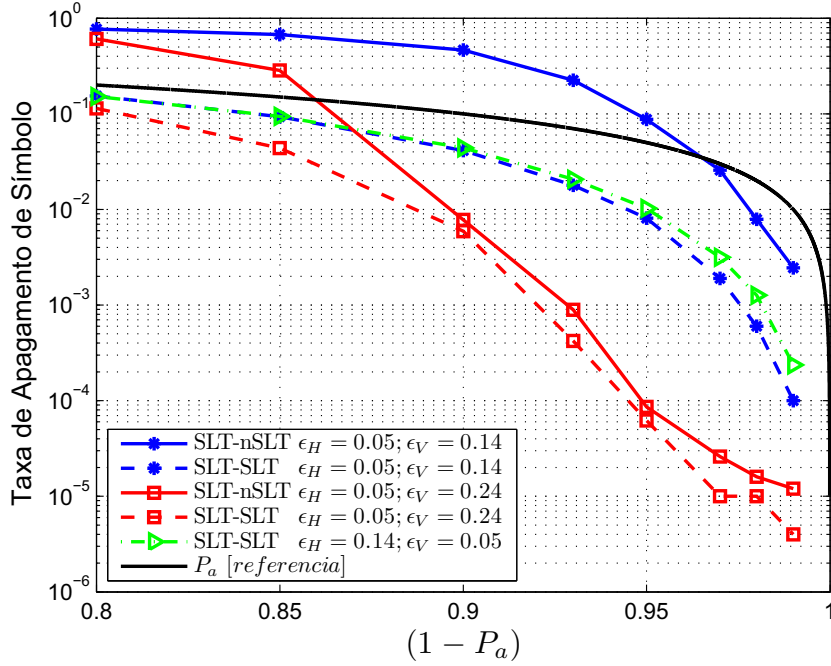


Figura 4.5: Comparação da TAS entre as configurações SLT-nSLT e SLT-SLT com diversas combinações de  $\epsilon_H$  e  $\epsilon_V$ . Matriz LT:  $(1000 \times 500)$ .

na análise da Figura 4.5 é que, enquanto os esquemas SLT-nSLT só melhoram seu desempenho em relação à curva de referência  $P_a$  em, aproximadamente,  $(1 - P_a) \geq 0.87$  (para  $\epsilon = 30\%$ ) e  $(1 - P_a) \geq 0.96$  (para  $\epsilon = 20\%$ ), ambos os esquemas SLT-SLT apresentam suas curvas abaixo de  $P_a$  em qualquer situação do canal. Isso significa que os esquemas SLT-nSLT pioram o canal em  $(1 - P_a) < 0.87$  ( $\epsilon = 30\%$ ) e em  $(1 - P_a) < 0.96$  ( $\epsilon = 20\%$ ) sendo preferível, nesses casos, transmitir a informação sem qualquer codificação. Já a configuração proposta (SLT-SLT) melhora o canal em qualquer situação da qualidade do canal.

Ainda na Figura 4.5, há uma curva SLT-SLT com  $\epsilon_H = 14\%$  e  $\epsilon_V = 5\%$ . Comparando-a à curva SLT-SLT com  $\epsilon_H = 5\%$  e  $\epsilon_V = 14\%$ , ou seja, invertendo os valores iniciais dos *overheads* horizontal e vertical, percebe-se que aumentando o valor de  $\epsilon_H$  o desempenho piora. Esse fato vai de encontro ao que foi concluído previamente na análise da Figura 4.3.

Uma análise semelhante àquela feita na Figura 4.3 para configurações SLT-nSLT é feita na Figura 4.6, agora com códigos SLT-SLT. Nesta figura, usa-se matrizes LT  $(1000 \times 1000)$  e também há curvas de códigos LT unidimensionais  $((1 \times 1000) - \text{SLT})$  com *overheads*  $\epsilon = 3\%$  e  $\epsilon = 13\%$  para comparação com os esquemas bidimensionais.

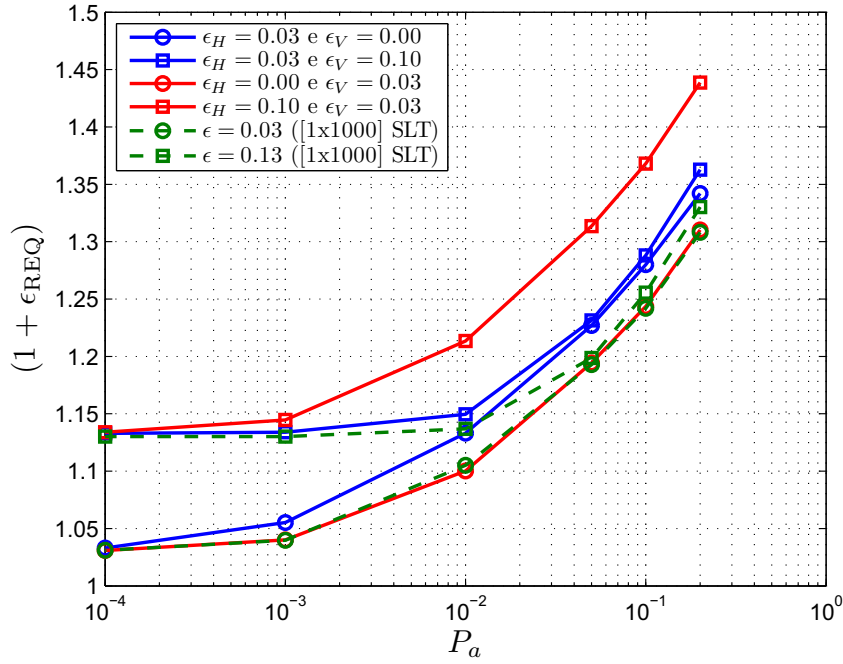


Figura 4.6: Comparação do *overhead* total requerido entre os esquemas SLT e SLT-SLT com diversas combinações de  $\epsilon_H$  e  $\epsilon_V$ . Matrizes LT:  $(1000 \times 1000)$ , SLT-SLT e  $(1 \times 1000)$ , SLT.

Fica claro que o esquema  $(1 \times 1000)$  com  $\epsilon = 3\%$  apresenta o mesmo desempenho do  $(1000 \times 1000)$  com  $\epsilon_H = 0\%$  e  $\epsilon_V = 3\%$ . Na verdade, era de se esperar pois, como o  $LT_H$  é sistemático e não há transmissão de novos símbolos horizontais (apenas verticais), na prática não há codificação horizontal nos esquemas bidimensionais com  $\epsilon_H = 0\%$ . Comparando-se essas curvas ao esquema SLT-SLT com  $\epsilon_H = 3\%$  e  $\epsilon_V = 0\%$ , vê-se que este último tem um desempenho pior, mais uma vez reforçando que o código  $LT_H$  não traz benefícios ao sistema.

Ainda na Figura 4.6, percebe-se que o esquema SLT com  $\epsilon = 13\%$  requer menos símbolos adicionais que o SLT-SLT com, aproximadamente, mesmo valor de *overhead* total, ou seja,  $\epsilon_H = 3\%$  e  $\epsilon_V = 10\%$ . Essa diferença no desempenho é ainda maior em relação ao SLT-SLT com  $\epsilon_H = 10\%$  e  $\epsilon_V = 3\%$ . Esse resultado, juntamente com o obtido na Figura 4.4, leva à conclusão de que os esquemas bidimensionais, da forma que estão sendo propostos, não trazem os benefícios esperados ao sistema, sendo melhor utilizar o esquema tradicional com apenas um código LT.

Os códigos LT bidimensionais são ainda uma proposta bastante recente [Paiba08] e, por isso, carecem de ser mais pesquisados. Nesta tese já se



conseguiu melhorar consideravelmente o desempenho dos mesmos através da substituição da configuração SLT-nSLT pela SLT-SLT. Conjectura-se que essa idéia possa sofrer uma evolução e alcançar resultados mais significativos.

## 4.2

### Códigos LT Convencionais

Inicialmente, nesta tese, deu-se continuidade ao estudo iniciado em [Paiba08], e tentou-se aprimorar os códigos LT bidimensionais propostos, através de modificações nas matrizes LT utilizadas (Figuras 4.1 e 4.2), de variações nos valores dos *overheads* horizontal e vertical (Figuras 4.3 e 4.6), do uso de distribuições de graus modificadas (Figura 4.4) e também através do uso de dois códigos LT sistemáticos (SLT-SLT) nos processos de codificação e decodificação (Figuras 4.5 e 4.6), conforme visto na seção anterior.

Embora o uso do esquema SLT-SLT tenha aprimorado o desempenho dos códigos LT bidimensionais, ainda não é possível afirmar que tais códigos apresentem uma real vantagem em relação aos códigos LT originais. A seguir, são apresentados alguns resultados obtidos com simulações de variações sobre os códigos LT tradicionais, visando uma melhora no desempenho.

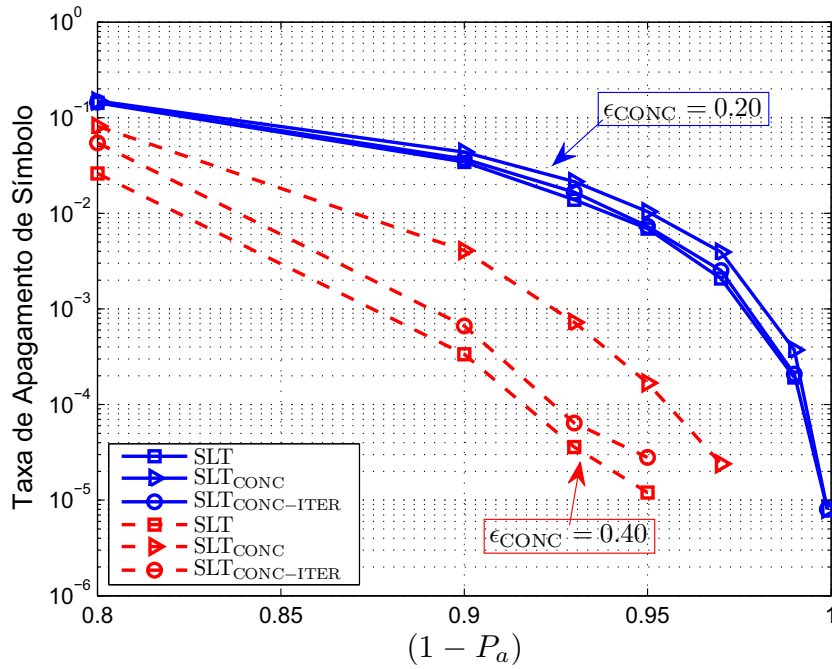


Figura 4.7: Comparação da TAS entre esquemas SLT,  $SLT_{CONC}$  e  $SLT_{CONC-ITER}$  com  $\epsilon_{CONC} = 0.20$  ou  $0.40$ .  $k = 500$ .

Primeiramente, propôs-se um esquema de concatenação serial de dois códigos LT sistemáticos ( $\text{SLT}_{\text{CONC}}$ ) onde o *overhead* total ( $\epsilon_{\text{CONC}}$ ) é resultante dos *overheads* de cada um dos códigos, conforme a Equação 3-1. Visando um melhoramento dessa configuração, propôs-se também um esquema onde, além da concatenação serial, há também uma iteração extra no processo de decodificação ( $\text{SLT}_{\text{CONC-ITER}}$ ). Uma comparação dessas duas propostas com o código LT sistemático tradicional (SLT), em duas situações ( $\epsilon_{\text{CONC}} = 20$  e  $40\%$ ), está na Figura 4.7. O esquema concatenado apresenta uma piora no desempenho em relação ao SLT, especialmente no caso onde  $\epsilon_{\text{CONC}} = 40\%$ . A iteração extra aplicada em conjunto com a concatenação serial praticamente iguala o desempenho do esquema SLT para valores de  $\epsilon_{\text{CONC}} = 20\%$ . Porém, com  $\epsilon_{\text{CONC}} = 40\%$ , percebe-se que o esquema tradicional SLT continua tendo valores mais baixos da taxa de apagamento de símbolo.

Como visto, o esquema  $\text{SLT}_{\text{CONC-ITER}}$  melhorou o desempenho do  $\text{SLT}_{\text{CONC}}$ , porém continuou sendo menos eficiente que o SLT tradicional. Como tentativa de melhorar isso, surgiu a idéia de inserir algumas perturbações na distribuição de graus utilizada, no caso a distribuição Sóliton Robusta Melhorada Truncada (DSRMT) para códigos LT sistemáticos. A Figura 4.8 mostra os resultados das simulações para duas perturbações distintas. Primeiramente, subtraiu-se em 1, todos os graus maiores ou iguais a 3 (se  $d_i \geq 3 \Rightarrow d'_i = d_i - 1$ , onde  $d'_i$  é o novo valor do grau  $d_i$ ). Este esquema é identificado na Figura 4.8 por  $\text{SLT}_{\text{CONC-ITER}}^1$ . A outra perturbação analisada foi subtrair-se em 1, todos os graus maiores ou iguais a 4 (se  $d_i \geq 4 \Rightarrow d'_i = d_i - 1$ ) e está identificada no gráfico por  $\text{SLT}_{\text{CONC-ITER}}^2$ .

Ambas as perturbações mostraram-se ineficientes, tendo em vista que apresentaram desempenho ligeiramente pior que o esquema  $\text{SLT}_{\text{CONC-ITER}}$  correspondente. Isso fica claro na Figura 4.8, sendo a visualização mais fácil nos casos com *overhead*  $\epsilon_{\text{CONC}} = 20\%$ .

Uma outra variação proposta nesta tese é a aplicação da técnica de alongamento nos códigos LT, explanada na Subseção 3.2.2. As figuras a seguir trazem os resultados das simulações realizadas utilizando-se essa técnica. Esse novo esquema é identificado nos gráficos como SLTA. O bloco **z** de símbolos conhecidos pelo codificador e pelo decodificador, que é concatenado ao bloco de informações nas figuras a seguir, é gerado de forma aleatória e com os símbolos sendo equiprováveis. Realizou-se testes com esses blocos sendo formados apenas por símbolos (*bits*) 0 ou por símbolos 1, sem comprometimento dos resultados.

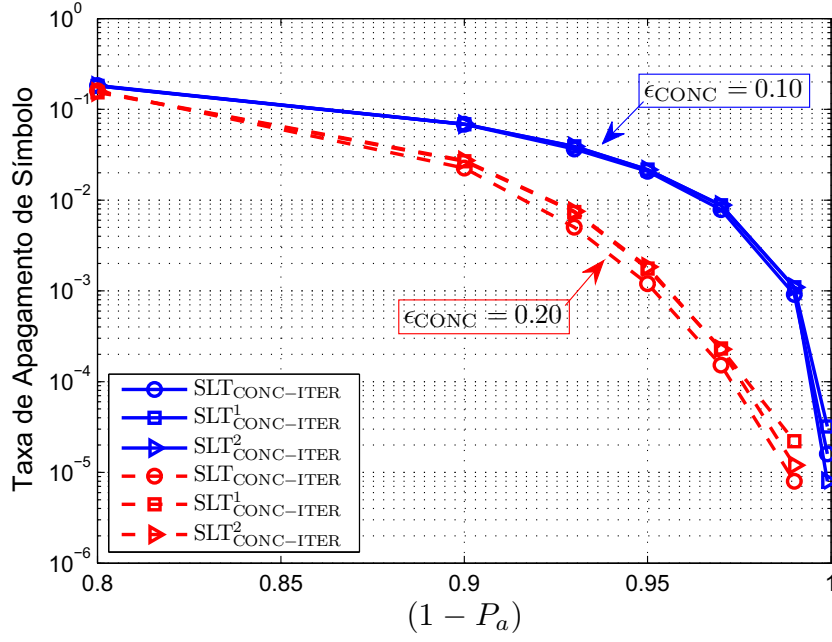


Figura 4.8: Comparação da TAS entre esquemas  $SLT_{CONC-ITER}$ ,  $SLT^1_{CONC-ITER}$  e  $SLT^2_{CONC-ITER}$  com  $\epsilon_{CONC} = 0.10$  ou  $0.20$ .  $k = 1000$ .

A Figura 4.9 mostra o gráfico da taxa de apagamento de símbolo (TAS) em função da qualidade do canal BEC onde são comparados quatro esquemas SLTA com diferentes tamanhos do bloco inserido de símbolos conhecidos, com o esquema SLT tradicional correspondente, ou seja, com valores de *overhead* aproximadamente iguais ( $\epsilon \approx \epsilon_{SLTA}$ ). O tamanho do bloco de informação utilizado nas simulações foi de  $k = 500$ . A Tabela 4.2 mostra, para cada curva apresentada na Figura 4.9, o tamanho  $|\mathbf{z}|$  do bloco  $\mathbf{z}$  de símbolos conhecidos inserido, o fator de alongamento  $\eta$ , o “*overhead* intermediário”  $\epsilon'$  utilizado e o tamanho  $|\mathbf{c}|$  do bloco codificado transmitido  $\mathbf{c}$  (ver Subseção 3.2.2). A Equação 3-10 faz-se importante para um bom entendimento da Tabela 4.2 a seguir.

Tabela 4.2: Parâmetros utilizados nas Figuras 4.9, 4.10 e 4.11 para  $k = 500$ .

$\eta$	$ \mathbf{z} $	$ \mathbf{u}_{\mathbf{z}} $	$\epsilon'$	$ \mathbf{c}_{\mathbf{z}} $	$ \mathbf{c} $
10%	50	550	10%	605	555
20%	100	600	9%	654	554
40%	200	700	8%	756	556
80%	400	900	6%	954	554

Observa-se, na Figura 4.9, que para aproximadamente  $P_a \leq 6\%$ , os esquemas SLTA apresentam menores valores de TAS que o código LT sistemático

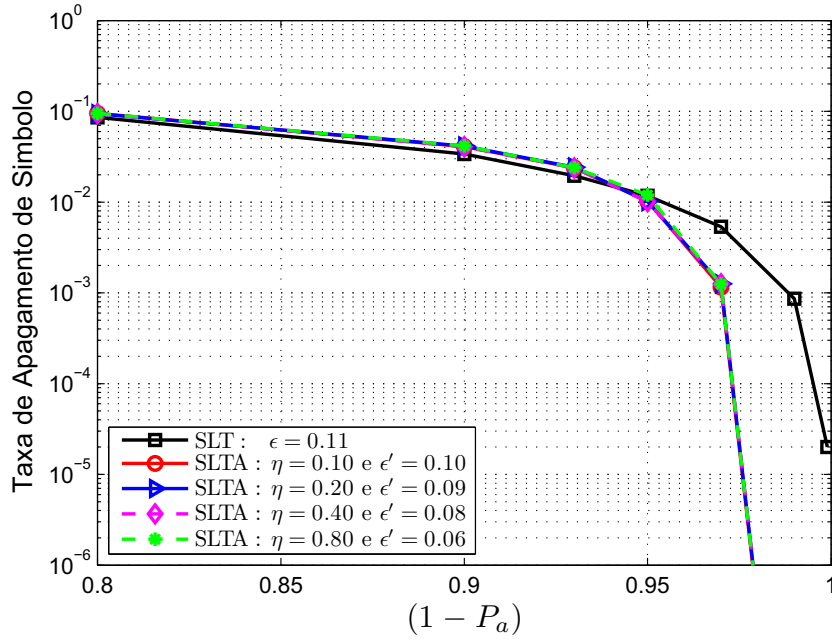


Figura 4.9: Comparação da TAS entre um esquema SLT e esquemas SLTA com diversas combinações de  $\eta$  e  $\epsilon'$ .  $\epsilon = 0.11$ ,  $\epsilon_{\text{SLTA}} \approx 0.11$  e  $k = 500$ .

sem modificações (SLT). Isso mostra uma real vantagem no desempenho dos códigos SLTA propostos em relação ao código SLT convencional.

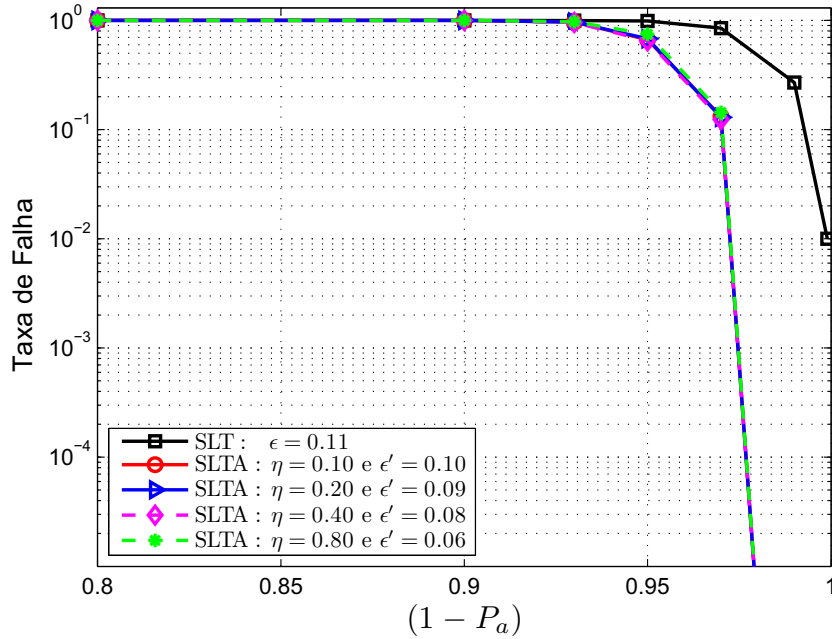


Figura 4.10: Comparação da TF entre um esquema SLT e esquemas SLTA com diversas combinações de  $\eta$  e  $\epsilon'$ .  $\epsilon = 0.11$ ,  $\epsilon_{\text{SLTA}} \approx 0.11$  e  $k = 500$ .

As Figuras 4.10 e 4.11 mostram os gráficos da taxa de falha (TF) e do *overhead* total requerido ( $1 + \epsilon_{\text{REQ}}$ ) para uma correta decodificação, para os mesmos esquemas utilizados na Figura 4.9. Por taxa de falha entenda-se a razão entre o número de blocos contendo ao menos um símbolo não recuperado (considerado apagado), após o processo de decodificação, e o número de blocos enviado. Estes blocos são então descartados. A observação dessas duas figuras também mostra um melhor desempenho dos esquemas SLTA propostos em relação aos SLT. A taxa de falha dos esquemas SLTA são mais baixas em relação ao esquema SLT em  $P_a \leq 7\%$ , enquanto que, em termos do *overhead* total requerido, os esquemas SLTA têm melhor desempenho para  $P_a \leq 12\%$ .

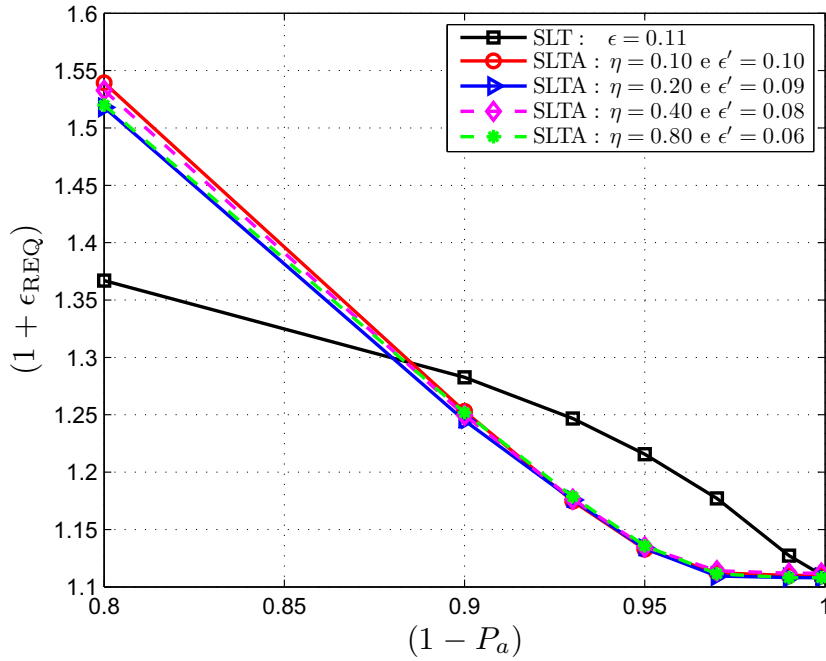


Figura 4.11: Comparação do *overhead* total requerido entre um esquema SLT e esquemas SLTA com diversas combinações de  $\eta$  e  $\epsilon'$ .  $\epsilon = 0.11$ ,  $\epsilon_{\text{SLTA}} \approx 0.11$  e  $k = 500$ .

A taxa de apagamento de símbolo e a taxa de falha, em função da qualidade do canal, dos mesmos esquemas mostrados nas Figuras 4.9 e 4.10, são apresentadas nas Figuras 4.12 e 4.13. Entretanto, os valores de  $\eta$  e  $\epsilon'$  utilizados, obedecem à Tabela 4.3, resultando em  $\epsilon_{\text{SLTA}} \approx 32\%$ .

O melhor desempenho dos códigos SLTA propostos em relação aos códigos LT é ainda maior nas Figuras 4.12 e 4.13. As TAS dos esquemas SLTA são menores que a do esquema SLT para valores de  $P_a \leq 8\%$ , segundo a Figura 4.12. Na Figura 4.13, percebe-se que as taxas de falha dos esquemas SLTA são menores que o caso SLT, mesmo em situações de péssima qualidade do canal.

Tabela 4.3: Parâmetros utilizados nas Figuras 4.12 e 4.13 para  $k = 500$ .

$\eta$	$ z $	$ u_z $	$\epsilon'$	$ c_z $	$ c $
5%	25	525	30%	680	655
10%	50	550	28%	704	654
20%	100	600	26%	756	656
40%	200	700	22%	854	654
80%	400	900	17%	854	654

Esses resultados, que são a principal contribuição desta tese, comprovam o sucesso da idéia de inserir um bloco de símbolos conhecidos – tanto pelo codificador, quanto pelo decodificador – num sistema de transmissão através de um canal BEC, fazendo uso de um código LT sistemático. Em outras palavras, mostrou-se com esses resultados que o uso da técnica de alongamento com códigos LT sistemáticos e, generalizando-se, com códigos Fontanais, pode ser bastante útil e eficaz. Apesar de as análises terem sido realizadas todas em canais BEC, conjectura-se que os resultados podem também ser satisfatórios em outros canais que possam ser tratados como canais com apagamento.

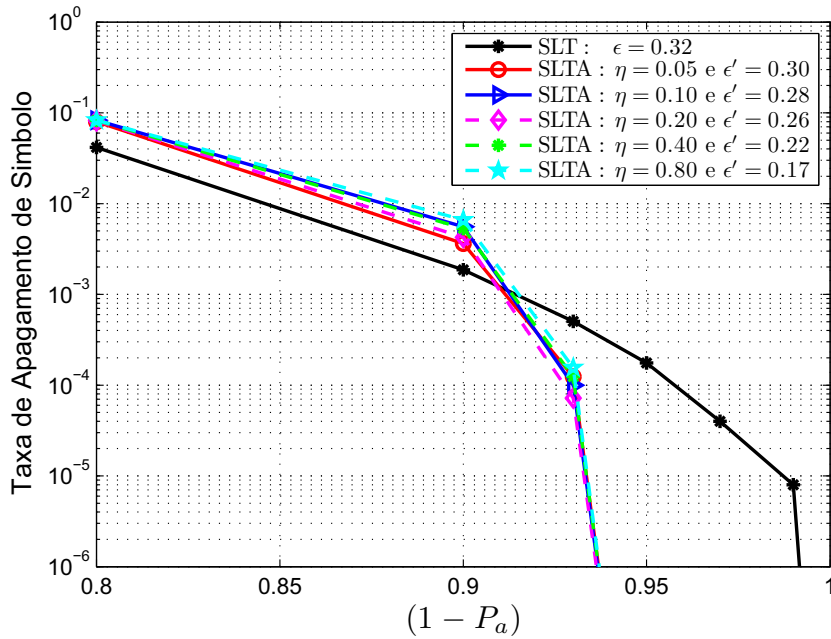


Figura 4.12: Comparação da TAS entre um esquema SLT e esquemas SLTA com diversas combinações de  $\eta$  e  $\epsilon'$ .  $\epsilon = 0.32$ ,  $\epsilon_{\text{SLTA}} \approx 0.32$  e  $k = 500$ .

Ao se comparar um esquema SLTA com um SLT de mesmos *overheads*, faz-se uma comparação justa, pois a mesma quantidade de símbolos (*bits*) é transmitida em cada caso. É o que ocorre nas Figuras 4.9 a 4.13. Através delas,

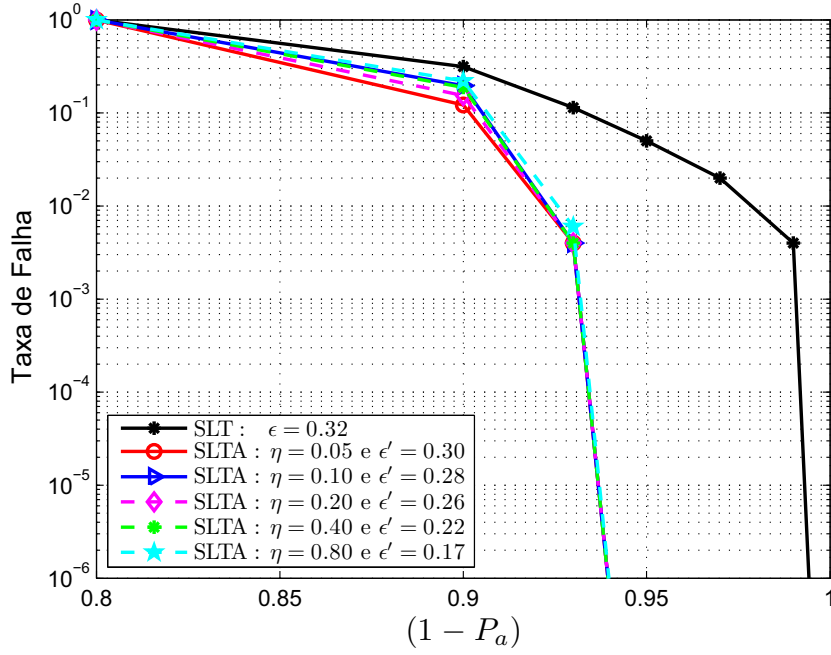


Figura 4.13: Comparação da TF entre um esquema SLT e esquemas SLTA com diversas combinações de  $\eta$  e  $\epsilon'$ .  $\epsilon = 0.32$ ,  $\epsilon_{\text{SLTA}} \approx 0.32$  e  $k = 500$ .

percebeu-se que os esquemas SLTA apresentam melhor desempenho, seja em função de TAS, TF ou *overhead* total requerido, em relação aos esquemas SLT sob as mesmas condições de redundância. As Figuras 4.14, 4.15 e 4.16 a seguir comparam um esquema SLT de *overhead*  $\epsilon = 32\%$  a três esquemas SLTA de *overheads*  $\epsilon_{\text{SLTA}} = 11, 21$  e  $32\%$ . Esses valores de *overhead* foram obtidos, respectivamente, através das seguintes combinações de  $\eta$  e  $\epsilon'$ : (a)  $\eta = 10\%$  e  $\epsilon' = 10\%$ ; (b)  $\eta = 10\%$  e  $\epsilon' = 19\%$  e (c)  $\eta = 10\%$  e  $\epsilon' = 28\%$ .

O objetivo dessas figuras é mostrar que o uso da técnica de alongamento nos códigos LT sistemáticos (SLTA) proporciona uma melhora de desempenho em relação a um esquema SLT, mesmo utilizando-se menores valores de *overhead*. Em outras palavras, transmitindo-se menos símbolos, pode-se obter resultados melhores, ao menos para algumas situações do canal.

Na Figura 4.14 nota-se que o esquema SLTA com  $\epsilon_{\text{SLTA}} = 21\%$  ( $\eta = 10\%$  e  $\epsilon' = 19\%$ ) apresenta menor valor de TAS que o esquema SLT com  $\epsilon = 32\%$ , para  $P_a \leq 5\%$ . Para  $P_a \leq 2.5\%$ , até mesmo o SLTA com  $\epsilon_{\text{SLTA}} = 11\%$  ( $\eta = 10\%$  e  $\epsilon' = 10\%$ ) tem TAS mais baixa que o SLT com  $\epsilon = 32\%$ . Por analogia, a mesma análise pode ser feita na Figura 4.15, agora em relação à taxa de falha.

Em relação ao *overhead* total requerido pelo sistema para uma correta

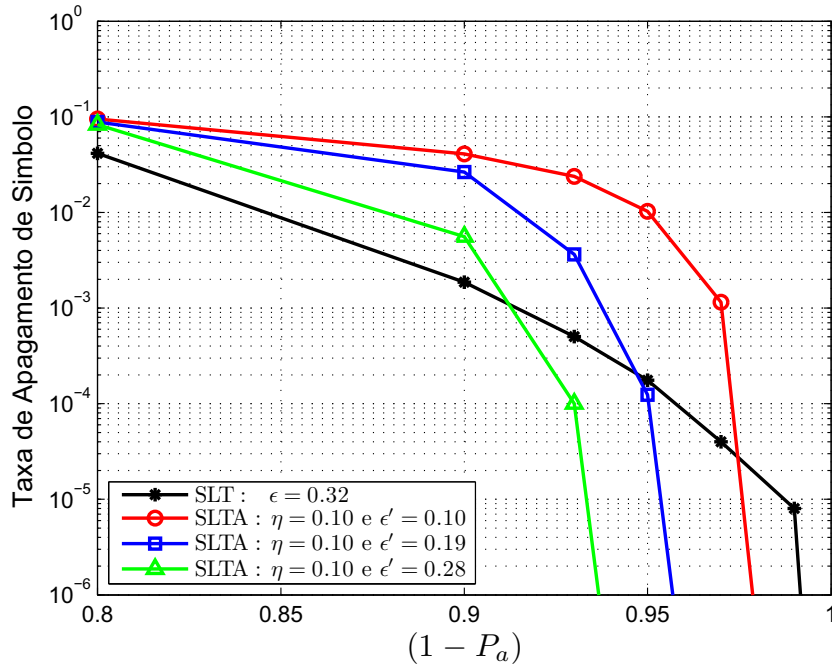


Figura 4.14: Comparação da TAS entre um esquema SLT com  $\epsilon = 32\%$  e esquemas SLTA com  $\eta = 10\%$  e  $\epsilon' = 10, 19$  e  $28\%$ .  $k = 500$ .

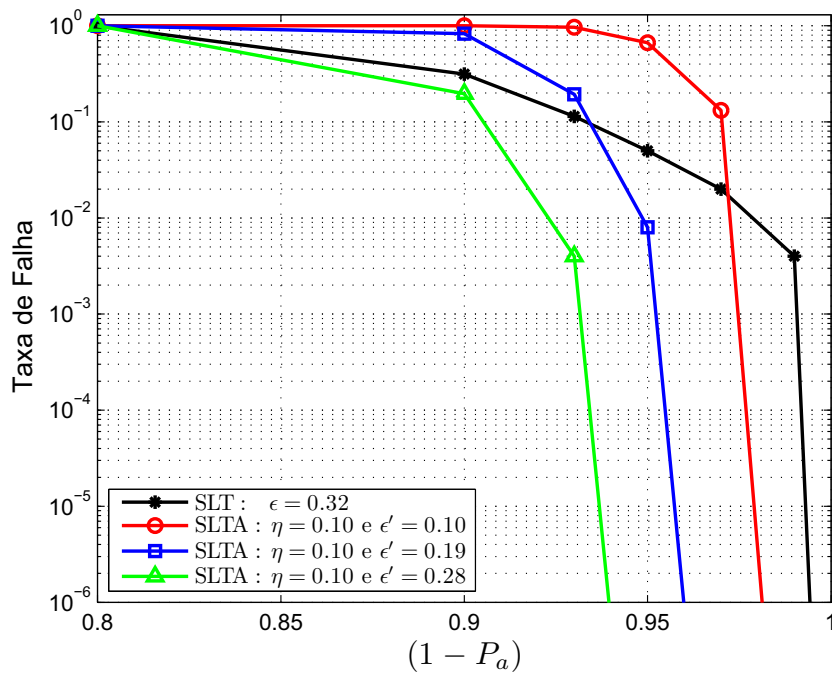


Figura 4.15: Comparação da TF entre um esquema SLT com  $\epsilon = 32\%$  e esquemas SLTA com  $\eta = 10\%$  e  $\epsilon' = 10, 19$  e  $28\%$ .  $k = 500$ .



decodificação dos símbolos de informação transmitidos, percebe-se que em situações de má qualidade do canal ( $P_a \geq 11\%$ ), o sistema SLT é mais indicado. Porém, à medida que o canal vai melhorando, os esquemas SLTA mostram-se mais vantajosos, pois necessitam de menos símbolos adicionais e, por consequência, apresentam menores valores de  $(1 + \epsilon_{REQ})$ . Isto pode ser comprovado no gráfico da Figura 4.16.

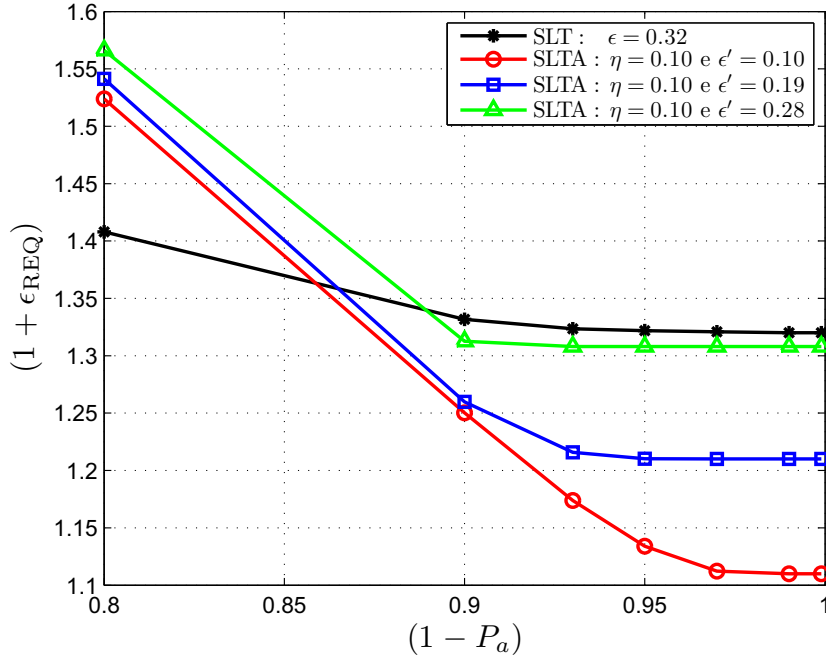


Figura 4.16: Comparação do *overhead* total requerido entre um esquema SLT com  $\epsilon = 32\%$  e esquemas SLTA com  $\eta = 10\%$  e  $\epsilon' = 10, 19$  e  $28\%$ .  $k = 500$ .

Os resultados das Figuras 4.9 a 4.16 mostram que os códigos LT sistemáticos alongados (SLTA) propostos nesta tese são uma boa opção para o aumento do desempenho dos códigos LT propostos por Luby em [Luby02]. Mostrou-se que, especialmente em condições de boa qualidade do canal BEC, tais códigos apresentam melhores resultados em termos de taxa de apagamento de símbolo (TAS), de taxa de falha (TF) e também de *overhead* total requerido para uma correta decodificação. Esses resultados são a principal contribuição científica desta tese.