

UNIVERSIDADE FEDERAL DO PARÁ  
INSTITUTO DE TECNOLOGIA  
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

MÉTODO DE INVERSÃO MATRICIAL RÁPIDO GSGR PARA VECTORING  
EM SISTEMAS XDSL, COM IMPLEMENTAÇÃO EM CHIP DSP

MARIA NILMA DA SILVA FONSECA

DM 28/2013

UFPA / ITEC / PPGEE  
Campus Universitário do Guamá  
Belém-Pará-Brasil  
2013

UNIVERSIDADE FEDERAL DO PARÁ  
INSTITUTO DE TECNOLOGIA  
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

MARIA NILMA DA SILVA FONSECA

MÉTODO DE INVERSÃO MATRICIAL RÁPIDO GSGR PARA VECTORING  
EM SISTEMAS XDSL, COM IMPLEMENTAÇÃO EM CHIP DSP

DM 28 / 2013

UFPA / ITEC / PPGEE  
Campus Universitário do Guamá  
Belém-Pará-Brasil  
2013

UNIVERSIDADE FEDERAL DO PARÁ  
INSTITUTO DE TECNOLOGIA  
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

MARIA NILMA DA SILVA FONSECA

MÉTODO DE INVERSÃO MATRICIAL RÁPIDO GSGR PARA VECTORING  
EM SISTEMAS XDSL, COM IMPLEMENTAÇÃO EM CHIP DSP

Dissertação submetida à Banca Examinadora do Programa de Pós-Graduação em Engenharia Elétrica da UFPA para a obtenção do Grau de Mestre em Engenharia Elétrica na área de Telecomunicações.

UFPA / ITEC / PPGEE  
Campus Universitário do Guamá  
Belém-Pará-Brasil  
2013

UNIVERSIDADE FEDERAL DO PARÁ  
INSTITUTO DE TECNOLOGIA  
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

MÉTODO DE INVERSÃO MATRICIAL RÁPIDO GSGR PARA VECTORING  
EM SISTEMAS XDSL, COM IMPLEMENTAÇÃO EM CHIP DSP

AUTOR: MARIA NILMA DA SILVA FONSECA

DISSERTAÇÃO DE MESTRADO SUBMETIDA À AVALIAÇÃO DA BANCA EXAMINADORA APROVADA PELO COLEGIADO DO PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA DA UNIVERSIDADE FEDERAL DO PARÁ E JULGADA ADEQUADA PARA OBTENÇÃO DO GRAU DE MESTRE EM ENGENHARIA ELÉTRICA NA ÁREA DE TELECOMUNICAÇÕES.

APROVADA EM \_\_\_\_/\_\_\_\_/\_\_\_\_

BANCA EXAMINADORA:

---

Prof. Dr. ALDEBARO BARRETO DA ROCHA KLAUTAU JUNIOR  
(ORIENTADOR – UFPA)

---

Prof. Dr. FRANCISCO CARLOS BENTES FREY MULLER  
(MEMBRO – UFPA)

---

Prof. Dr. HERMÍNIO SIMÕES GOMES  
(MEMBRO – UFPA)

VISTO:

---

Prof. Dr. EVALDO GONÇALVES PELAES  
(COORDENADOR DO PPGEE/ITEC/UFPA)

## AGRADECIMENTOS

*Dedico este trabalho e agradeço a todos que me ajudaram a realizá-lo, em especial:  
a minha mãezinha Naile e meu paizinho Nélio por toda a dedicação, inspiração na escolha  
da profissão e apoio pra me tornar uma pessoa um pouco melhor a cada dia;  
a minha irmã Norma pelo apoio incondicional em toda minha vida;  
a minha irmã Neiva, companheira e quase filha;  
ao meu orientador Prof. Aldebaro por acreditar em mim e me indicar os caminhos;  
ao Prof. Pelaes por resolver tantos assuntos pra mim, sempre atencioso;  
ao Prof. Müller pelas nossas reuniões esclarecedoras quase semanais;  
ao Prof. Hermínio, alguns momentos de conversa e meu deu tantas conselhos;  
aos colegas Kelly, Diego, Marcos, Iago, Bruno, Cláudio, Fernanda, Igor, Diogo, Patrício,  
Leonardo, Ilan e todos do Laps, obrigada sinceramente pela ajuda preciosa;  
e a Deus por dar vida aos sonhos.*

## SUMÁRIO

1	INTRODUÇÃO.....	1
2	REVISÃO DO ESTADO DA ARTE DE MÉTODOS PARA INVERSÃO MATRICIAL.....	3
2.1	RESOLUÇÃO DE SISTEMAS LINEARES.....	4
2.2	MÉTODOS DIRETOS.....	5
2.2.1	Método de Substituição .....	5
2.2.2	Método de Retrossubstituição .....	5
2.2.3	Decomposição LU .....	6
2.2.4	Decomposição de <i>Cholesky</i> .....	6
2.2.5	Decomposição QR.....	7
2.3	MÉTODOS ITERATIVOS.....	8
2.3.1	Método de <i>Jacobi</i> .....	9
2.3.2	Método de <i>Gauss-Seidel</i> .....	9
2.3.3	Método SOR ( <i>Successive Overrelaxation</i> ).....	10
2.4	CÁLCULO EM PONTO FLUTUANTE.....	10
3	REVISÃO DO ESTADO DE ARTE DAS TECNOLOGIAS DE LINHA DIGITAL PARA ASSINANTE ( <i>DIGITAL SUBSCRIBER LINE - XDSL</i> ) .....	13
3.1	AMBIENTE XDSL.....	15
3.2	MODULAÇÃO DMT.....	16
3.3	CARACTERÍSTICAS ASSUMIDAS PARA O MODELO DO CANAL EM SISTEMAS XDSL .....	20
3.4	<i>CROSSTALK</i> .....	20
3.5	PRINCÍPIOS DE CANCELAMENTO DE <i>CROSSTALK FEXT</i> .....	22
3.6	TRANSMISSÃO <i>DOWNSTREAM</i> COM VECTORING .....	24
3.6.1	Pré-codificador <i>Zero-Forcing</i> .....	25
3.7	TRANSMISSÃO <i>UPSTREAM</i> COM VECTORING.....	26
3.7.1	Cancelador <i>Zero-Forcing (ZF)</i> .....	27
4	MATRIZES DE CANAL DE SISTEMAS XDSL <i>G.FAST</i> .....	29
4.1	CARACTERÍSTICAS DAS MATRIZES <i>G.FAST</i> CONSIDERADAS .....	29
4.2	ESTIMAÇÃO DAS MATRIZES DE CANAL <i>G.FAST</i> .....	34
4.3	REQUISITOS DE TEMPO DE INVERSÃO E ACURÁCIA .....	36
4.3.1	Coefficientes de acoplamento <i>FEXT</i> em <i>downstream</i> .....	38

4.3.2	Coeficientes de acoplamento FEXT em <i>upstream</i> .....	40
5	MÉTODO DE INVERSÃO MATRICIAL PROPOSTO GSGR APLICADO EM PRÉ-CODIFICAÇÃO DE SISTEMAS XDSL <i>G.FAST</i> .....	42
5.1	DECOMPOSIÇÃO QR COM MÉTODO DE <i>GIVENS ROTATIONS</i> (GR).....	42
5.2	DECOMPOSIÇÃO QR COM <i>SQUARED GIVENS ROTATIONS</i> (SGR) .....	44
5.2.1	Ocorrência de elementos nulos durante o processo .....	47
5.3	INVERSÃO DA MATRIZ TRIANGULAR OBTIDA COM SGR .....	48
5.4	CÁLCULO MATRICIAL EM BLOCO - COMPLEMENTO DE SCHUR.....	49
5.5	MELHORAMENTOS PROPOSTOS .....	50
6	RESULTADOS DA APLICAÇÃO DO MÉTODO DE INVERSÃO MATRICIAL GSGR A MATRIZES <i>G.FAST</i> .....	52
6.1	RESULTADOS DOS TESTES DE DESEMPENHO .....	52
6.2	RESULTADOS DOS TESTES DE ACURÁCIA .....	58
6.3	RESULTADOS DE TESTES DE FIGURAS DE MÉRITO .....	65
7	CONCLUSÕES E TRABALHOS FUTUROS .....	71
8	REFERÊNCIAS .....	72
A	APÊNDICE: CONCEITOS BÁSICOS DE MATRIZES EM ÁLGEBRA LINEAR .....	74

## LISTA DE FIGURAS

Figura 3.1	Ilustrações de modelos de rede: (a) FTTN; (b) FTTdp; (c) FTTB e FTTH. ....	14
Figura 3.2	Conceito de divisão da largura de banda de frequência de um canal em subcanais, em um sistema de modulação por multiportadora. ....	16
Figura 3.3	SNR de um canal hipotético e possível distribuição de <i>bits</i> por subcanal .....	17
Figura 3.4.	Ilustração de uma transmissão DMT em cabeamento de pares de fios de cobre .	18
Figura 3.5	Ilustração de uma rede híbrida típica xDSL – ON e <i>crosstalk</i> FEXT e NEXT no cabeamento. ....	21
Figura 3.6	Descrição funcional do método de <i>Vectoring</i> em <i>downstream</i> e <i>upstream</i> [11].	23
Figura 3.7	Modelo de referência para um sistema com <i>vectoring</i> , ilustrado para a linha $n = 1$ de um grupo de $N$ linhas. ....	24
Figura 3.8	Ilustração do modelo funcional de um grupo de linhas vetoradas usando pré-codificador de cancelamento de FEXT em <i>downstream</i> .....	25
Figura 3.9	Ilustração da transmissão <i>upstream</i> com <i>vectoring</i> .....	27
Figura 4.1	Representação de uma matriz de canal <i>G.fast</i> , de 24 x 24 elementos de medições normalizadas, na frequência de 5,17 MHz. (a) valores reais; (b) valores imaginários; e (c) valores absolutos. ....	30
Figura 4.2	Representação do somatório de 51 matrizes de canal <i>G.fast</i> , de 24 x 24 elementos de medições normalizadas, na faixa de frequências de 5,17 MHz a 7,76 MHz. (a) valores reais; (b) valores imaginários; e (c) valores absolutos .....	31
Figura 4.3	Representação de uma matriz de canal <i>G.fast</i> , de 24 x 24 elementos de medições normalizadas, na frequência de 211,91 MHz. (a) valores reais; (b) valores imaginários; e (c) valores absolutos. ....	32
Figura 4.4	Representação do somatório de 97 matrizes de canal de <i>G.fast</i> , de 24 x 24 elementos de medições normalizadas, na faixa de frequências de 206,95 MHz a 211,91 MHz. (a) valores reais; (b) valores imaginários; e (c) valores absolutos .....	33
Figura 4.5	Tensão aplicada à carga $R_n$ .....	34
Figura 4.6	Tensão aplicada à carga $R_n$ com a inserção de uma rede de porta dupla .....	34
Figura 4.7	Visão geral do procedimento de inicialização .....	37
Figura 4.8	Definição de $X_{lin}$ em <i>downstream</i> – tensão FEXT do canal. ....	39
Figura 4.9	Definição de $X_{lin}$ em <i>downstream</i> – tensão direta do canal. ....	39
Figura 4.10	Definição de $X_{lin}$ em <i>upstream</i> – tensão de FEXT do canal. ....	41
Figura 4.11	Definição de $X_{lin}$ em <i>upstream</i> – tensão direta do canal .....	41
Figura 5.1	Ocorrência de zeros indesejados na matriz a ser triangularizada. ....	47
Figura 5.2	Ilustração da aplicação do método SGR em uma matriz $A_{3 \times 3}$ para obtenção de uma matriz triangular superior $U$ .....	48

Figura 6.1	Comparações do número de operações de adição (a), multiplicação (b) e divisão (c) necessário para inverter uma matriz de dimensão $n$ , usando os algoritmos LU, MSGR e GSGR. ....	54
Figura 6.2	Comparação do número total de operações de ponto flutuante teóricas – <i>flops</i> necessário para inverter uma matriz de dimensão $n$ , usando os algoritmos LU, MSGR e GSGR. ....	55
Figura 6.3	Períodos de tempo para inversão de 20 matrizes $A_{64 \times 64}$ em uma plataforma DSP executando os algoritmos LU, MSGR e GSGR. ....	57
Figura 6.4	Desvios máximo e médio de $ I - A.A^{-1} $ dos algoritmos MGS, MSGR, GSGR, LU e da função inversa do Matlab, resultantes da inversão de 100 matrizes de $100 \times 100$ . 60	60
Figura 6.5	Desvios máximo, médio e padrão de $ I - A.A^{-1} $ na inversão de 20 matrizes $A_{64 \times 64}$ pelos algoritmos GSGR, LU e MSGR em um DSP TMS320C6670. ....	61
Figura 6.6	Desvios máximo, médio e padrão de $ I - A.A^{-1} $ para o algoritmo GSGR e função inversa do Matlab usando precisão simples (PS) e precisão dupla (PD). As matrizes $A$ são matrizes de ganho de canal <i>G.fast</i> $24 \times 24$ obtidas de medições realizadas nas frequências de 206,95 MHz a 211,91 MHz. ....	62
Figura 6.7	Comparação de desvios máximos de $ I - A.A^{-1} $ para os algoritmos GSGR, MSGR e LU em precisão dupla (PD). As matrizes $A$ usadas nesta simulação são matrizes de ganho de canal de um cabeamento com 24 linhas, obtidas de medições realizadas nas frequências desde 0 até 211,91 MHz. ....	63
Figura 6.8	Comparação de desvios máximos de $ I - A.A^{-1} $ para os algoritmos GSGR, MSGR e LU em precisão dupla (PD). As matrizes $A$ usadas nesta simulação são matrizes de ganho de canal de um cabeamento de 30m e com 10 linhas, obtidas de medições realizadas nas frequências desde 0 até 211,91 MHz. ....	64
Figura 6.9	Taxas de sinal-ruído (SNR) obtidas com pré-codificação para cancelamento de <i>crossstalk</i> , usando o GSGR e INV-Matlab. Dados de um cabeamento de 100 metros e 24 linhas. ....	66
Figura 6.10	Densidade de potência (PSD) média entre linhas após pré-codificação para cancelamento de <i>crossstalk</i> , usando o GSGR e INV-Matlab. Dados de um cabeamento de 100 metros e 24 linhas. ....	67
Figura 6.11	Taxas de sinal-ruído (SNR) obtidas com pré-codificação para cancelamento de <i>crossstalk</i> , usando o GSGR e INV-Matlab. Dados de um cabeamento de 30 metros e 10 linhas. ....	68
Figura 6.12	Densidade de potência (PSD) média das linhas após pré-codificação para cancelamento de <i>crossstalk</i> , usando o GSGR e INV-Matlab. Dados de um cabeamento de 100 metros e 24 linhas. ....	69

**LISTA DE TABELAS**

Tabela 2.1	Parâmetros dos formatos básicos de números em ponto flutuante do padrão IEEE 754/2008.....	12
Tabela 6.1	Comparação de complexidade dos algoritmos testados com relação ao número de operações em função da dimensão $n$ da matriz .....	52
Tabela 6.2	Comparação de complexidade dos algoritmos testados com relação ao número de operações para uma matriz de dimensão $n = 4$ .....	53
Tabela 6.3	Pesos usados para o cálculo teórico do número de operações em ponto flutuante - flops para uma operação de inversão .....	55
Tabela 6.4	Total ponderado de operações - flops dos algoritmos de inversão testados.....	56
Tabela 6.5	Comparação da redução de flops obtida pelo método GSQR quando aplicado para inversão de 4096 matrizes de 64x64 .....	56
Tabela 6.6	Tempo máximo de execução dos algoritmos de inversão testados.....	57

## LISTA DE ABREVIATURAS

ADSL - *asymmetric DSL* – DSL assimétrica  
 AN – *access node* – ponto ou nó de acesso  
 BER – *bit error rate* – taxa de erros de bits  
 CO - *central office* – instalação do operador de telefonia  
 CP – *customer premise* – instalação do usuário  
 CPE – *customer premise equipment* – equipamento do usuário  
 DMT – *discrete multi tone* – modulação discreta multi-ton  
 DSL – *digital subscriber line* – linha digital de assinante  
 DSLAM – *DSL multiplexer* – multiplexador DSL  
 DSP – *digital signal processing* – processamento digital de sinais  
 FDD – *frequency division duplexing* – transmissão por divisão de frequência  
 FEXT – *far end crosstalk*  
 FTTB – *fiber to the building* - rede de fibra óptica até a entrada de prédios  
 FTTdP – *fiber to the distribution point* - rede de fibra óptica até pontos de distribuição  
 FTTN – *fiber to the neighbourhood* – rede de fibra óptica até as proximidades dos assinantes  
 FTTx – *fiber to the X* – representa os tipos de rede de fibra óptica  
 G.FAST – *fast subscriber line* – linha de assinante rápida  
 GR – *Givens rotations* – método de Rotações de *Givens*  
 GSGR – *Gauss squared givens rotations*  
 ITU-T – *International Telecommunications Union – Telecommunications Standardization Sector*  
 LS – *least squares* – mínimos quadrados  
 MSGR – *modified squared givens rotations*  
 NEXT – *near crosstalk*  
 NT – *network termination* - terminação de rede  
 ON – *optical network* – rede óptica  
 ONU – *optical network unit* – unidade de rede óptica  
 PSD – *power spectral density* – densidade espectral de potência  
 SGR – *squared givens rotation*  
 SNR – *signal noise ratio* - taxa sinal-ruído  
 SOC – *system on chip* – sistema em um circuito integrado  
 SOR – *successive overrelaxation* – sobre-relaxação sucessiva  
 TDD – *time division duplexing*- transmissão por divisão de tempo  
 TI – *Texas Instruments*  
 VDSL – *very high speed DSL* – DSL de velocidade muito alta  
 xDSL – família de tecnologias DSL  
 ZF – *zero forcing*

## RESUMO

Este trabalho apresenta um método rápido de inversão de matrizes densas, e uma possível aplicação com métodos de *Vectoring*, em pré-codificação e cancelamento de *crosstalk* de sistemas xDSL. A família de tecnologias xDSL utiliza os pares trançados de fios de cobre telefônicos como meio físico para transmitir dados digitais. O *crosstalk* é a principal causa de degradação de sinais na mais nova geração de sistemas xDSL, o G.fast, e para combatê-lo são utilizadas técnicas de pré-codificação e cancelamento, chamadas de *Vectoring*. O método proposto, chamado de GSGR, consiste em uma abordagem diferente para o método clássico de *Squared Givens Rotations* (SGR), adequado a implementações em plataformas embarcadas de processamento digital de sinais. Foram realizados testes comparativos do método GSGR com métodos diretos clássicos de inversão, utilizando uma plataforma digital *multicore* baseada no *chip* TI DSP TMS320C6670 e a plataforma de *software* Matlab. Os resultados dos testes de inversão de matrizes usando dados reais e dados simulados mostraram que o GSGR foi superior em velocidade de execução sem apresentar perdas significativas de acurácia para a aplicação em sistemas xDSL.

**ABSTRACT**

*This study presents a fast method for dense matrix inversion, and a possible application with methods of Vectoring, in pre-coding and crosstalk cancellation of xDSL systems. The family of xDSL technologies uses twisted pairs of copper telephone wires as medium to transmit digital data. Crosstalk is the major cause of degradation in signals in the newest generation of xDSL systems, G.fast, and to combat it, pre-coding and cancellation techniques, called Vectoring, are used. The proposed method GSGR consists of a different approach to the classic method of Squared Givens Rotations (SGR), suitable for implementations in embedded digital signal processing platforms. Comparison testing of the GSGR with classical direct inversion methods were performed using a digital multicore platform based on TI DSP chip TMS320C6670 and Matlab software platform. The results of the matrix inversion tests using real and simulated data showed that the GSGR was superior in execution speed without significant loss of accuracy for the application on xDSL systems.*

## 1 INTRODUÇÃO

A resolução de sistemas de equações lineares é comumente realizada por métodos de inversão de matrizes em vários problemas práticos de comunicações, de processamento de sinais, de ciência dos materiais, de teoria de controle, de formação de feixe de radar (*beam forming*), e muitos outros [1][5].

Em sistemas de comunicações, na estimação de canal e na equalização por exemplo, a inversão de matrizes frequentemente fornece um resultado ótimo, porém envolve operações complexas para um equipamento de *hardware* embarcado. Para reduzir o custo computacional dessas operações os projetistas procuram continuamente propor algoritmos sub-ótimos adequados às condições específicas de cada aplicação. Para isso foram desenvolvidas diversas técnicas de cálculo como as baseadas em decomposição LU e em decomposição QR, que podem ser mais ou menos eficientes conforme as características da estrutura da matriz considerada, tais como se é bem ou mal condicionada, simétrica, esparsa, densa, etc. [1][2].

Para atingir um alto desempenho nos atuais sistemas eletrônicos digitais, os algoritmos precisam ainda ser otimizados para rapidez no processo de inversão, utilização de menos recursos computacionais e acurácia dos resultados suficiente para a aplicação [20][21].

Este trabalho propõe um método de inversão de matrizes inicialmente buscado para aplicação a sistemas de comunicação em plataformas embarcadas com foco em alta velocidade e acurácia moderada dos resultados, podendo ser usado em outras aplicações embarcadas que apresentem necessidades equivalentes.

Para os atuais sistemas de comunicação de banda larga a fibra óptica é uma solução ótima em termos de capacidade de transmissão de dados, porém em termos de implantação é uma solução lenta, devido ao alto investimento necessário, e por envolver trabalhos de construção consideráveis. Uma solução alternativa consiste na busca contínua de melhoria da qualidade de transmissão dos sistemas DSL (*Digital subscriber lines*) existentes [7].

Os sistemas DSL são baseados nas instalações de linhas telefônicas de pares de fio de cobre (*loops*), e fazem a conexão entre os equipamentos dos usuários (CPEs – *customer premise equipments*) e as redes de fibra óptica de alta capacidade. Como os sistemas DSL podem utilizar as instalações telefônicas existentes, o custo de implantação é minimizado em comparação ao de um sistema completo com fibra óptica [10].

Assim sendo, os operadores continuam a buscar soluções de curto e médio prazo para utilização das instalações de pares de fio de cobre existentes em velocidades mais altas, e então postergar ou dividir o trabalho e investimento em redes ópticas ao longo de períodos de

tempo maiores e conforme o aumento da demanda por serviços em cada área a ser abrangida [7].

A aplicação aqui considerada é em sistemas de comunicação xDSL com métodos de cancelamento de *crosstalk*, também chamados de métodos de *Vectoring*. Os principais métodos para aplicação de *Vectoring* utilizam operações de inversão ou decomposição matricial para processar as matrizes de ganho de canal xDSL, em atividades de pré-codificação e de cancelamento de *crosstalk* [9].

A complexidade dos cálculos envolvidos na inversão de matrizes em sistemas de comunicação xDSL pode ser ilustrada considerando-se que para um ponto de distribuição em um sistema VDSL, para 192 usuários e 4096 tons de modulação, seria necessário obter a inversão de 4096 matrizes de números complexos de  $192 \times 192$ , além de periodicamente atualizá-las de acordo com as alterações das características do canal e número de linhas ativas [9]. Embora sejam empregadas técnicas como interpolação e cancelamento parcial para reduzir esses requisitos, à medida que a banda de frequência de transmissão é ampliada para elevar a capacidade de transmissão - como é o caso da próxima geração da família de tecnologias DSL atualmente em desenvolvimento pelo ITU-T, o *G.fast*, essas técnicas de aproximação se tornam menos eficazes em virtude da diminuição da esparsidade e das características dominantes das matrizes, como por exemplo, os valores da diagonal deixam de ser muito superiores aos demais [27].

No Capítulo 2 são descritos alguns dos principais métodos de inversão matricial estudados para o desenvolvimento deste trabalho. O Capítulo 3 apresenta alguns dos conceitos das tecnologias DSL necessários para situar a aplicação desenvolvida. Em sequência o Capítulo 4 apresenta as características, os meios de medição e estimação dos elementos dessas matrizes e os requisitos de tempo de inversão e de acurácia relativos às matrizes *G.fast*. O Capítulo 5 apresenta o método de decomposição QR e os melhoramentos introduzidos para obter o método proposto, chamado de GSGR. O Capítulo 6 apresenta os resultados de simulações e testes da aplicação do método GSGR a conjuntos de medidas reais e simuladas de canais na banda de frequências de *G.fast*. Finalmente, o Capítulo 7 apresenta as considerações finais para este trabalho e possíveis propostas de estudos futuros.

## 2 REVISÃO DO ESTADO DA ARTE DE MÉTODOS PARA INVERSÃO MATRICIAL

Os métodos para cálculo de inversão matricial podem ser classificados entre métodos iterativos e diretos [29].

Entre os métodos iterativos clássicos pode-se destacar o de *Gauss Seidel* e o de *Jacobi*. Esses métodos normalmente requerem uma estimativa inicial da matriz inversa, e a cada iteração atualizam os valores dos elementos conforme o erro prévio. Como desvantagens, esses métodos necessitam de uma razoável estimativa inicial e apresentam cálculos matriciais sequenciais de alta complexidade de implementação.

Por outro lado os métodos diretos como o da eliminação de *Gauss*, *Householder* e *QR* por exemplo, se caracterizam por possuir um número finito de operações, ainda que de considerável complexidade, e podem ser bem ajustados para execução paralela e otimizada nos atuais *hardwares* dedicados.

Para matrizes mais densas e quadradas, como as matrizes encontradas no caso deste trabalho, a triangularização é o principal passo em quase todos os métodos diretos para resolver sistemas lineares, embora ele seja ainda de alto custo computacional. Os algoritmos para execução de triangularização requerem tipicamente um número de operações da ordem de  $n^3$  operações em matrizes de dimensão  $n \times n$ .

O processo de triangularização, conversão de um sistema linear quadrado para um sistema triangular com a mesma solução, pode ser executado por métodos tradicionais como a decomposição LU, que é uma representação matricial do método clássico de eliminação de Gauss. Para sistemas lineares determinados ou sobre-determinados, como é o caso da aplicação em análise, o procedimento mais confiável envolve a redução por fatoração da matriz  $\mathbf{A}$  [1], isto é a matriz  $\mathbf{A}$  é representada por um produto de outras matrizes, as quais podem ser mais facilmente invertidas por meio de transformações ortogonais.

O método direto de resolução preferido quando a matriz  $\mathbf{A}$  é quadrada, densa e sem uma aparente estrutura dominante, como é o caso das matrizes encontradas em canais *G.fast* de alta frequência, é o método de eliminação de Gauss na forma da decomposição LU.

Neste capítulo é apresentado o problema de resolução de sistemas lineares na seção 2.1. Nas seções em sequência de 2.2 e 2.3 são apresentados alguns métodos diretos e iterativos de solução de sistemas lineares estudados durante o desenvolvimento deste trabalho. Na seção 2.4 são apresentadas as características do padrão IEEE 754/2008 utilizadas para

representação dos coeficientes das matrizes em ponto flutuante na plataforma de *hardware* embarcada.

## 2.1 RESOLUÇÃO DE SISTEMAS LINEARES

O problema de resolução de um sistema linear representado por:

$$\mathbf{A} \mathbf{x} = \mathbf{b} \quad (2.1)$$

é fundamental na computação científica e é comum em um grande número de aplicações de comunicações e de processamento de sinais [1][2].

O problema de mínimos quadrados – *least square* (LS) é equivalente à resolução de um sistema de equações lineares, também chamadas equações normais.

Uma solução exata para (2.1) é definida como:

$$\mathbf{x} = \mathbf{A}^{-1} \mathbf{b} \quad (2.2)$$

onde  $\mathbf{A}$  é uma matriz  $n \times n$ ,  $\mathbf{x}$  e  $\mathbf{b}$  são vetores de dimensão  $n$ , e  $\mathbf{A}^{-1}$  é a matriz inversa de  $\mathbf{A}$ .

A complexidade do cálculo direto da inversa de uma matriz, ou seja, o número de operações necessárias para a inversão, é da ordem de  $n^3$  [1]. Portanto, busca-se continuamente por métodos que reduzam esse custo computacional. Os métodos para resolução do problema de LS podem ser classificados em duas grandes classes: métodos diretos e iterativos.

Os métodos iterativos estimam uma solução inicial e a atualizam sucessivamente baseando-se no erro da iteração prévia, até que uma solução suficientemente acurada seja alcançada. Entretanto, este procedimento sequencial não tem um número fixo de operações. Assim, torna-se um método limitado quando se pretende explorar características de paralelismo em *hardware* a fim de obter grande capacidade de processamento. Por outro lado, os métodos diretos calculam soluções em um número fixo de operações e são melhor aproximados em *hardware* de processamento paralelo.

São apresentados a seguir alguns métodos diretos e iterativos estudados durante o desenvolvimento deste trabalho e que são comumente utilizados para resolver sistemas lineares e obter a inversa de uma matriz.

## 2.2 MÉTODOS DIRETOS

A estratégia utilizada pela maioria dos métodos diretos para resolução de um sistema linear é aplicar transformações a esse sistema a fim de convertê-lo em um outro, com forma triangular superior, ou triangular inferior, mas que possua a mesma solução do sistema original. Em seguida, para solucionar o sistema triangular obtido, faz-se uso de métodos de Substituição (*Forward Substitution*) e Retrossubstituição (*Back Substitution*), conforme são descritos a seguir.

### 2.2.1 Método de Substituição

Para ilustrar a resolução de um dado sistema triangular inferior por substituição considere-se o sistema linear 2 x 2 a seguir:

$$\begin{bmatrix} l_{11} & 0 \\ l_{21} & l_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} \quad (2.3)$$

Se  $l_{11}l_{22} \neq 0$  então pode-se determinar as variáveis de forma sequencial como:

$$x_1 = b_1/l_{11} \quad (2.4)$$

$$x_2 = (b_2 - l_{21}x_1)/l_{22} \quad (2.5)$$

Por generalização para um sistema de  $n$  equações, pode-se então escrever a  $i$ -ésima variável  $x$  como:

$$x_i = \left( b_i - \sum_{j=1}^{i-1} l_{ij}x_j \right) / l_{ii} \quad (2.6)$$

### 2.2.2 Método de Retrossubstituição

Para resolução de um dado sistema triangular superior 2 x 2 por retrossubstituição considere-se o sistema linear 2 x 2 a seguir:

$$\begin{bmatrix} u_{11} & u_{12} \\ 0 & u_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} \quad (2.7)$$

Se  $u_{11}u_{22} \neq 0$  então pode-se determinar as variáveis de forma sequencial como:

$$x_2 = b_2/u_{22} \quad (2.8)$$

$$x_1 = (b_1 - u_{12}x_2)/u_{11} \quad (2.9)$$

Por generalização para um sistema de  $n$  equações, pode-se então calcular a  $i$ -ésima variável  $x$  pela expressão:

$$x_i = \left( b_i - \sum_{j=i+1}^n u_{ij}x_j \right) / u_{ii} \quad (2.10)$$

### 2.2.3 Decomposição LU

Uma decomposição matricial consiste em fatorar, ou representar, uma matriz em um produto de matrizes, com o objetivo de simplificar a solução de um problema [3]. Desta forma, se uma matriz pode ser representada como o produto de duas matrizes triangulares, uma superior e outra inferior, a solução do respectivo sistema linear é reduzida então à solução mais simples de dois sistemas triangulares.

O processo de decomposição ou fatoração LU pode ser compreendido como uma descrição algébrica de alto nível da eliminação de Gauss [1]. Esse método permite decompor a matriz  $\mathbf{A}$  do sistema de (2.1) em:

$$\mathbf{A} = \mathbf{L} \mathbf{U} \quad (2.11)$$

Em (2.11)  $\mathbf{L}$  é uma matriz triangular inferior com os elementos da diagonal iguais a 1 e  $\mathbf{U}$  é uma matriz triangular superior. Assim a solução para o vetor  $\mathbf{x}$  pode ser encontrada resolvendo-se os sistemas em (2.12) e (2.13) por substituição e retrossubstituição, respectivamente:

$$\mathbf{L} \mathbf{y} = \mathbf{b} \quad (2.12)$$

$$\mathbf{U} \mathbf{x} = \mathbf{y} \quad (2.13)$$

### 2.2.4 Decomposição de *Cholesky*

Se a matriz do sistema linear a ser resolvido é simétrica e positiva definida, o método de *Cholesky* pode ser usado com vantagem. Neste método uma matriz  $\mathbf{A}$  é decomposta na seguinte forma:

$$\mathbf{A} = \mathbf{U}^T \mathbf{U} \quad (2.14)$$

onde  $\mathbf{U}$  é uma matriz triangular superior com todos os elementos da diagonal positivos, e  $\mathbf{U}^T$  é a transposta de  $\mathbf{U}$ .

O sistema de equações em (2.1) pode assim ser escrito como:

$$(\mathbf{U}^T \mathbf{U}) \mathbf{x} = \mathbf{b} \quad (2.15)$$

Escrevendo:

$$\mathbf{U} \mathbf{x} = \mathbf{y} \quad (2.16)$$

obtem-se um sistema triangular inferior dado por:

$$\mathbf{U}^T \mathbf{y} = \mathbf{b} \quad (2.17)$$

O vetor  $\mathbf{x}$  pode então ser encontrado resolvendo-se as equações (2.16) e (2.17) pelos métodos de retrossubstituição e de substituição respectivamente.

Sendo a matriz  $\mathbf{A}$  positiva definida, então ela também é não singular, e portanto o método de *Cholesky* é estável para resolução de (2.1).

Entretanto, a complexidade desse método ainda é alta para implementação embarcada e em tempo real, sendo da ordem de  $n^3/3$  [1].

### 2.2.5 Decomposição QR

Na decomposição QR representa-se uma matriz  $\mathbf{A} \in \mathbf{R}^{m \times n}$  pelo produto de uma matriz  $\mathbf{Q}$ , cujas colunas são uma base ortonormal de  $\mathbf{A}$ , por uma matriz triangular superior  $\mathbf{R}$ :

$$\mathbf{A} = \mathbf{Q} \mathbf{R} \quad (2.18)$$

Pode-se provar que a partir de qualquer matriz com colunas linearmente independentes pode ser obtida uma matriz ortonormal [3]. Outra interessante característica resultante desta decomposição é que a diagonal da matriz  $\mathbf{R}$  é composta somente de elementos positivos.

O sistema em (2.1) pode assim ser transformado em um sistema triangular:

$$\mathbf{Q}^T (\mathbf{Q} \mathbf{R}) \mathbf{x} = \mathbf{Q}^T \mathbf{b} \quad (2.19)$$

$$\mathbf{R} \mathbf{x} = \mathbf{Q}^T \mathbf{b} \quad (2.20)$$

A solução para  $\mathbf{x}$  em (2.20) pode ser obtida pelo método de retrosubstituição.

A decomposição QR descrita acima pode ser executada por muitos métodos baseados em transformações lineares. Para sistemas lineares sobredeterminados, como os encontrados em matrizes de sistemas xDSL, onde  $\mathbf{A} \in \mathbf{R}^{m \times n}$ , com  $m \geq n$  e  $\mathbf{b} \in \mathbf{R}^m$ , a solução para a minimização de  $\|\mathbf{Ax} - \mathbf{b}\|_2$  mais confiável envolve a redução da matriz  $\mathbf{A}$  a uma forma canônica por meio de transformações ortogonais [1], tais como reflexões de *Householder* e rotações de *Givens*.

O método de rotações de *Givens* (*Givens Rotations*) tem vantagens tais como paralelização em *hardware* dedicado mais fácil comparado ao método de *Householder*, e também uma versão sem operações de raiz quadradas e divisões: o método *Squared Givens Rotations* (SGR). O SGR, proposto por Dohler [4], é composto apenas por operações de multiplicação e adição, e portanto, mais adequado a plataformas de *hardware* embarcadas.

### 2.3 MÉTODOS ITERATIVOS

Em contraponto com os métodos diretos estão os métodos iterativos. A estratégia dos métodos iterativos é buscar a cada iteração uma melhor solução, de forma a convergir para uma solução ótima. Entre as vantagens dos métodos iterativos estão a possibilidade de aplicação em sistemas esparsos e de grande dimensão com bastante eficiência, e ainda a utilização essencialmente apenas de operações de multiplicação entre vetores e matrizes.

Para resolver um sistema linear  $\mathbf{Ax} = \mathbf{b}$  por métodos iterativos, assumindo-se que a matriz  $\mathbf{A} \in \mathbf{R}^{n \times n}$  tem elementos não nulos na diagonal principal, como é o caso da aplicação escolhida neste trabalho, e  $\mathbf{b} \in \mathbf{R}^n$ , é necessário uma estimativa inicial para  $\mathbf{x}$ , um vetor  $\mathbf{x}^{(0)}$  em  $\mathbf{R}^n$  que aproxime a solução. A partir de  $\mathbf{x}^{(0)}$  gera-se outra estimativa  $\mathbf{x}^{(1)}$ , depois gera-se  $\mathbf{x}^{(2)}$ , e assim sequencialmente com o objetivo de convergir para a solução verdadeira  $\mathbf{x}$ , ou uma solução  $\mathbf{x}^{(k)}$ , após  $k$  iterações, suficientemente próxima  $\mathbf{x}$ .

Entre os principais tipos de métodos iterativos estão os estacionários e os não estacionários. Os métodos estacionários clássicos para resolução de sistemas lineares são os métodos de *Jacobi*, *Gauss-Seidel* e *SOR* (*Successive Overrelaxation* – sobre-relaxação sucessiva). Comparados aos métodos não estacionários, os estacionários são mais simples, porém menos efetivos. Os métodos não estacionários são de desenvolvimento mais recente, incluindo os métodos de descida, o método de gradiente conjugado (CG) e outros, sendo a maioria de grande complexidade para implementação em plataformas embarcadas, embora altamente efetivos em matrizes esparsas e de grande dimensão [4]. A seguir apresentamos

brevemente os métodos clássicos de *Jacobi*, *Gauss-Seidel* e SOR estudados ao longo da pesquisa.

### 2.3.1 Método de *Jacobi*

O método de *Jacobi* é talvez o método iterativo mais simples [1]. Este método é aplicável para matrizes que tem elementos não nulos na diagonal principal. Pode-se especificar este método descrevendo como obter o resultado  $\mathbf{x}^{(k+1)}$  da iteração  $k+1$  a partir do resultado  $\mathbf{x}^{(k)}$  da iteração  $k$ . Considerando-se então a  $i$ -ésima equação de  $\mathbf{A} \mathbf{x} = \mathbf{b}$ , e reescrevendo-a para  $x_i$ :

$$\sum_{j=1}^n a_{ij}x_j = b_i \quad (2.21)$$

$$x_i = \frac{1}{a_{ii}} \left( b_i - \sum_{j \neq i} a_{ij}x_j \right) \quad (2.22)$$

Substituindo-se  $x_j$  por  $x_j^{(k)}$  obteremos um novo valor para  $x_i$ , igual a  $x_i^{(k+1)}$  por definição:

$$x_i^{(k+1)} = \frac{1}{a_{ii}} \left( b_i - \sum_{j \neq i} a_{ij}x_j^{(k)} \right) \quad (2.23)$$

Repetindo-se o processo para cada uma das  $i$  equações,  $i = 1, \dots, n$ , obtém-se  $\mathbf{x}^{(k+1)}$  pelo Método de *Jacobi*, com a esperança de que após repetidas iterações o resultado convirja para a solução verdadeira de  $\mathbf{x}$ .

### 2.3.2 Método de *Gauss-Seidel*

No método de *Jacobi* não é utilizado o valor mais recente de cada  $x_j$  ao calcular-se  $x_i^{(k+1)}$ . Por exemplo ao calcular-se  $x_2^{(k+1)}$ , utiliza-se o valor de  $x_1^{(k)}$  mesmo que  $x_1^{(k+1)}$  já seja conhecido.

O Método de *Gauss-Seidel* é definido usando-se o mesmo processo de *Jacobi*, porém de uma forma sequencial e utilizando-se as estimativas mais recentes. Assim, ao calcular-se  $x_i$  utiliza-se os valores já calculados de  $x_1^{(k+1)}, x_2^{(k+1)}, \dots, x_{i-1}^{(k+1)}$ , obtendo-se:

$$x_i^{(k+1)} = \frac{1}{a_{ii}} \left( b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k+1)} - \sum_{j=i+1}^n a_{ij}x_j^{(k)} \right) \quad (2.24)$$

Comparado ao Método de *Jacobi*, o Método de *Gauss-Seidel* precisa apenas armazenar uma cópia de  $\mathbf{x}$ , facilitando ainda a programação pois armazena os novos valores  $\mathbf{x}^{(k+1)}$  logo após serem calculados, sem precisar manter os anteriores  $\mathbf{x}^{(k)}$  até o término da iteração  $k+1$ . Por outro lado o Método de *Jacobi* pode ser executado simultaneamente de forma paralela, enquanto o Método de *Gauss-Seidel* precisa ser adaptado para execução paralela.

### 2.3.3 Método SOR (*Successive Overrelaxation*)

O processo de cálculo, ou correção, da estimativa de  $\mathbf{x}$  em cada equação é às vezes chamado de relaxamento (*relaxation*), por analogia com uma montagem de peças que não se ajustam corretamente, e estão em um estado de tensões entre si. Cada correção das variáveis de  $\mathbf{x}$  é uma tentativa de relaxar a tensão entre as peças. Desta forma, o método de *Gauss-Seidel* executa relaxamentos sequencialmente e o método de *Jacobi* pode executar relaxamentos simultâneos.

A estratégia do método de sobre relaxamento (SOR) é tentar acelerar a convergência para a solução por meio de uma correção maior em cada iteração. Na técnica mais simples escolhe-se um fator de relaxação  $w > 1$  e corrige-se o resultado por esse fator a cada passo:

$$x_i = \frac{1}{a_{ii}} \left( b_i - \sum_{j \neq i} a_{ij} x_j \right) \quad (2.25)$$

$$\delta = \hat{x} - x_i \quad (2.26)$$

$$x_i = x_i + w \delta \quad (2.27)$$

Escolhendo-se um valor apropriado para  $w$  a convergência pode ser até dez vezes mais rápida em comparação com o método de *Gauss-Seidel* [5]. Porém esse ajuste de  $w$  em geral depende dos coeficientes da matriz  $\mathbf{A}$ , e para escolhê-lo apropriadamente pode ser necessário uma análise sofisticada dos autovalores de  $\mathbf{A}$  [1].

## 2.4 CÁLCULO EM PONTO FLUTUANTE

Cálculos em computadores e processadores digitais são executados aproximando-se um conjunto infinito de números reais por um conjunto finito de números.

O padrão IEEE 754 / 2008 [6] fornece instruções para execução de cálculos em ponto flutuante em sistemas computacionais, que produzam resultados dependentes apenas dos valores de entrada, da operação e da destinação, e não dependentes da plataforma de *hardware* e/ou *software* utilizada.

O padrão especifica:

- formatos para dados em ponto flutuante binários e decimais;
- formatos para operações de adição, subtração, multiplicação, divisão, multiplicação e adição conjuntas, raiz quadrada e outras;
- conversões entre formatos de números inteiros e de ponto flutuante;
- conversões entre formatos diferentes de números em ponto flutuante;
- conversões de formatos em ponto flutuante para intercâmbio de dados por sequências de caracteres;
- tratamento de ocorrências de exceções nas operações com números em ponto flutuante.

Os formatos em ponto flutuante do padrão IEEE para representar um subconjunto finito dos números reais são caracterizados por um radical, um valor de precisão e uma faixa exponencial. São definidos 5 formatos básicos. Três formatos são binários de comprimento 32, 64 e 128 bits. E dois formatos são decimais, de comprimentos de 64 e de 128 bits.

As representações de dados em ponto flutuante desse padrão consistem de:

- trincas de números compostas de sinal, expoente e “significando” ou coeficiente, associadas a um radical  $b$ , de valor igual a 2 ou 10, da seguinte forma:

$$(-1)^{\text{sinal}} \times b^{\text{expoente}} \times \text{coeficiente} \quad (2.28)$$

- representação para  $+\infty$  e  $-\infty$ ;
- representação para exceções NaN (*not a number*).

Desta forma o padrão IEEE 754 descreve um mapeamento da representação em ponto flutuante para uma representação em uma cadeia de bits. Essa operação de codificação pode resultar em mais de um tipo de representação em cadeia de bits, por exemplo, pode-se obter diferentes tipos de representação variando o expoente.

O conjunto finito de números representável por um dado formato é determinado pelos seguintes parâmetros inteiros:

- radical  $b$ , de valor 2 ou 10;
- número  $p$  de dígitos do “significando”;

- número máximo do expoente  $emax$ ;
- número mínimo do expoente  $emin$ , definido como igual a  $1 - emax$  em todos os formatos.

Os valores possíveis para os parâmetros em cada formato básico são apresentados na Tab. 2.1 abaixo:

Tabela 2.1 Parâmetros dos formatos básicos de números em ponto flutuante do padrão IEEE 754/2008

Parâmetros	Formatos				
	b = 2 (formato binário)			b = 10 (formato decimal)	
<b>Radical b</b>	b = 2 (formato binário)			b = 10 (formato decimal)	
<b>Nº de bits</b>	32	64	128	164	128
<b>Nº de dígitos <math>p</math></b>	24	53	113	16	32
<b>Expoente máximo <math>emax</math></b>	+127	+1023	+16383	+384	+6144

Os dados em ponto flutuante abaixo são representados em cada um dos formatos:

- números em ponto flutuante, inclusive zero, com sinal, na forma:

$$(-1)^s \times b^e \times m \quad (2.29)$$

onde  $s$  é igual a 0 (número positivo) ou 1 (número negativo);  $e$  é o expoente tal que  $emin \leq e \leq emax$ ;  $m$  é igual a  $d_0 . d_1 . d_2 \dots d_{p-1}$ ; e  $d_i$  é um dígito inteiro tal que  $0 < d_i < b$ .

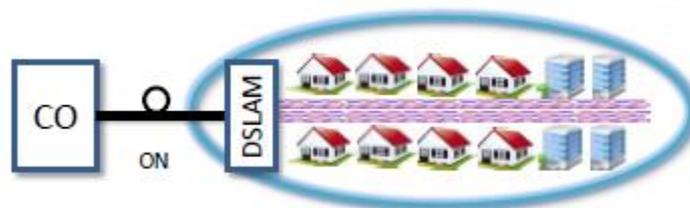
- $+\infty$  e  $-\infty$ ;
- exceções NaN.

Considerando-se as características da aplicação estudada e dos formatos de representação em ponto flutuante disponíveis na plataforma DSP, escolheu-se a representação binária de 32 bits.

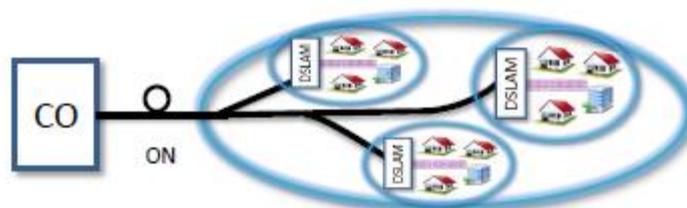
### 3 REVISÃO DO ESTADO DE ARTE DAS TECNOLOGIAS DE LINHA DIGITAL PARA ASSINANTE (*DIGITAL SUBSCRIBER LINE - XDSL*)

A família de tecnologias xDSL (*Digital subscriber line - DSL*) foi desenvolvida para prover serviços de transmissão digital de dados em altas velocidades por meio das instalações de linhas telefônicas de pares de fios de cobres existentes, sem que ocorram interferências perceptíveis ou interrupções no serviço telefônico. Assim é possível a prestação de serviços telefônicos de forma simultânea com os serviços de transmissão de dados com baixos custos de implantação.

Vários modelos de desenvolvimento de redes de acesso com tecnologia xDSL foram concebidos para obter capacidades de transmissão mais próximas daquelas possíveis com sistemas de fibra óptica, mas com as vantagens da economia e facilidade de instalação que podem ser obtidas pela exploração da infraestrutura de pares de fios de cobre telefônicos existentes. Entre os modelos de desenvolvimento mais modernos e respectivos comprimentos de *loops* de fios de cobre estão: o FTTN (*fiber to the node* – fibra até ponto de acesso) com comprimentos da ordem de 500m; FTTdp (*fiber to de distribution point* – fibra até ponto de distribuição) de cerca de 100m; e FTTB (*fiber to the building* – fibra até prédio) e FTTH (*fiber to the home* – fibra até domicílio), esses últimos com fibra até a entrada da instalação de prédio ou domicílio [7].



(a)



(b)

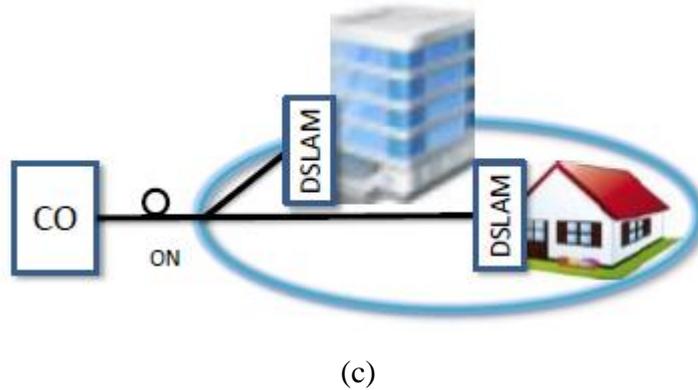


Figura 3.1 Ilustrações de modelos de rede: (a) FTTN; (b) FTTdp; (c) FTTB e FTTH.

CO – *central office* – instalações do provedor de serviços de comunicações.

ON – *optical network* – rede de fibra óptica.

DSLAM – *DSL access multiplexer* – multiplexador de acesso à rede DSL

À medida que os *loops* se tornam cada vez mais curtos entre a terminação da rede de fibra óptica e o equipamento do usuário assinante a quantidade total de atenuação do sinal elétrico no cabeamento de cobre diminui, permitindo aumentar a largura de banda de frequências utilizadas, e conseqüentemente elevar a capacidade de transmissão de dados.

Para explorar essa característica de menor atenuação do cabeamento tem-se buscado o desenvolvimento de novas tecnologias para essa camada física da rede. Nessas novas configurações de canais de menor atenuação a interferência de *crosstalk* FEXT (*far end crosstalk* – interferência causada por um transmissor de uma linha de assinante em um receptor de outra linha, dentro de um mesmo cabeamento) é o principal fator de degradação do sinal, podendo alcançar níveis superiores aos sinais de ruído.

A tecnologia xDSL estabelecida mais recentemente é a VDSL2. A VDSL2 utiliza um método chamado de *Vectoring*, de processamento em conjunto (multi-usuário) dos sinais em um cabeamento de várias linhas de usuários, para redução da interferência de *crosstalk*. Esse novo método permite elevar consideravelmente as taxas de transmissão nos modelos de rede FTTN e FTTdp, para mais de 100Mbps.

Em 2009 foi apresentado pela primeira vez na revista científica *IEEE Communications Magazine* [7][8] o conceito de desenvolvimento de tecnologia em fios de cobre para banda larga em velocidades superiores a 1 Gb/s ( $10^9$  bits por segundo) para redes FTTdp. Esse novo e importante conceito motivou consideráveis interesses e investimentos em pesquisas tanto pela academia quanto pela indústria de telecomunicações, e impulsionou o início de diversos estudos de padronização atualmente em desenvolvimento para o projeto completo da nova tecnologia [7].

Em 2011 o ITU (*International Telecommunication Union*) - agência especializada das Nações Unidas para assuntos de tecnologias de informação e comunicação - iniciou o projeto de padronização do protocolo da camada física de rede dessa próxima geração de tecnologias xDSL, chamada de *G.fast*. O *G.fast* está sendo desenvolvido para os modelos FTTdp e FTTB, com *loops* de até aproximadamente 250 metros, e almeja-se atingir taxas de transmissão da ordem de 1 Gb/s para um perfil de banda de frequências de transmissão até 106MHz, tanto na transmissão de dados da rede para o usuário (*downstream*), quanto a partir da rede para o usuário (*upstream*). Em um próximo estágio de desenvolvimento, o objetivo é atingir velocidades ainda maiores com um perfil de até 212 MHz [7][27].

Embora a demanda de mercado residencial esteja ainda abaixo da ordem de gigabits por segundo, a previsão é de que o consumo continue crescendo significativamente, e será desejável atingir taxas de pico da ordem de gigabits/s. Adicionalmente, as altas taxas de transmissão do *G.fast* possibilitarão sua utilização com sucesso por pequenas empresas, e também como infraestrutura de acesso de rede por dispositivos móveis (*backhaul mobile*) [7].

Um dos principais desafios para o completo estabelecimento dessa nova tecnologia é o desenvolvimento de técnicas de *Vectoring* – para o cancelamento de *crossstalk* – aplicáveis em linhas dentro de um mesmo cabeamento de transmissão e ao longo de todo o perfil de frequências de transmissão estendido do *G.fast* [7].

### 3.1 AMBIENTE XDSL

A família de tecnologias xDSL utiliza os pares trançados de fios de cobre telefônicos como meio físico para transmitir dados digitais. As xDSL não interferem nos serviços telefônicos porque operam em bandas de frequências acima das utilizadas pelos sistemas de transmissão de voz, os quais são restritos a frequências inferiores a 4kHz (kilohertz). Para elevar a capacidade de transmissão as bandas de frequência utilizadas foram sendo ampliadas a cada nova geração DSL, por exemplo, a frequência máxima da ADSL (*Asymmetric DSL* – DSL assimétrica) é de 1,1Mhz, para a ADSL2+ é de 12MHz, e para a VDSL (*Very high speed DSL* – DSL com taxas de transmissão de bits muito altas) é de 30MHz.

Outro recurso utilizado pelas xDSL é o FDD (*frequency division duplexing* – transmissão por divisão de banda de frequência, simultaneamente nos dois sentidos), ou seja a alocação de faixas de frequências diferentes para as transmissões nos sentidos *upstream* (do usuário para a rede) e *downstream* (da rede para o usuário), sendo a largura de banda alocada para *dowstream* normalmente maior que para *upstream* – alocação assimétrica. Para o *G.fast*,

de forma a diminuir a complexidade do *hardware* para lidar com uma faixa de frequências estendida até 212MHz, optou-se por utilizar TDD (*time division duplexing* – transmissão por divisão de tempo nos dois sentidos) ao invés de FDD [9].

### 3.2 MODULAÇÃO DMT

Quanto à modulação os sistemas xDSL podem ser divididos entre sistemas modulados de portadora única (*single-carrier*) e sistemas modulados em multi-tons discretos (*discrete multi-tone* - DMT). Este trabalho foi desenvolvido considerando-se a utilização de modulação DMT, em razão de ela ter sido escolhida como padrão para o desenvolvimento dos sistemas a partir da tecnologia ADSL [10].

Em sistemas modulados por múltiplas portadoras a banda de frequência do canal é dividida em vários subcanais. enquanto na modulação por portadora única a taxa de transmissão de bits é feita constante dentro de toda a largura de banda do canal de operação conforme uma SNR de todo o canal e uma probabilidade de erro desejada, na modulação por multiportadora é possível acomodar as taxas de bits de acordo com a SNR de cada subcanal e uma probabilidade de erro desejada. A Figura 3.2 ilustra o conceito de partição da largura de banda do canal em vários subcanais:

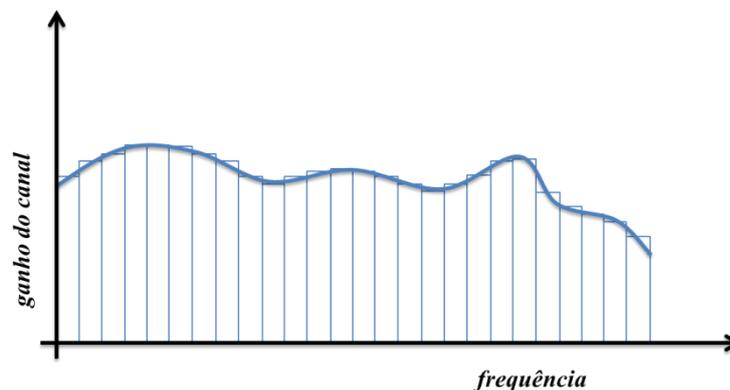


Figura 3.2 Conceito de divisão da largura de banda de frequência de um canal em subcanais, em um sistema de modulação por multiportadora.

Na Fig. 3.3 é ilustrada a SNR de um canal hipotético e uma possível distribuição de bits proporcional à SNR de cada subcanal. Uma vantagem importante nesse esquema que pode ser observada é a possibilidade de se evitar a transmissão em subcanais ruidosos ou fortemente atenuados, e utilizar taxas de transmissão mais elevadas em subcanais com SNRs superiores.

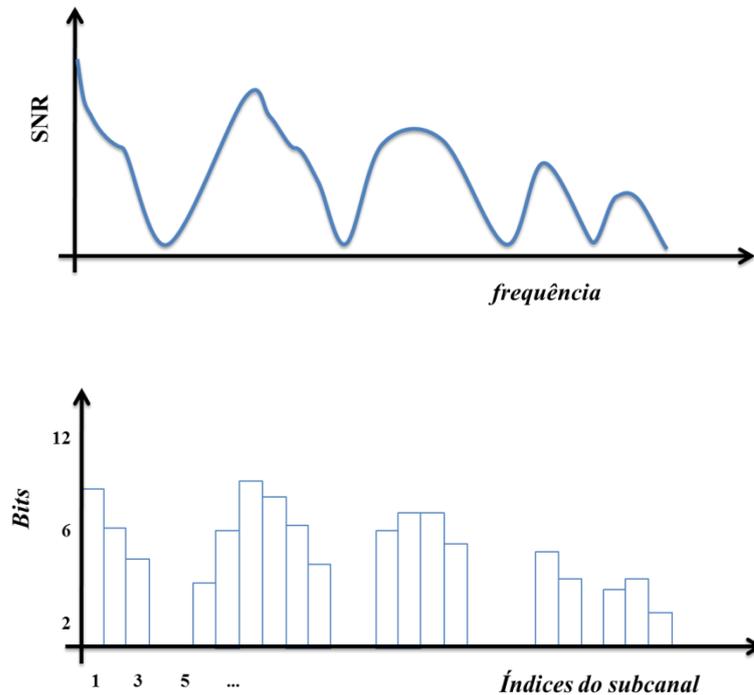


Figura 3.3 SNR de um canal hipotético e possível distribuição de *bits* por subcanal

A cada subcanal é associada uma subportadora. Um subcanal  $n$  pode assim ser considerado como um canal de portadora única independente com uma taxa de bits  $b_n$  por símbolo, proporcional à respectiva SNR e à probabilidade de erro desejada  $p_n$ . Para um sistema de  $N$  canais paralelos o total de bits por símbolo pode ser representado por:

$$B = \sum_{n=1}^{N-1} b_n \quad (3.1)$$

Considera-se que não são utilizados os canais da frequência de *Nyquist*  $N$  e o canal 0, conforme padronizado para as tecnologias DSL.

A taxa de transmissão de bits é indicada como [10]:

$$R = \frac{B}{T} \quad (3.2)$$

onde  $T$  é o período de tempo de transmissão do símbolo.

Basicamente o método de modulação por multiportadoras divide um fluxo de bits a transmitir por um canal em vários subfluxos, e os envia através de subcanais centrados em diferentes frequências ao longo do canal original. Como usualmente esses subcanais são escolhidos ortogonais, sob condições ideais de propagação, esse método é também chamado

de multiplexação por divisão de frequência ortogonal – *orthogonal frequency division multiplexing* (OFDM) [10]. O número de subfluxos é escolhido de forma a garantir que cada subcanal tenha uma largura de banda menor que a largura de banda de coerência<sup>1</sup> do canal, resultando em uma atenuação aproximadamente plana nos subcanais, e conseqüentemente em uma diminuição da interferência entre símbolos - ISI.

O método de transmissão por subportadoras pode ser implementado de várias maneiras. Na implementação discreta, também chamada de DMT (*discrete multitone* – modulação multitom ou multiportadora), a interferência ISI pode ser completamente eliminada com o uso de prefixo cíclico.

A técnica de modulação digital de transmissão DMT foi escolhida como padrão a partir da ADSL. Na modulação DMT o espectro de frequências disponível é dividido em vários subcanais estreitos com resposta em frequência mais plana, e então se transmite um número maior ou menor de bits em cada subcanal proporcionalmente à taxa sinal ruído do respectivo subcanal. Dessa forma o sistema multitom pode ser considerado equivalente a múltiplos sistemas de portadora única operando independentes e em paralelo, cada um em uma determinada frequência subportadora. As vantagens dessa técnica incluem a possibilidade de se evitar transmitir em regiões do espectro excessivamente atenuadas ou muito ruidosas, e ao mesmo tempo aproveitar ao máximo as regiões do espectro menos propensas a ruídos e interferências [10].

Uma rede xDSL com  $N$  usuários, modulação DMT com  $k$  tons de frequência (subportadoras), e assumindo-se que não há interferência entre portadoras (ICI – *inter-carrier interference*) pode ser ilustrada pela Fig. 3.4 e representada pela seguinte expressão:

$$\mathbf{y}_k = \mathbf{H}_k \mathbf{x}_k + \mathbf{z}_k \quad (3.3)$$

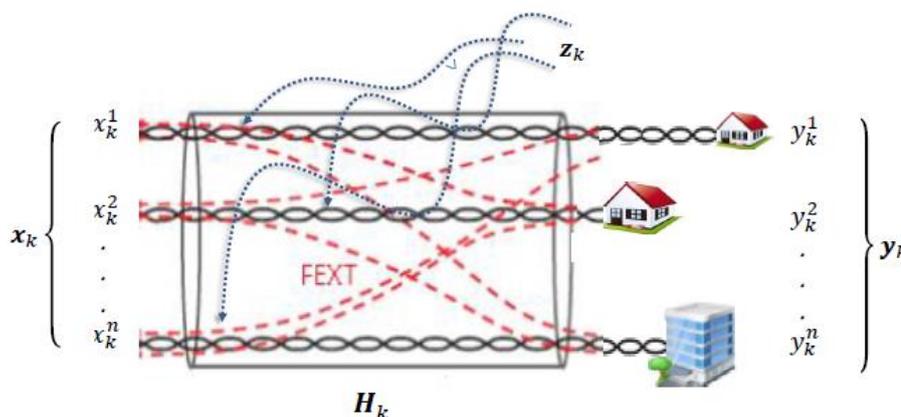


Figura 3.4. Ilustração de uma transmissão DMT em cabeamento de pares de fios de cobre

O vetor  $\mathbf{x}_k \triangleq [x_k^1, x_k^2, \dots, x_k^N]^T$  contém os sinais transmitidos para os usuários no tom de frequência  $k$ , e  $x_k^n$  é o sinal transmitido para o usuário  $n$  no tom de frequência  $k$ . Da mesma forma o vetor  $\mathbf{y}_k \triangleq [y_k^1, y_k^2, \dots, y_k^N]^T$  refere-se aos sinais recebidos pelos  $N$  usuários na frequência  $k$ , e  $\mathbf{z}_k \triangleq [z_k^1, z_k^2, \dots, z_k^N]^T$  refere-se aos sinais de ruído e interferência adicionados pelo canal físico para todos os  $N$  usuários. Em  $\mathbf{z}_k$  incluem-se ruído térmico, interferências externas (*alien crosstalk*) e interferências de radiofrequência.

A matriz  $\mathbf{H}_k$  é uma matriz de dimensão  $N \times N$  em que cada elemento  $h_k^{n,m}$  é o ganho do canal a partir do transmissor do sinal  $x_k^m$  do usuário  $m$  para o receptor do usuário  $n$  na frequência  $k$ .

A densidade de potência transmitida (PSD – *power spectral density*) pelo transmissor do usuário  $n$  na frequência  $k$  é definida como:

$$s_k^n \triangleq \mathbb{E}\{|x_k^n|^2\}/\Delta_f \quad (3.4)$$

onde  $\mathbb{E}\{\cdot\}$  indica o operador valor esperado e  $\Delta_f$  é o espaçamento de frequências entre as subportadoras.

A potência total a ser transmitida para cada usuário  $n$  sobre todos os tons de frequência é definida por um valor limite  $P^n$ . A potência limite por tom de frequência  $s_k^{n,mask}$  é chamada de máscara espectral para o usuário  $n$  no tom de frequência  $k$ .  $P^n$  e  $s_k^{n,mask}$  podem ser definidas como:

$$\Delta_f \sum_k s_k^n \leq P^n, \quad \forall n \quad (3.5)$$

$$0 \leq s_k^n \leq s_k^{n,mask}, \quad \forall n, k \quad (3.6)$$

Considerando-se os ruídos e interferências externas como fenômenos aleatórios pode-se aproximá-los, ou o seu somatório, por uma variável estatística de distribuição gaussiana, e assim pode-se definir a taxa de transmissão de bits atingível para o usuário  $n$  na frequência  $k$  como:

$$b_k^n \triangleq \log_2 \left( 1 + \frac{|h_k^{n,n}|^2 s_k^n}{\Gamma \left( \sum_{m \neq n} |h_k^{n,m}|^2 s_k^m + \sigma_k^n \right)} \right) \quad (3.7)$$

Em (3.7)  $\Gamma$  é a margem da taxa de sinal-ruído, que é uma função da taxa de erro de bits (BER – *bit error rate*) estabelecida como requisito, do ganho de codificação e da margem de

ruído. A margem  $\Gamma$  é uma forma de indicar a habilidade do sistema de suportar imperfeições que não foram antecipadas, como as causadas por flutuações de temperatura, ruídos não esperados, etc. Comumente essa margem tem sido usada para determinar quanto de aumento de ruído e interferência pode ser suportada pela linha até que a probabilidade de erro aumente acima de um valor máximo permitido.

A densidade de potência de ruído  $\sigma_k^n$  é definida como  $\sigma_k^n \triangleq \mathcal{E}\{|z_k^n|^2\}/\Delta_f$ .

A taxa de transmissão total atingível em bits por segundo para o usuário  $n$  é então:

$$R^n = f_s \sum_k b_k^n \quad (3.8)$$

Em (3.8)  $f_s$  é a taxa dos símbolos DMT.

### 3.3 CARACTERÍSTICAS ASSUMIDAS PARA O MODELO DO CANAL EM SISTEMAS XDSL

Neste trabalho foram assumidas características comumente adotadas para o desenvolvimento de sistemas xDSL. Desta forma, para a estimação de *crosstalk* em canais DSL, foi assumida a existência das seguintes características:

- utilização de modulação DMT;
- carregamento de entrada de bits (*bit-loading*) contínuo;
- estimação perfeita das características do canal;
- transmissão síncrona de um ponto de distribuição (AN) até o equipamento dos usuários;
- comprimento do prefixo cíclico longo o suficiente para que os canais possam ser considerados livres de interferências entre símbolos (ISI) e entre as subportadoras (ICI);
- ruído gaussiano e branco.

### 3.4 CROSSTALK

Em uma implementação híbrida típica são conectadas redes de fibra óptica e xDSL a partir do CO do provedor de serviços – vide Fig. 3.5. Uma rede de fibra é terminada em uma unidade óptica de rede (ONU – *optical network unit*), e na sequência os sinais são transmitidos pela infraestrutura de cobre (xDSL) até os equipamentos dos usuários (CPE – *customer premise equipment*). Entretanto usar altas taxas de dados em um canal originalmente projetado para transmissão de voz resulta em um acoplamento eletromagnético

significativo entre os pares trançados de fio de cobre agrupados em um mesmo cabeamento, comumente chamado de *crosstalk*.

Esse acoplamento eletromagnético aumenta com o comprimento do cabeamento e com a frequência, sendo de maior intensidade entre pares de fios adjacentes. Entre outras características, a função de acoplamento não muda significativamente com o tempo. Mas as várias funções de acoplamento entre pares de fios diferentes em um mesmo cabeamento não se correlacionam e variam bastante com as características e a configuração física da rede – número de pares de linhas, número de ramos, comprimento do cabeamento, características dos fios, etc. Para evitar lidar diretamente com funções de acoplamento complicadas podem ser usados sistemas com modelos simplificados dessas funções de acoplamento, ou sistemas que meçam o acoplamento entre as linhas periodicamente.

O *crosstalk* de sinais provenientes do mesmo lado do cabeamento é referido como NEXT (*near-end crosstalk*). Para evitar NEXT nas gerações xDSL anteriores emprega-se FDD, de forma que as bandas de frequência de downstream e upstream não se sobreponham. Para evitar NEXT no *G.fast*, como será empregado TDD, as transmissões serão sincronizadas e a taxa de *upstream-downstream* deverá ser a mesma para todas as linhas de um cabeamento, podendo variar todas juntas no tempo conforme a demanda [7][10].

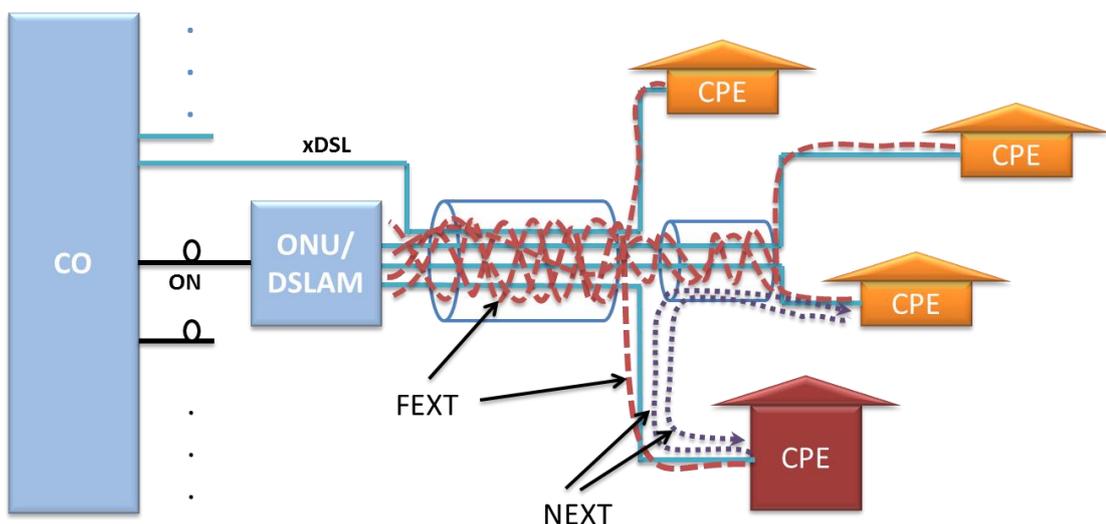


Figura 3.5 Ilustração de uma rede híbrida típica xDSL – ON e *crosstalk* FEXT e NEXT no cabeamento.

O *crosstalk* resultante do acoplamento entre os sinais recebidos e os sinais do lado oposto do cabeamento é chamado de FEXT (*far-end crosstalk*). O FEXT é a principal causa de degradação do sinal no *G.fast* e pode reduzir significativamente a capacidade do sistema. Para redução do FEXT em transmissão *upstream* são usadas técnicas de cancelamento,

normalmente baseadas no fato de que os modems receptores estão localizados fisicamente juntos na interface com a rede óptica (ONU). Na transmissão *downstream* por outro lado aproveita-se o fato de que os transmissores estão localizados fisicamente juntos na ONU, e então utilizam-se técnicas de pré-codificação para cancelamento de FEXT.

### 3.5 PRINCÍPIOS DE CANCELAMENTO DE *CROSSTALK* FEXT

Devido à natureza distribuída dos CPEs, cada um instalado na propriedade de seu respectivo assinante, normalmente todo tratamento coordenado dos sinais dos usuários precisa ser realizado por pré-codificação nos transmissores em *dowstream* e por cancelamento nos receptores em *upstream*.

A interferência FEXT gerada por uma linha de assinante  $m$  (um par trançado de fios de cobre) em uma outra linha vítima  $n$  pode ser cancelada em uma frequência subportadora  $f_i$  pela subtração do sinal interferente  $x_m(f_i)$  multiplicado pela função de transferência FEXT da linha  $m$  na linha  $n$ :  $h_{nm}(f_i)$ . Desta forma para  $N$  linhas em um cabeamento e  $K$  subportadoras de transmissão cada sinal de *crosstalk* a ser cancelado é calculado pela multiplicação do vetor  $\mathbf{x}$  de sinais a transmitir pelo vetor de acoplamento  $\mathbf{h}_m$ , ambos de dimensão  $N$ , por isso chamou-se o método de *Vectoring* [11]. O *Vectoring* para VDSL2 foi padronizado pela Rec. ITU-T 993.5 [12][13].

O método de *Vectoring* faz uma coordenação da transmissão dos sinais dos usuários em um cabeamento para reduzir a interferência de FEXT entre os pares de fios de cobre e assim melhorar o desempenho. A melhoria obtida no desempenho depende das características do canal. O *vectoring* pode ser aplicado apenas a um usuário, parte deles, ou todos os usuários, com diferentes graus de melhoria.

No sentido de transmissão *downstream* os transmissores estão posicionados fisicamente no multiplexador de acesso de DSL (DSLAM), e assim os sinais podem ser processados de forma a pré-compensar a distorção que será sofrida durante a transmissão pelo cabeamento. Para a transmissão no sentido de *upstream* os receptores é que estão posicionados no DSLAM, e assim os sinais recebidos das CPEs são processados para cancelamento do *crosstalk* introduzido durante a transmissão [28] – vide Fig. 3.6.

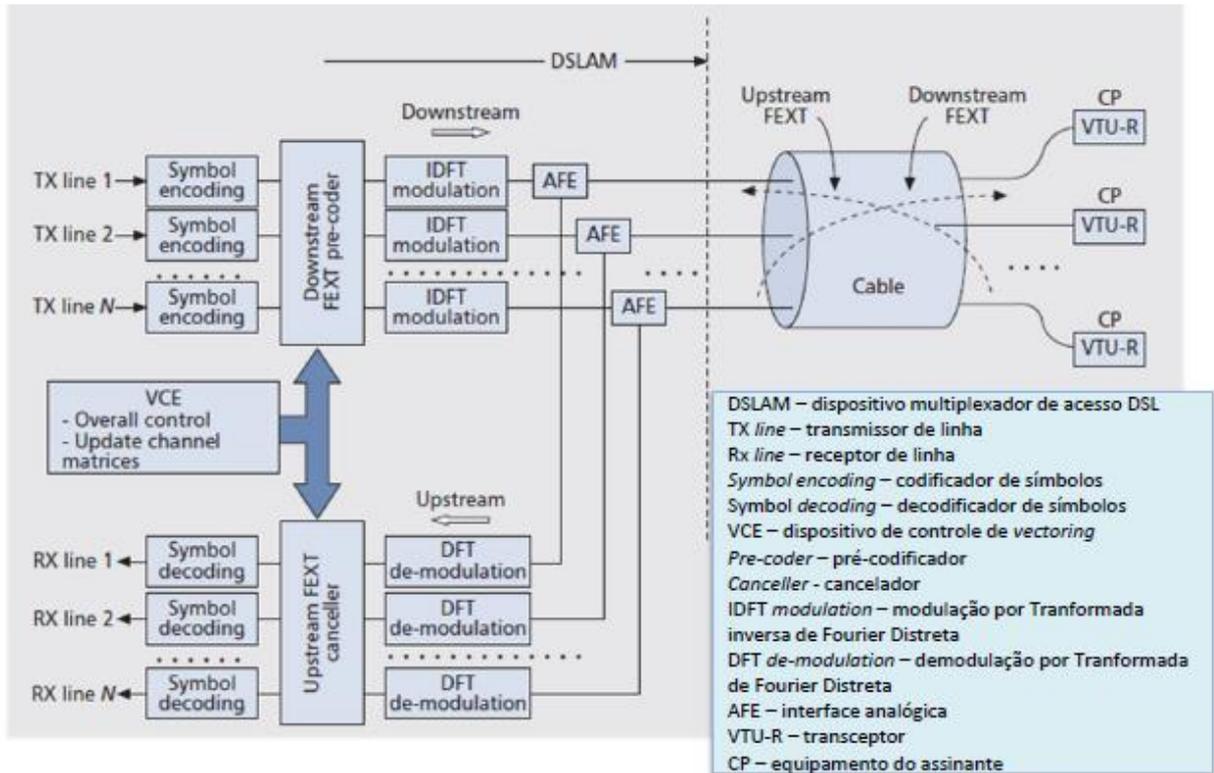


Figura 3.6 Descrição funcional do método de *Vectoring* em *downstream* e *upstream* [11].

A pré-compensação e o cancelamento são calculados de forma a eliminar, ou reduzir consideravelmente, o *crosstalk* introduzido pelo cabeamento, e assim atingir uma taxa de transmissão muito mais alta do que seria possível sem a pré-compensação ou o cancelamento.

A Rec. ITU-T G.993.5 [12][13] apresenta um modelo de referência – parcialmente reproduzido na Fig. 3.7 - para um sistema “vetorizado” (com uso de *vectoring*) para VDSL2, que está sendo usado como ponto de partida para o sistema vetorizado para *G.fast*.

Nesse modelo, a transmissão coordenada e a recepção coordenada dos sinais das linhas do grupo é realizada no chamado ponto de acesso à rede (AN) e os sinais das linhas são representados em um vetor onde cada componente é o sinal de uma das linhas. A coordenação das linhas é gerenciada por um sistema de gerenciamento de rede (NMS), uma entidade de gerenciamento (ME) e entidades de controle de *vectoring* (VCEs). Cada VCE controla diretamente um único grupo vetorizado e os transceptores VTU-O-n de cada linha *n* do grupo. Os blocos PHY representam a camada física da interface do AN com a rede, e da interface de NT (terminação de rede) com o equipamento do usuário (CP). Os blocos L2+ representam interfaces com uma camada de rede Ethernet para outras funcionalidades específicas do sistema, indicados no modelo apenas mostrar o fluxo de dados.

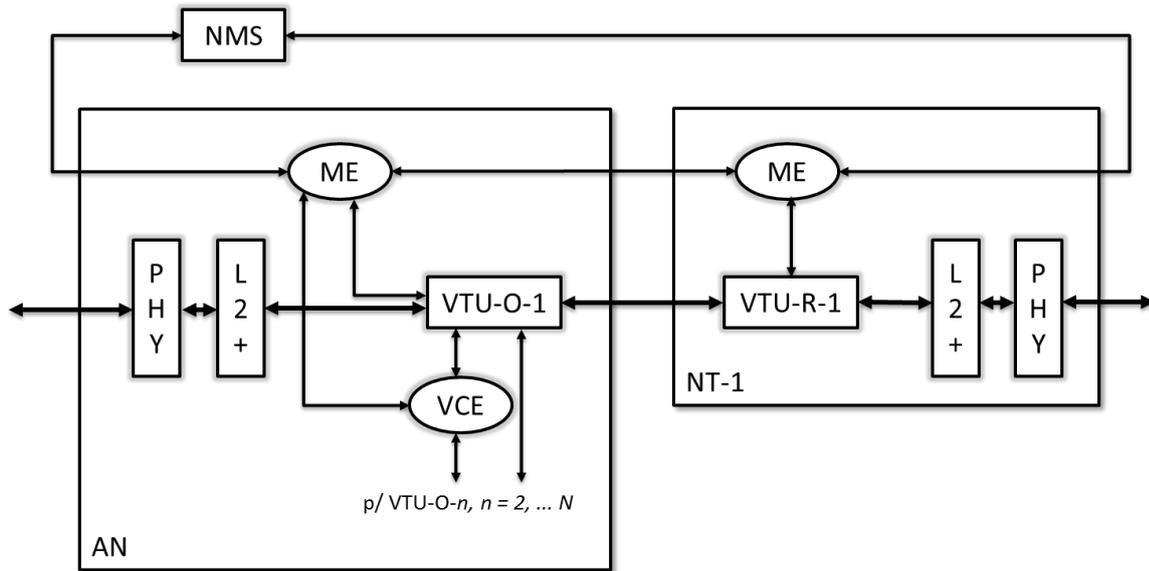


Figura 3.7 Modelo de referência para um sistema com *vectoring*, ilustrado para a linha  $n = 1$  de um grupo de  $N$  linhas.

Entre os principais métodos de aplicação de *Vectoring* em *downstream* pode-se citar o pré-codificador linear *Zero-Forcing* (ZF) [14] e em *upstream* pode-se citar o cancelador *Zero-Forcing* [15], descritos brevemente a seguir.

### 3.6 TRANSMISSÃO *DOWNSTREAM* COM VECTORING

A Rec. 993.5 [12,13] define que a pré-codificação seja realizada no AN na parte da DSLAM da Fig. 3.6, para combater o FEXT em *downstream*. A Fig. 3.8 reproduzida parcialmente de [12] ilustra o modelo funcional de um sistema de  $N$  linhas vetoradas. Em cada linha de usuário é incluído um pré-codificador de  $N$  entradas, resultando no sistema de  $N$  pré-codificadores que compõem o pré-codificador de cancelamento de FEXT.

Como será descrito nas próximas seções, uma matriz de pré-codificação  $\mathbf{P}$  é calculada a partir da matriz de característica do canal  $\mathbf{H}$ , e usada para compensar o FEXT de cada linha do grupo. Na Fig. 3.8 esse processo é ilustrado pela matriz de pré-codificadores de dimensão  $N \times N$ , para cada subportadora. É assumido ainda que todo o processamento dos valores de  $\mathbf{H}$  e  $\mathbf{P}$  é realizado no AN.

Periodicamente é necessário a troca de informações entre os transceptores de cada linha para identificação e manutenção dos valores da matriz de canal  $\mathbf{H}$  e sua respectiva matriz  $\mathbf{P}$ .

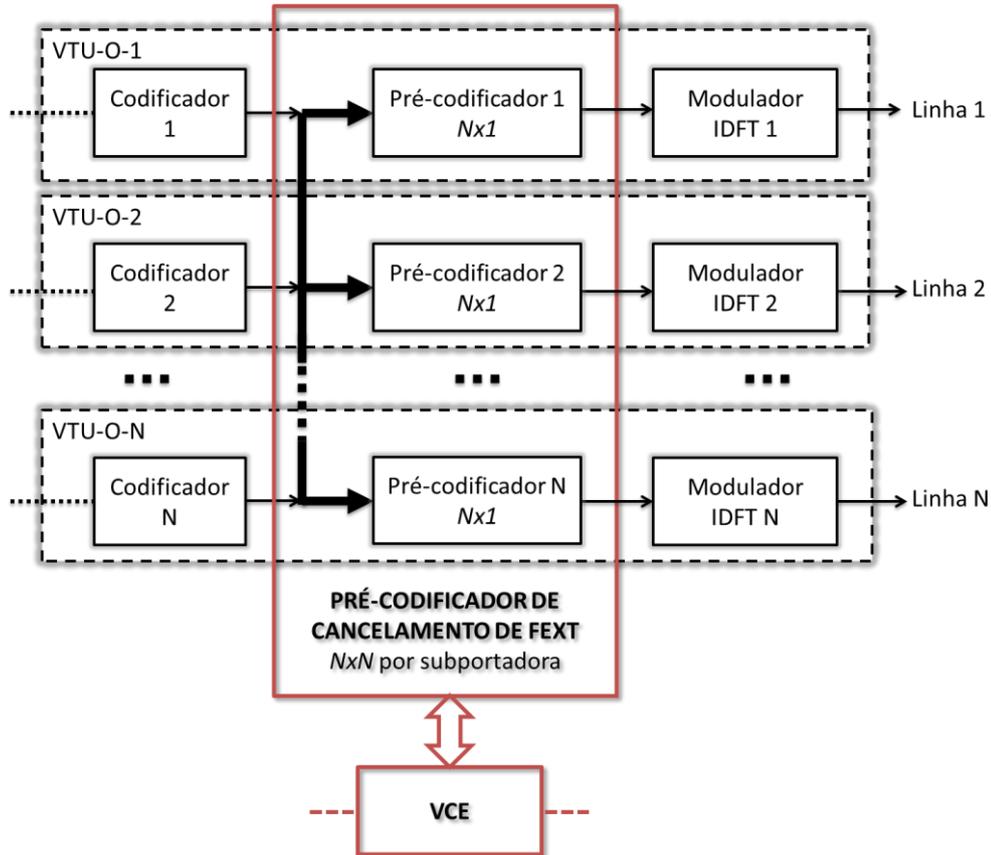


Figura 3.8 Ilustração do modelo funcional de um grupo de linhas vetoradas usando pré-codificador de cancelamento de FEXT em *downstream*

### 3.6.1 Pré-codificador *Zero-Forcing*

O pré-codificador ZF utiliza a inversa da matriz do canal  $\mathbf{H}$  como matriz pré-codificadora, para cada subportadora  $k$ :

$$\mathbf{P}_k \triangleq \mathbf{H}_k^{-1} \quad (3.9)$$

O sinal recebido pelo transceptor do usuário é dado então por:

$$\mathbf{y}_k = \mathbf{H}_k(\mathbf{P}_k \mathbf{x}_k) + \mathbf{z}_k = \mathbf{H}_k(\mathbf{H}_k^{-1} \mathbf{x}_k) + \mathbf{z}_k \quad (3.10)$$

Embora o método ZF tenha o potencial de cancelar a interferência no sinal, pode resultar em aumento da potência total a transmitir e exceder os limites disponíveis de PSD, especialmente se a matriz é mau condicionada. Para garantir que os limites de PSD não sejam excedidos [14] propôs um fator de escala  $\beta$  para a matriz pré-codificadora. O fator de escala garante que o sinal após ser aplicada a pré-codificação permaneça dentro dos limites de

potência disponíveis. O fator de escala para cada usuário é obtido a partir da definição das restrições de potência por usuário  $n$  e por subportadora  $k$ :

$$s_k^{n,mask} > \mathcal{E}\{|x_k^n|^2\} \quad (3.11)$$

Após a pré-codificação a PSD original  $s_k^{n,mask}$  se torna  $\tilde{s}_k^{n,mask}$ :

$$\tilde{s}_k^{n,mask} = \mathcal{E}\left\{\frac{1}{\beta_k^n} \left| [\mathbf{H}_k^{-1}]_{n,m} x_k^n \right|^2\right\} = \frac{1}{\beta_k^n} \sum_m \left| [\mathbf{H}_k^{-1}]_{n,m} \right|^2 s_k^{n,mask} \quad (3.12)$$

Assim pode-se escolher o fator de escala idêntico para todos os usuários como:

$$\beta_k \triangleq \max_n \{\beta_k^n\} = \max_n \left\{ \sum_m \left| [\mathbf{H}_k^{-1}]_{n,m} \right|^2 \right\} \quad (3.13)$$

O sinal efetivamente recebido quando da aplicação do pré-codificador ZF com o fator de escala será então:

$$\mathbf{y}_k = \frac{1}{\beta_k} \mathbf{H}_k (\mathbf{P}_k \mathbf{x}_k) + \mathbf{z}_k = \frac{1}{\beta_k} \mathbf{x}_k + \mathbf{z}_k \quad (3.14)$$

Como o ganho efetivo do canal em (3.14) depende do fator de escala mais restrito, [14] propôs o pré-codificador DP (*diagonalizing precoder*), que obtém um desempenho quase ótimo, dado por:

$$\begin{aligned} \mathbf{y}_k &= \frac{1}{\beta_k^{diag}} \mathbf{H}_k (\mathbf{P}_k^d \mathbf{x}_k) + \mathbf{z}_k = \frac{1}{\beta_k^{diag}} \mathbf{H}_k (\mathbf{H}_k^{-1} \text{diag}(\mathbf{H}_k) \mathbf{x}_k) + \mathbf{z}_k \\ \mathbf{y}_k &= \frac{1}{\beta_k^{diag}} \text{diag}(\mathbf{H}_k) \mathbf{x}_k + \mathbf{z}_k \end{aligned} \quad (3.15)$$

O fator de escala do pré-codificador DP é então dado por:

$$\beta_k^{diag} = \max_n \left\{ \sum_m \left| [\mathbf{H}_k^{-1}]_{n,m} h_k^{m,m} \right|^2 \right\} \quad (3.16)$$

### 3.7 TRANSMISSÃO UPSTREAM COM VECTORING

A configuração física de uma rede xDSL em transmissão *upstream*, com os transmissores posicionados em locais diferentes, torna quase impraticável fazer uma pré-codificação nos transmissores. Mas, como os sinais de um grupo são recebidos no mesmo

ponto de acesso (AN) ou DSLAM, pode-se coordenar essa recepção para cancelar ou mitigar o *crosstalk* do sinal de cada usuário, conforme ilustrado na Fig. 3.9.

A seguir descreve-se brevemente um dos principais métodos de cancelamento de FEXT, em que o método de inversão proposto também pode ser utilizado com sucesso.

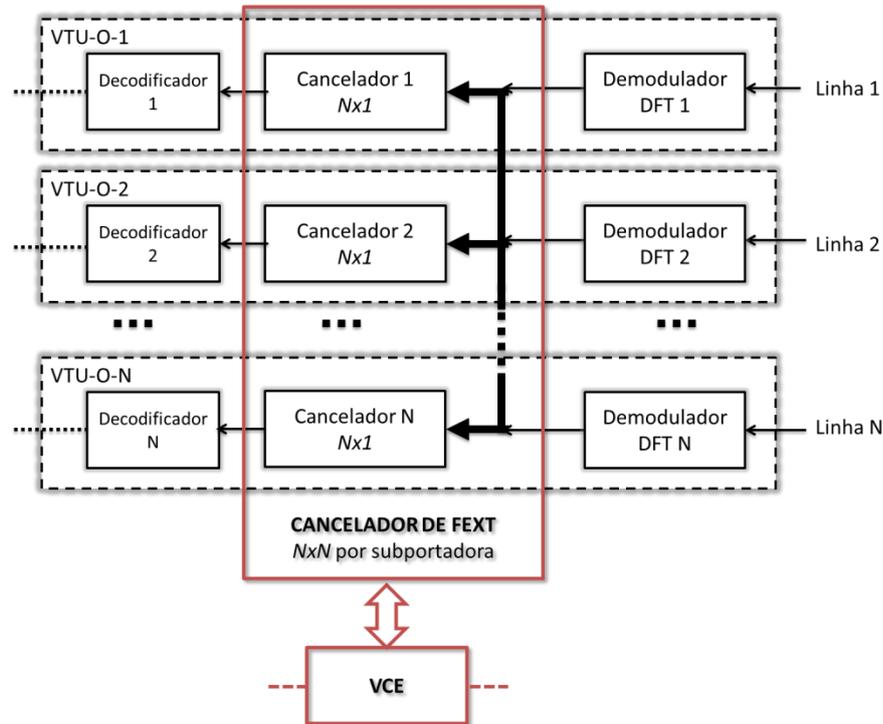


Figura 3.9 Ilustração da transmissão *upstream* com *vectoring*.

### 3.7.1 Cancelador *Zero-Forcing* (ZF)

O cancelador ZF também utiliza a inversa da matriz do canal  $\mathbf{H}$ , no caso como matriz de cancelamento, para cada subportadora  $k$ . Assim a aplicação do cancelador ZF ao sinal recebido de um transceptor VTU-R, em uma transmissão em *upstream*, pode ser representado como:

$$\tilde{\mathbf{y}}_k = \mathbf{Q}_k \mathbf{y}_k = \mathbf{Q}_k (\mathbf{H}_k \mathbf{x}_k + \mathbf{z}_k) = \mathbf{H}_k^{-1} (\mathbf{H}_k \mathbf{x}_k + \mathbf{z}_k) = \mathbf{x}_k + \mathbf{H}_k^{-1} \mathbf{z}_k. \quad (3.17)$$

Embora o cancelador ZF tenha o potencial de cancelar o *crosstalk* em *upstream* ele também eleva a PSD do ruído, especialmente se a matriz é mau condicionada. A potência de ruído original  $\sigma_k^n$  se torna:

$$\tilde{\sigma}_k^n = \varepsilon \{ |\mathbf{H}_k^{-1}(n, :) \mathbf{z}_k|^2 \} / \Delta f = \| [\mathbf{H}_k^{-1}(n, :)] \|^2 \sigma_k^n. \quad (3.18)$$

As matrizes de canal VDSL em *upstream* têm característica de diagonal dominante CWDD (*column wise diagonal dominance*), ou seja, em cada coluna da matriz o elemento da diagonal tem valor bastante superior aos demais. Isto se deve ao fato dos receptores estarem posicionados próximos na transmissão *upstream*, de os sinais de *crosstalk* provenientes de uma linha  $i$  em outra linha  $j$  serem atenuados durante todo o percurso de transmissão da linha  $i$ , e ainda pela isolação entre os condutores, resultando na característica de que  $|h_k^{i,i}| \gg |h_k^{j,i}|$  para  $i \neq j$ . Então se a matriz é CWDD pode-se aproximar a PSD do ruído após o cancelamento a partir de (3.18) como:

$$\tilde{\sigma}_k^n \approx \frac{\sigma_k^n}{|h_k^{n,n}|^2} \quad (3.19)$$

Observando-se (3.19), como  $|h_k^{n,n}|^2 < 1$  a PSD modificada é maior que a original.

## 4 MATRIZES DE CANAL DE SISTEMAS XDSL *G.FAST*

### 4.1 CARACTERÍSTICAS DAS MATRIZES *G.FAST* CONSIDERADAS

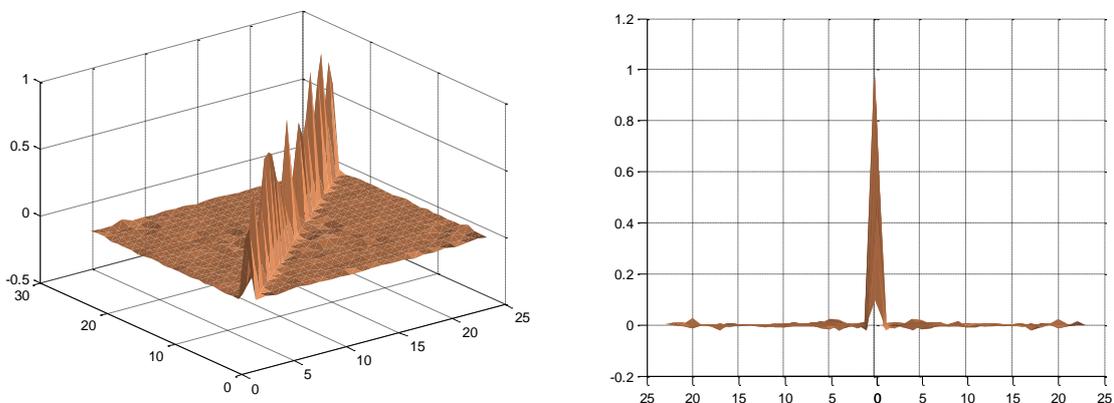
Uma das premissas para o desenvolvimento do *G.fast* é o número reduzido de usuários por ponto de distribuição em razão de suas menores distâncias de transmissão, de até 250 metros. Considerou-se neste trabalho um número máximo de 64 usuários por ponto de distribuição, resultando em matrizes de canal de dimensão máxima igual a 64.

No modelo adotado para o desenvolvimento de xDSL os elementos da diagonal de  $\mathbf{H}$  correspondem aos ganhos diretos do canal, e os demais correspondem aos valores de *crosstalk* nas linhas. Assume-se ainda um número  $N$  de linhas, ou usuários, e portanto,  $\mathbf{H}$  é uma matriz quadrada  $N \times N$ .

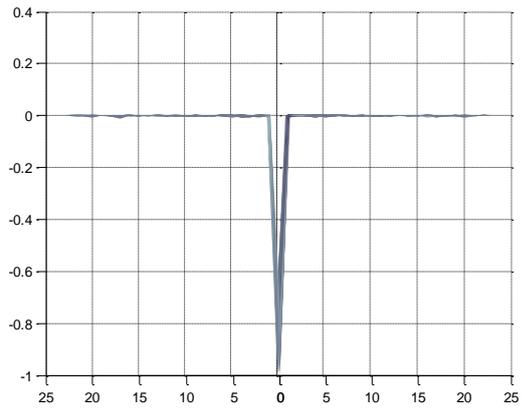
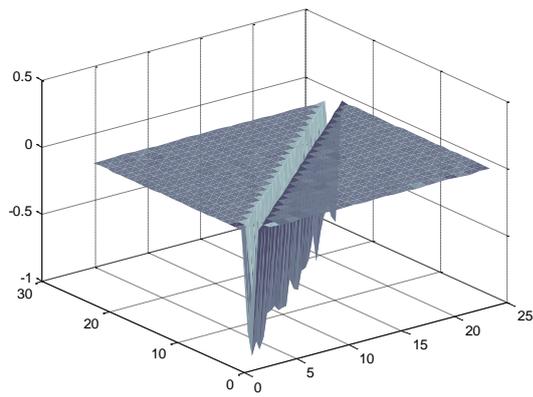
As matrizes de *G.fast* a serem invertidas têm ainda a característica de valores da diagonal principal dominante nas baixas frequências. Pode-se escrever que uma matriz  $\mathbf{H} = [h_{ij}]$  tem a diagonal fortemente dominante se:

$$|h_{ii}| > \sum_{i \neq j} |h_{ij}| \quad (4.1)$$

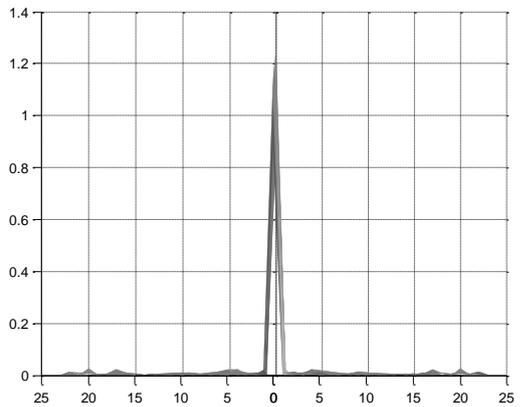
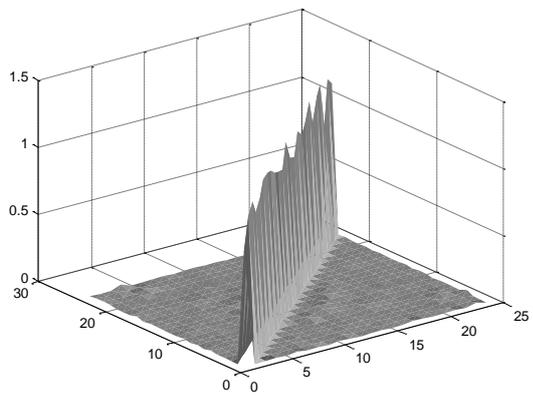
A Fig. 4.1 mostra uma representação tridimensional de uma matriz de canal real na frequência de 5,17 MHz. Na Fig. 4.2 é mostrado o somatório de matrizes na faixa de frequências de 5,17 MHz a 7,76 MHz, onde pode-se observar que a característica de diagonal dominante ainda prevalece.



(a)

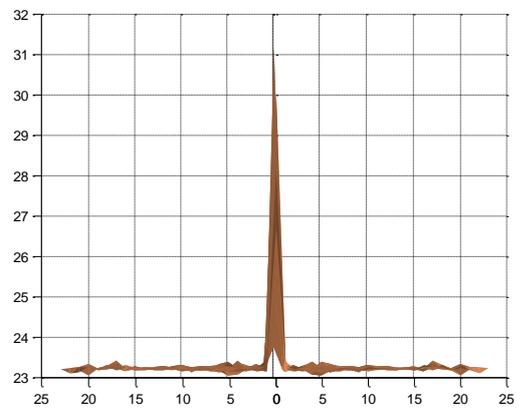
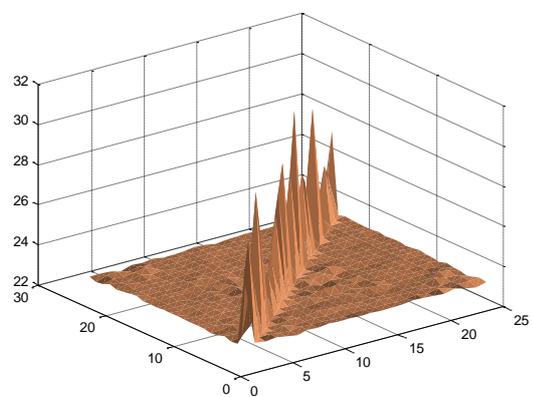


(b)



(c)

Figura 4.1 Representação de uma matriz de canal *G.fast*, de 24 x 24 elementos de medições normalizadas, na frequência de 5,17 MHz. (a) valores reais; (b) valores imaginários; e (c) valores absolutos.



(a)

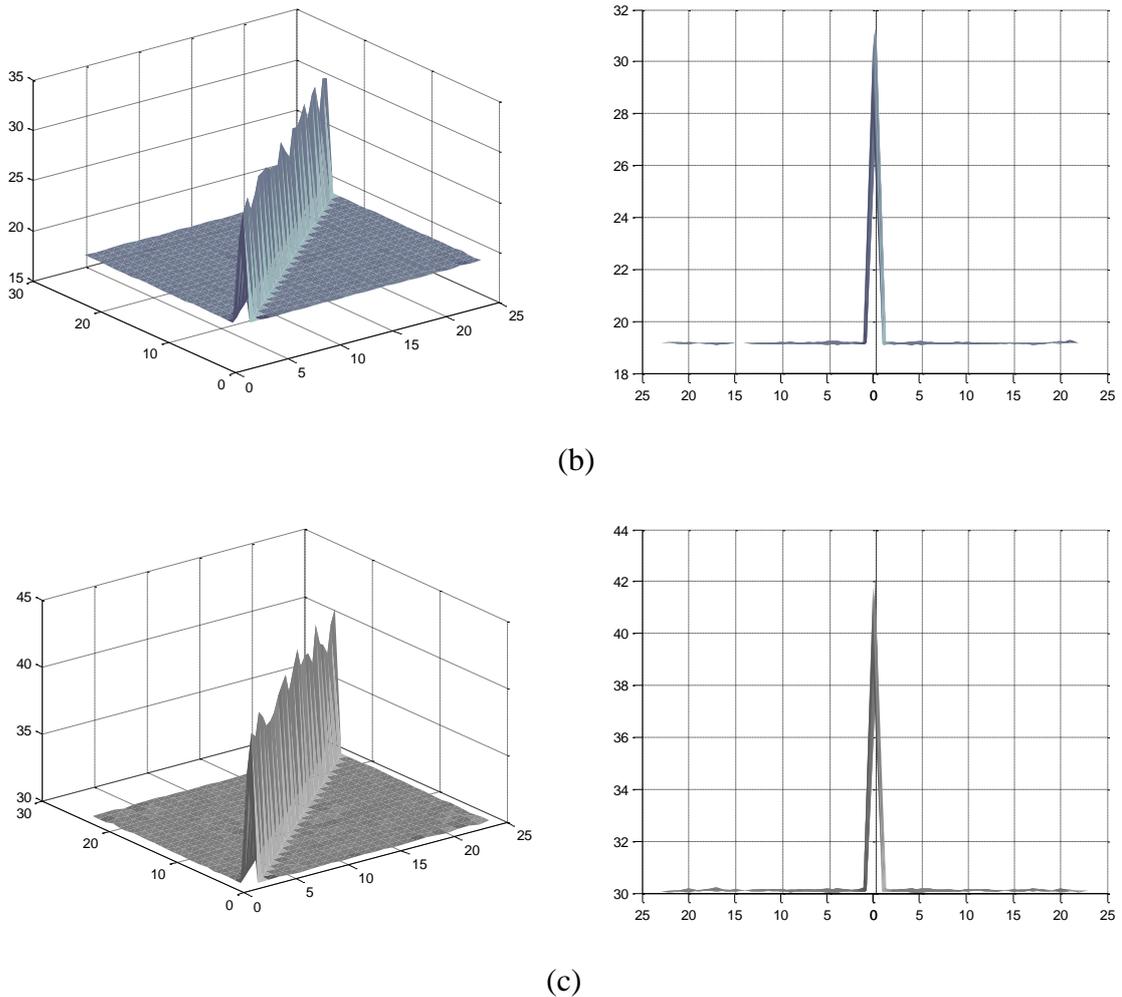
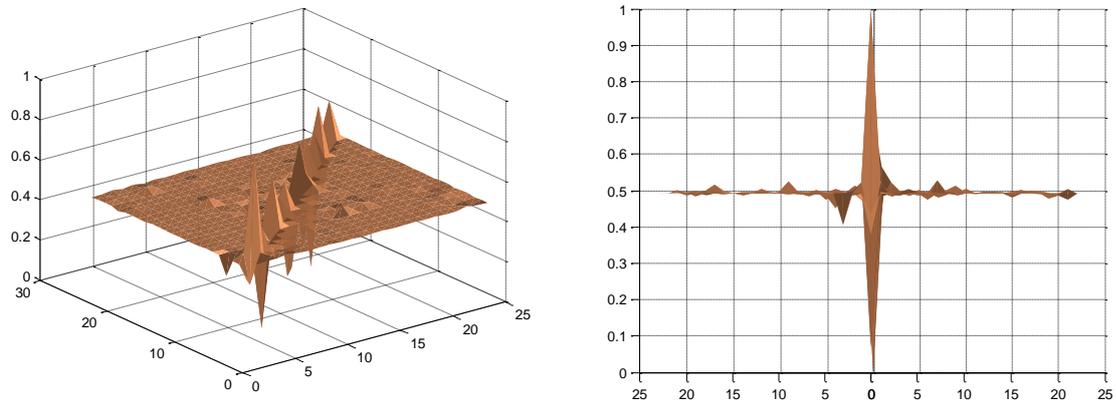


Figura 4.2 Representação do somatório de 51 matrizes de canal *G.fast*, de 24 x 24 elementos de medições normalizadas, na faixa de frequências de 5,17 MHz a 7,76 MHz. (a) valores reais; (b) valores imaginários; e (c) valores absolutos

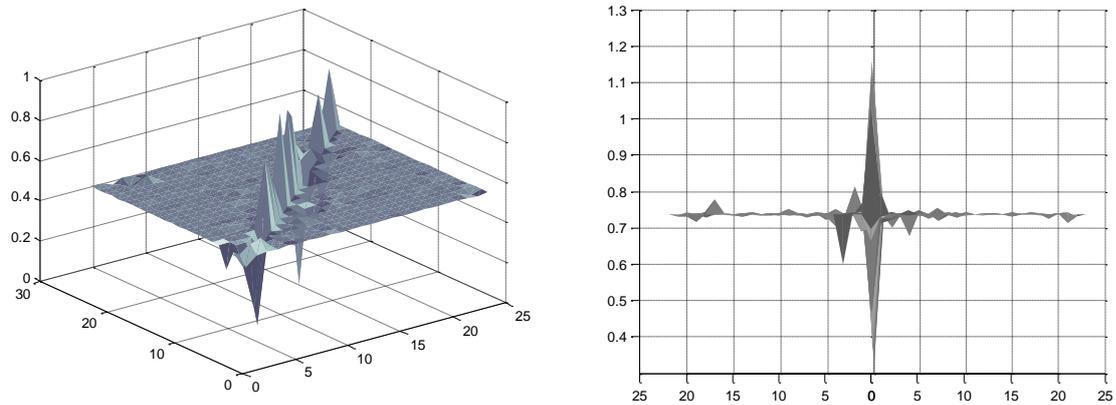
Por outro lado, nas matrizes de frequências mais elevadas, especialmente acima de 200 MHz, as diagonais principais se tornam fracamente dominantes. Para uma matriz  $\mathbf{H} = [h_{ij}]$  com diagonal fracamente dominante pode-se escrever que:

$$\max_{\substack{i,j \\ i \neq j}} \frac{|h_{ij}|}{|h_{ii}|} < 1 \quad (4.2)$$

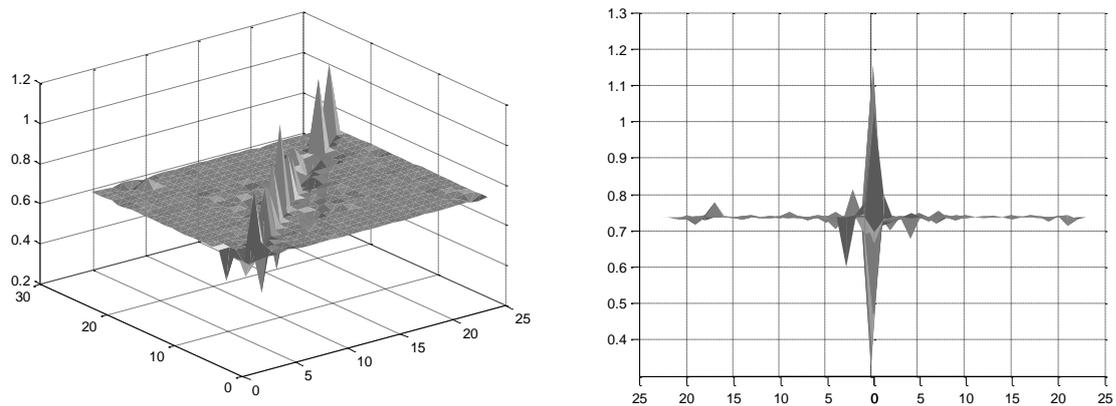
A Fig. 4.3 mostra uma representação tridimensional de uma matriz de canal real na frequência de 211,91 MHz, onde verifica-se o aumento de todos os elementos da matriz, ou seja o aumento de *crossstalk* em frequências maiores. Na Fig. 4.4 é mostrado o somatório de matrizes na faixa de frequências de 207 MHz a 211,91 MHz, onde se pode observar que a característica é de diagonal fracamente dominante.



(a)



(b)



(c)

Figura 4.3 Representação de uma matriz de canal *G.fast*, de 24 x 24 elementos de medições normalizadas, na frequência de 211,91 MHz. (a) valores reais; (b) valores imaginários; e (c) valores absolutos.

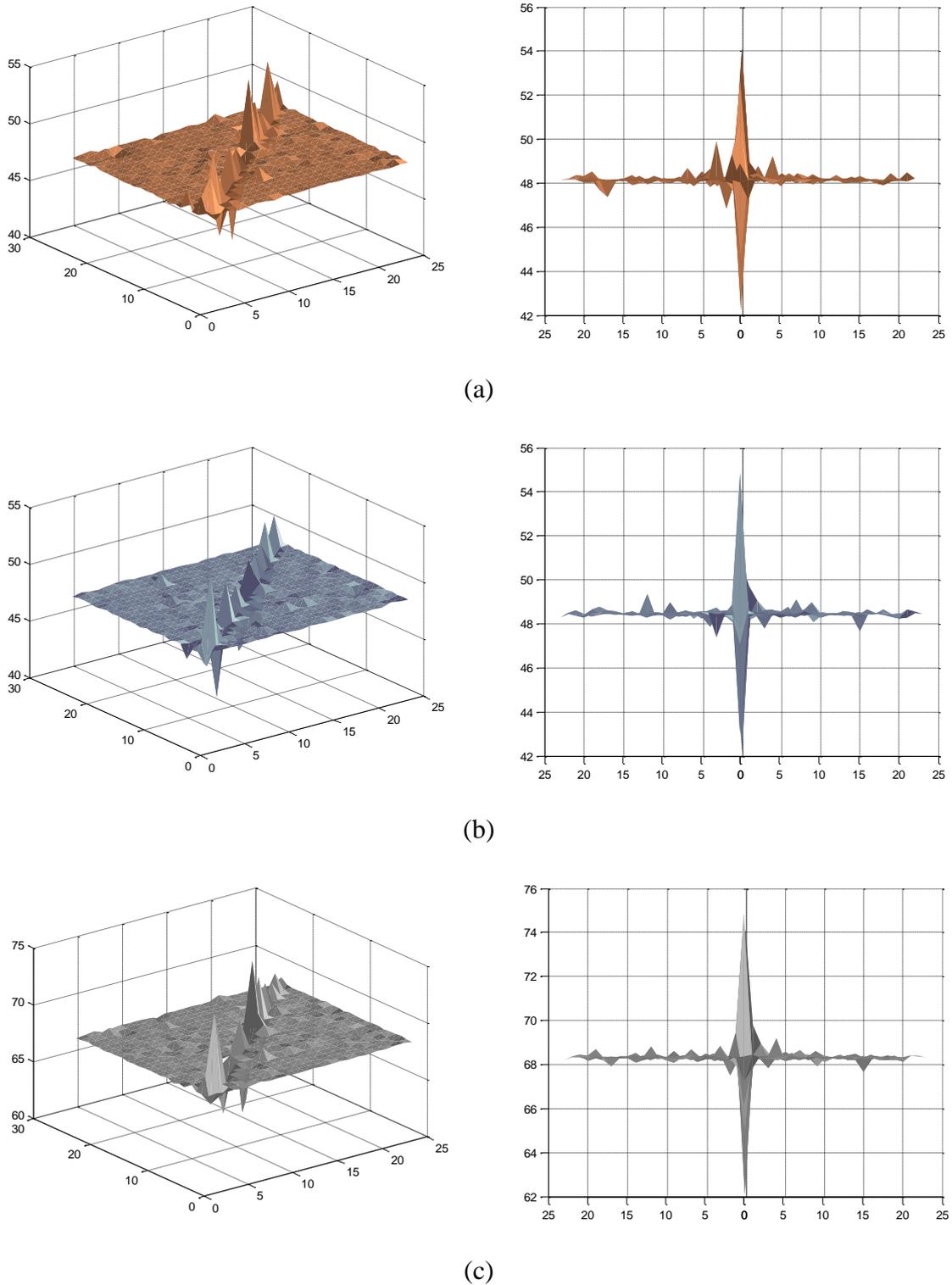


Figura 4.4 Representação do somatório de 97 matrizes de canal de *G.fast*, de 24 x 24 elementos de medições normalizadas, na faixa de frequências de 206,95 MHz a 211,91 MHz. (a) valores reais; (b) valores imaginários; e (c) valores absolutos

## 4.2 ESTIMAÇÃO DAS MATRIZES DE CANAL *G.FAST*

Em um sistema em funcionamento as matrizes de canal *G.fast* serão compostas por elementos obtidos pela medição de parâmetros de linha e pelos erros obtidos na transmissão de sequências de teste piloto, em similaridade com a tecnologia anterior, o VDSL2.

Parâmetros de teste são requisitados pela entidade ME, vide Fig. 3.7, na inicialização das linhas, periodicamente para identificar problemas com os *loops* de cobre e também para verificar a margem de desempenho do sistema em operação, ou após um reparo. Entre os principais parâmetros de teste está a função característica do canal  $\mathbf{H}(f)$  por subportadora.

Seria útil que medições de parâmetros como  $\mathbf{H}(f)$  fossem executadas frequentemente durante a operação normal. Entretanto esse procedimento repetido para cada linha e cada subportadora, tem um alto custo computacional para os transceptores, e para diminuí-lo a Rec. 993.2 [16] sugere que as informações completas de  $\mathbf{H}(f)$  obtidas durante a inicialização sejam associadas com medidas de  $\text{SNR}(f)$  da inicialização e da operação normal. Desta forma as linhas são monitoradas e  $\mathbf{H}(f)$  é medida apenas eventualmente.

Os elementos de FEXT de  $\mathbf{H}(f)$  são estimados pelo VCE por meio dos erros reportados pelos transceptores VTU-R durante a transmissão de sequências piloto de teste.

A função característica  $\mathbf{H}$  em uma frequência  $f$ ,  $\mathbf{H}(f)$ , é uma quantidade relacionada aos valores de impedância da fonte do transmissor e da carga no receptor. Uma definição simplificada foi adotada pela Rec. G.993.2 [16], que considera ambas as impedâncias da carga e da fonte resistivas, e iguais a  $R_n$ . A função característica de canal  $\mathbf{H}(f)$  é associada com uma rede de porta dupla, normalizada para um valor de  $R_n$ , sendo definida como de valores complexos iguais às taxas de  $U_2 / U_1$  - vide Figs. 4.5 e 4.6.

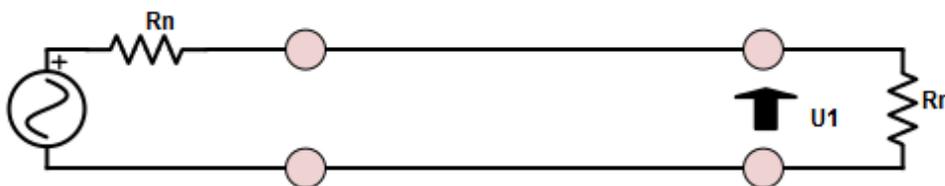


Figura 4.5 Tensão aplicada à carga  $R_n$

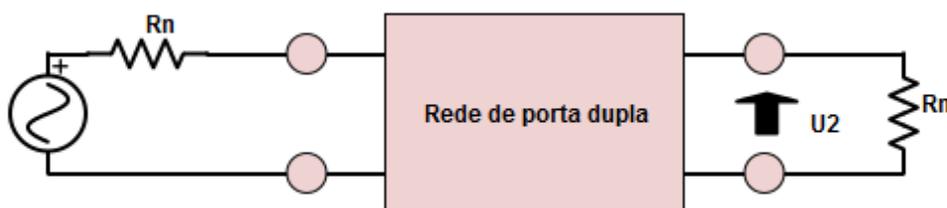


Figura 4.6 Tensão aplicada à carga  $R_n$  com a inserção de uma rede de porta dupla

As medições de  $U_1$  e  $U_2$  e, portanto, da função característica de canal são o resultado em cascata de 3 funções:

- a função característica do filtro de transmissão;
- a função característica do canal; e
- a função característica do filtro de recepção.

Espera-se que a função  $\mathbf{H}(f)$  não seja dependente das características dos filtros de recepção de diferentes fabricantes, então as medidas são corrigidas quanto ao ganho e características dos receptores.

São definidos pela Rec. G.993-2 [16] dois formatos para representar a matriz de característica do canal: a matriz  $H_{lin}(f)$ , composta por valores complexos em escala linear e  $H_{log}(f)$ , composta por valores em escala logarítmica de base 10.  $H_{lin}(f)$  e  $H_{log}(f)$  são medidos separadamente devido precisarem de correções diferentes relacionadas às características dos receptores e filtros. As medições são realizadas por um período superior a 1 segundo, e com pelo menos 256 símbolos DMT de teste.

A função característica do canal é representada no formato ( $H_{lin}$ ,  $H_{log}$ ). Assim como os parâmetros de SNR e QLN, a função ( $H_{lin}$ ,  $H_{log}$ ) pode ser definida por grupo de subportadoras, para reduzir a quantidade de medições, a quantidade de memória de armazenamento e o número de transmissões desses valores, ao invés de tomar esses parâmetros em todas as frequências subportadoras. O número de subportadoras  $G$  de um grupo é definido por (4.1), e pode assumir os valores 1, 2, 4 ou 8.

$$G = \text{pow2}(\Theta/512) \quad (4.1)$$

Em (4.1)  $\Theta$  é o maior índice de subportadoras de frequência suportado pelo transmissor, e a função  $\text{pow2}(x)$  indica o valor mais próximo da potência de 2 maior ou igual a  $x$ .

A função característica  $H_{lin}(f)$  é assim definida como:

$$H_{lin}(k \ G \ \Delta f) = \frac{\text{fator} \cdot (a(k) + jb(k))}{2^{15}} \quad (4.2)$$

Em (4.2):

- $k$  é o índice do grupo de subportadoras entre 0 e 511;
- $\Delta f$  é a largura de banda do subcanal;
- $fator$  é um fator de escala representado como um número inteiro de 16 bits sem sinal, e escolhido de forma que o valor máximo de  $|a(k)|$  ou  $|b(k)|$  em qualquer  $k$  seja  $2^{15}-1$ , para maximizar a acurácia;
- $a(k)$  e  $b(k)$  representam um número complexo normalizado, e são representados cada um deles por números de 16 bits com sinal e em complemento de 2.

A representação de  $H_{lin}$  possibilita uma granularidade de  $2^{-15}$  e uma excursão de cerca de 96 dB, de +6dB a -90dB. Valores acima de 0dB, ou seja que indicariam ganhos ao invés de perdas, são permitidos porque podem ser encontrados como resultado de variações no fabrico de receptores e de filtros.

$H_{log}$  é definido em [16] como:

$$H_{log}(k \ G \ \Delta f) = 6 - m(k) / 10 \quad (4.3)$$

Em (4.3),  $m(k)$  é um número inteiro de 10 bits sem sinal e  $k$  é o índice do grupo de subportadoras, entre 0 e 511.

A granularidade de  $H_{log}$  é de 0.1 dB e a excursão dinâmica é de cerca de 102 dB, de +6 dB a -96 dB.

#### 4.3 REQUISITOS DE TEMPO DE INVERSÃO E ACURÁCIA

O procedimento de inicialização de cada linha de usuário por meio de um par de transceptores VTU-O e VTU-R inclui entre outras funções a identificação do canal e a troca de informações de parâmetros de transmissão, tais como de *bit loading* e de tabelas de ganho de canais.

A linha de tempo ilustrada na Fig. 4.7 conforme [16] apresenta uma visão geral do procedimento de inicialização, composto por 4 fases: *Handshake*, Identificação do canal, Treinamento e Análise do canal.

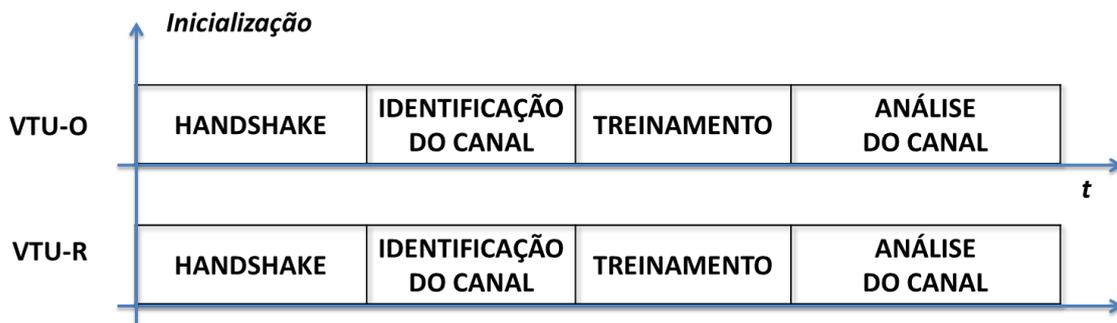


Figura 4.7 Visão geral do procedimento de inicialização

Os procedimentos recomendados para estabelecimento de comunicação inicial – *handshake* – entre os transceptores DSL são descritos na Rec. ITU-T G.994-1 [17]. Essa recomendação define os sinais, mensagens e procedimentos iniciais para o estabelecimento de comunicação automática entre os equipamentos de DSL.

Durante a fase de Análise do canal (*Channel analysis and exchange*), os transceptores fazem medições das características do canal e trocam informações dos parâmetros que serão utilizados durante a operação normal do sistema - *showtime*.

Um período de tempo máximo é definido para cada fase para evitar a suspensão do procedimento de inicialização. Desta forma, a demora superior ao tempo definido como *time-out* ou a incapacidade para completar alguma atividade resultará na interrupção do processo de ativação. O procedimento de inicialização é então interrompido imediatamente após qualquer dos seguintes eventos:

- *Time-out* de qualquer fase;
- Perda ou atividade incompleta durante alguma fase;
- Violação do protocolo de inicialização durante qualquer fase;
- Detecção de mais de 250 ms de silêncio não esperado.

Em todas as fases um contador de *time-out* deve ser iniciado quando o VTU entra em uma fase. Após completada uma fase, esse contador é reiniciado.

Para a identificação da matriz do canal, durante a fase de identificação do canal (*channel discovery*) o tempo de espera máximo (*time-out*) é definido como igual a 10 segundos.

As Rec. ITU-T G.993.5 [12] e G.997.1 [18] definem parâmetros de configuração e parâmetros de teste que devem ser disponibilizados. A seguir são descritos os parâmetros mais relevantes para o presente trabalho.

### 4.3.1 Coeficientes de acoplamento FEXT em *downstream*

O ganho de inserção de FEXT resultante da transmissão de uma linha L2 em uma linha L1 em *downstream*, na frequência  $f$ , é indicado como  $FEXT\_IG\_DS_{L1L2}(f)$ , e é igual à tensão FEXT recebida em uma carga de 100 ohm na linha L1 dividida pela tensão transmitida para uma carga também de 100 ohm na linha L2.

Se a tensão transmitida na linha L2 em uma carga de 100 ohms é  $V\_REFERENCE\_O(f)$  e a tensão FEXT recebida na linha L1, enquanto ambas as terminações têm uma carga de 100 ohms é  $V\_FEXT\_R(f)$  – vide Fig. 4.4, então o ganho de inserção de FEXT em *downstream* resultante da linha L2 na linha L1 em escala linear é dado pela equação abaixo:

$$FEXT\_IG\_DS_{L1L2}(f) = V\_FEXT\_R(f) / V\_REFERENCE\_O(f) \quad (4.4)$$

O coeficiente de acoplamento de FEXT *downstream* da linha L2 na linha L1, na frequência  $f$  é definido como o valor de  $FEXT\_IG\_DS_{L1L2}(f)$  dividido pelo ganho de inserção do canal direto de L1, ou seja a função característica de canal  $\mathbf{H}$  da linha L1:

$$Xlinds_{L1L2}(f) = FEXT\_IG\_DS_{L1L2}(f) / H_{L1}(f) \quad (4.5)$$

O coeficiente de acoplamento de FEXT *downstream* pode também ser representado em termos do canal direto  $V\_DIRECT\_R(f)$  recebido na linha L1, e da tensão de FEXT recebida pela linha L1:

$$Xlinds_{L1L2}(f) = V\_FEXT\_R(f) / V\_DIRECT\_R(f) \quad (4.6)$$

A Fig. 4.9 ilustra que  $V\_DIRECT\_R(f)$  é a tensão recebida em uma carga de 100 ohm em uma linha L1 quando um transmissor com uma tensão de transmissão igual a  $V\_REFERENCE\_O(f)$  é mantido em um estado de transmissão e conectado à mesma linha. Pela Fig. 4.8,  $V\_FEXT\_R(f)$  é a tensão recebida na linha L1 quando esta linha é terminada com uma carga de 100 ohm em ambos os lados e o transmissor com uma mesma tensão de transmissão é conectado à linha L2.

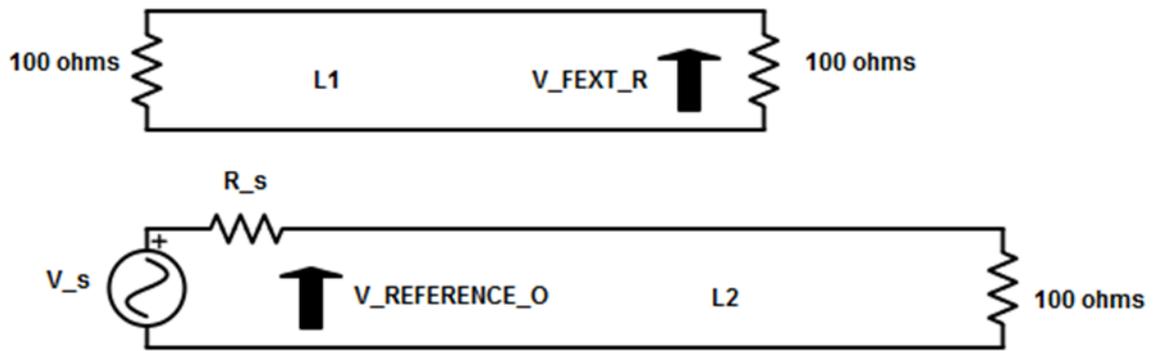


Figura 4.8 Definição de  $Xlin$  em *downstream* – tensão FEXT do canal.

Quando requisitado pelo ME os coeficientes de acoplamento de FEXT  $Xlinpsds_{i,k}(n,\Delta f)$  de uma faixa de frequências  $n.\Delta f$  são reportados.  $Xlinpsds_{i,k}(n,\Delta f)$  é representado em formato linear por um fator de escala e um número complexo normalizado  $a(n) + j.b(n)$ , a partir de valores de  $Xlinds$ , onde  $n$  é o índice da subportadora. O fator de escala  $Xlinscds$  deve ser codificado como um número inteiro sem sinal de 16 bits. Os números  $a(n)$  e  $b(n)$  devem ser codificados como inteiros com sinal de 16 bits e em complemento de 2.

O valor de  $Xlinpsds(n,\Delta f)$  é definido como:

$$Xlinpsds(n,\Delta f) = (Xlinscds / 2^{15}) \cdot (a(n) + j \cdot b(n)) / 2^{15} \quad (4.7)$$

O fator de escala  $Xlinscds$  é escolhido de forma que o valor máximo de  $\max(|a(n)|, |b(n)|)$  em todos os valores de  $n$  seja igual a  $2^{15}-1$ .

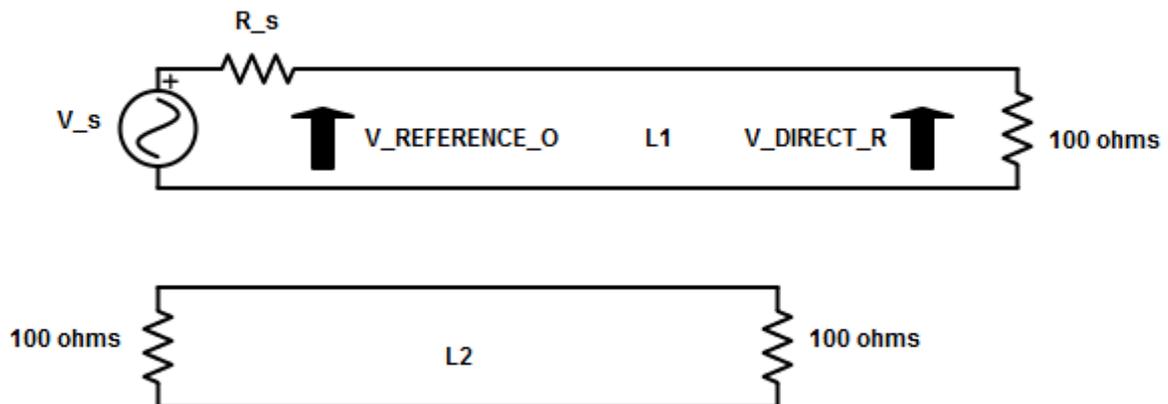


Figura 4.9 Definição de  $Xlin$  em *downstream* – tensão direta do canal.

O formato dos dados possibilita uma granularidade de  $2^{-15}$  e uma excursão de aproximadamente 6 dB a -90 dB.

A magnitude de  $Xlinpsds_{i,k}(n,\Delta f)$  em dB é  $Xlogpsds_{i,k}(n,\Delta f)$ :

$$Xlogpsds_{i,k}(n.\Delta f) = 20 \log_{10}(|Xlinpsds_{i,k}(n.\Delta f)|) \quad (4.8)$$

Os requisitos de acurácia de  $Xlinpsds$  são formulados em termos da magnitude somente, ou seja de  $Xlogpsds$ .

### 4.3.2 Coeficientes de acoplamento FEXT em *upstream*

O ganho de inserção de FEXT de uma linha L2 em uma linha L1 em *upstream*, na frequência  $f$ , chamado de  $FEXT\_IG\_US_{L1L2}(f)$ , é definido como o valor da tensão FEXT recebida em uma carga de 100 ohm em uma linha L1 dividido pela tensão de transmissão para uma carga de 100 ohm da linha L2. Se a tensão de transmissão na linha L2 em uma carga de 100 ohm é chamada de  $V\_REFERENCE\_R(f)$  e a tensão FEXT recebida na linha L1 é  $V\_FEXT\_O(f)$ , enquanto ambas as terminações tem cargas de 100 ohms como mostra a Fig. 4.10, então o ganho de inserção de FEXT em *upstream* da linha L2 na linha L1 em escala linear é dado pela seguinte equação:

$$FEXT\_IG\_US_{L1L2}(f) = V\_FEXT\_O(f) / V\_REFERENCE\_R(f) \quad (4.9)$$

O coeficiente de acoplamento FEXT em *upstream* da linha L2 na linha L1, em uma frequência  $f$  é definido como a taxa de inserção de ganho FEXT da linha L2 na linha L1 pelo ganho de canal direto da linha L2, ou seja a função característica  $\mathbf{H}$  para a linha L2):

$$Xlinus_{L1L2}(f) = FEXT\_IG\_US_{L1L2}(f) / h_{L2}(f) \quad (4.10)$$

O coeficiente de acoplamento FEXT de *upstream* pode também ser representado em termos do canal direto,  $V\_DIRECT\_O(f)$  recebido da linha L2 e da tensão FEXT recebida na linha L1:

$$Xlinus_{L1L2}(f) = V\_FEXT\_O(f) / V\_DIRECT\_O(f) \quad (4.11)$$

Conforme ilustrado na Fig. 4.11,  $V\_DIRECT\_O(f)$  é a tensão recebida em uma carga de 100 ohms na linha L2, quando o transmissor com uma tensão de transmissão igual à  $V\_REFERENCE\_R(f)$  é mantida em seu estado de transmissão e conectado à mesma linha. Como mostrado na Fig. 4.6,  $V\_FEXT\_O(f)$  é a tensão recebida na linha L1 quando esta linha é terminada com carga de 100 ohms em ambos os lados e o transmissor com a mesma tensão de transmissão é conectado à linha L2.

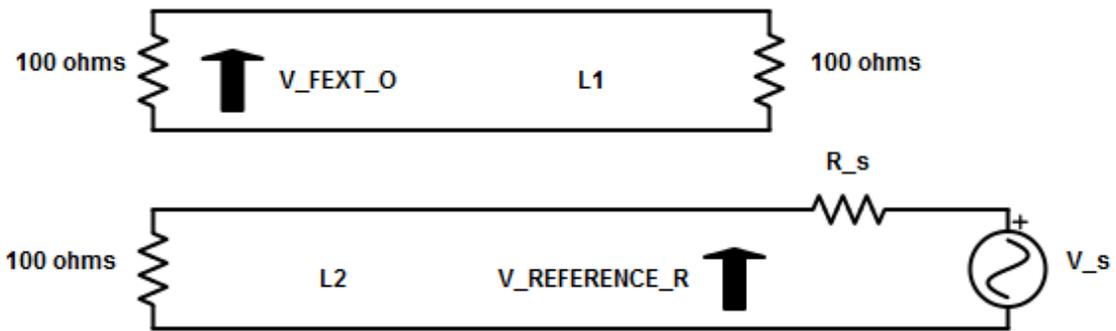


Figura 4.10 Definição de  $Xlin$  em *upstream* – tensão de FEXT do canal.

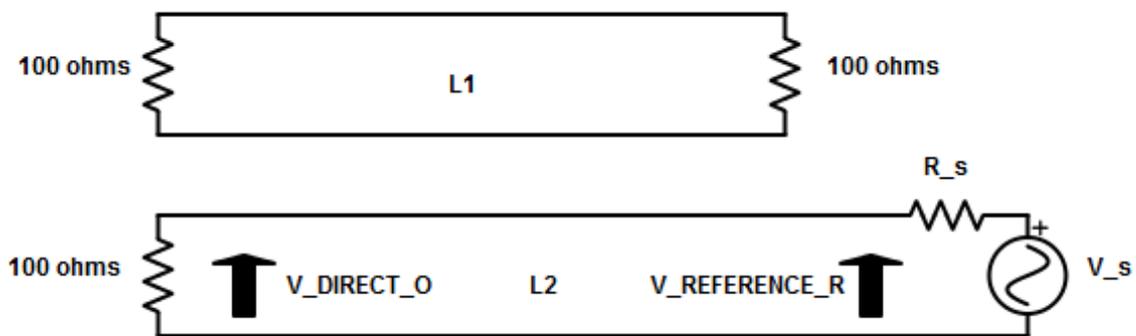


Figura 4.11 Definição de  $Xlin$  em *upstream* – tensão direta do canal

Quando requisitado pelo ME os coeficientes de acoplamento de FEXT  $Xlinpsus_{i,k}(n,\Delta f)$  de uma faixa de frequências  $n,\Delta f$  são reportados.  $Xlinpsus_{i,k}(n,\Delta f)$  é representado em formato linear por um fator de escala e um número complexo normalizado  $a(n) + j.b(n)$ , a partir de valores de  $Xlinus$ , onde  $n$  é o índice da subportadora. O fator de escala  $Xlinscus$  deve ser codificado como um número inteiro sem sinal de 16 bits. Os números  $a(n)$  e  $b(n)$  devem ser codificados como inteiros com sinal de 16 bits e em complemento de 2.

O valor de  $Xlinpsus(n,\Delta f)$  é definido como:

$$Xlinpsus(n,\Delta f) = (Xlinscus / 2^{15}) \times (a(n) + j.b(n)) / 2^{15} \quad (4.12)$$

O formato dos dados possibilita uma granularidade de  $2^{-15}$  e uma excursão de aproximadamente 6 dB a -90 dB.

A magnitude de  $Xlinpsus_{i,k}(n,\Delta f)$  em dB é  $Xlogpsus_{i,k}(n,\Delta f)$

$$Xlogpsus_{i,k}(n,\Delta f) = 20 \log_{10}(|Xlinpsus_{i,k}(n,\Delta f)|) \quad (4.13)$$

Os requisitos de acurácia de  $Xlinpsus$  são formulados em termos da magnitude somente, ou seja de  $Xlogpsus$ .

## 5 MÉTODO DE INVERSÃO MATRICIAL PROPOSTO GSGR APLICADO EM PRÉ-CODIFICAÇÃO DE SISTEMAS XDSL *G.FAST*

O método de inversão matricial proposto neste trabalho é um método direto baseado na eliminação de Gauss e na decomposição QR.

Estimadores baseados em decomposição QR podem ser usados para resolver sistemas lineares quadrados como os considerados no estudo de caso deste trabalho. Mas considerando-se o número de operações em ponto flutuante, o método de eliminação de Gauss (LU) é o método mais econômico. Entretanto, se considerada a quantidade de memória necessária e de dados com informações de vetores (*overheads*), QR é comparável em eficiência [1].

Outro aspecto a ser considerado é que o método de Gauss tem um fator de crescimento significativo, especialmente em matrizes com grandes dimensões [1], isto é, a cada ciclo, as operações com divisões por componentes de pequeno valor vão resultando em componentes maiores, aumentando a chance de erros de *overflow*.

Para tentar minimizar os erros de processamento na inversão de matrizes *G.fast* com dimensão igual a 64 no mínimo, o método GSGR foi desenvolvido a partir dos métodos de GR (*Givens Rotations*) e SGR (*Squared Givens Rotations*), incluindo-se um processamento em bloco com Complemento de *Schur*, e modificando-o até quase aproximá-lo pelo método de Gauss, como descrito nas próximas seções.

### 5.1 DECOMPOSIÇÃO QR COM MÉTODO DE *GIVENS ROTATIONS* (GR)

O método clássico de *Givens Rotations* (GR) [19] consiste em uma transformação ortogonal de rotação unitária em um plano, podendo ser representado por uma matriz da forma:

$$\mathbf{G}(i, k, \theta) = \begin{bmatrix} 1 & \dots & 0 & \dots & 0 & \dots & 0 \\ \vdots & \ddots & \vdots & 0 & \vdots & 0 & \vdots \\ 0 & \dots & c & \dots & s & \dots & 0 \\ \vdots & 0 & \vdots & \ddots & \vdots & 0 & \vdots \\ 0 & \dots & -s & \dots & c & \dots & 0 \\ \vdots & 0 & \vdots & 0 & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & \dots & 0 & \dots & 1 \end{bmatrix} \quad (5.1)$$

onde:

$$c = \cos \theta ;$$

$$s = \sin \theta ;$$

$\theta$  é o ângulo de rotação em radianos;

$i$  e  $k$  são as linhas não nulas e representam um plano de coordenadas.

A multiplicação de  $\mathbf{G}^T$  por um vetor  $\mathbf{x}$  ou matriz  $\mathbf{A}$  resulta em uma rotação de  $\theta$  radianos no plano  $(i, k)$ , no sentido anti-horário. Como apenas as linhas  $i$  e  $k$  são alteradas pode-se escrever:

$$\mathbf{y} = \mathbf{G}(i, k, \theta)^T \mathbf{x} \quad (5.2)$$

$$\begin{bmatrix} c & -s \\ s & c \end{bmatrix} \begin{bmatrix} x_i \\ x_k \end{bmatrix} = \begin{bmatrix} q \\ 0 \end{bmatrix} = \begin{bmatrix} y_i \\ y_k \end{bmatrix} \quad (5.3)$$

$$y_i = c x_i - s x_k \quad (5.4)$$

$$y_k = s x_i + c x_k \quad (5.5)$$

$$y_j = x_j \quad \text{para } j \neq i, k \quad (5.6)$$

Para obter  $y_k = 0$  seleciona-se:

$$c = \frac{x_i}{q}$$

$$s = \frac{-x_k}{q}$$

$$q = \sqrt{x_i^2 + x_k^2} \quad (5.7)$$

Para descrever o caso de rotação das linhas  $\mathbf{r}$  e  $\mathbf{a}$  de uma matriz considere-se a matriz  $\mathbf{A}_{m \times n}$  parcialmente reduzida, com uma linha  $\mathbf{a} = (0, \dots, 0, a_k, \dots, a_n)$  cujo elemento  $a_k \neq 0$  deva ser anulado na próxima rotação – de forma a obter a triangularização, e outra linha  $\mathbf{r} = (0, \dots, 0, r_k, \dots, r_n)$  já com o número desejado de zeros. Tem-se as seguintes equações para as novas linhas  $\bar{\mathbf{r}}$  e  $\bar{\mathbf{a}}$  com um zero na  $k$ -ésima posição de  $\bar{\mathbf{a}}$ :

$$\mathbf{A}_{m \times n} = \begin{matrix} \mathbf{r}: \\ \mathbf{a}: \end{matrix} \begin{bmatrix} X & X & \dots & X & \dots & X \\ 0 & X & \dots & \dots & \dots & X \\ 0 & 0 & \dots & r_k & \dots & r_n \\ 0 & 0 & \dots & a_k & \dots & a_n \\ 0 & 0 & \dots & X & \dots & X \\ 0 & 0 & \dots & X & \dots & X \end{bmatrix} \quad (5.8)$$

$$\begin{bmatrix} c & s \\ -s & c \end{bmatrix} \begin{bmatrix} \mathbf{r} \\ \mathbf{a} \end{bmatrix} = \begin{bmatrix} \bar{\mathbf{r}} \\ \bar{\mathbf{a}} \end{bmatrix}, \quad c = \frac{r_k}{q}, \quad s = \frac{a_k}{q} \quad (5.9)$$

$$q = \sqrt{r_k^2 + a_k^2} \quad (5.10)$$

$$\bar{\mathbf{r}} = \frac{r_k \mathbf{r} + a_k \mathbf{a}}{q} \quad (5.11)$$

$$\bar{\mathbf{a}} = \frac{-a_k \mathbf{r} + r_k \mathbf{a}}{q} \quad (5.12)$$

Para completar o processo de triangularização da matriz  $\mathbf{A}$  são necessárias tantas rotações – multiplicações  $\mathbf{G}^T \mathbf{A}$  – quantas forem os elementos da matriz a serem anulados. Então para  $t$  rotações:

$$\mathbf{G}_t^T \dots \mathbf{G}_2^T \mathbf{G}_1^T \mathbf{A} = \mathbf{R} \quad (5.13)$$

$$(\mathbf{G}_1 \mathbf{G}_2 \dots \mathbf{G}_t)^T \mathbf{A} = \mathbf{R} \quad (5.14)$$

$$\mathbf{Q}^T \mathbf{A} = \mathbf{R}, \quad \text{com } \mathbf{Q}^T = \mathbf{Q}^{-1} \quad (5.15)$$

Como se tratam de transformações ortogonais a matriz  $\mathbf{Q}^T$  é unitária, logo a expressão acima equivale a  $\mathbf{A} = \mathbf{Q} \mathbf{R}$ .

## 5.2 DECOMPOSIÇÃO QR COM SQUARED GIVENS ROTATIONS (SGR)

Em aperfeiçoamento do método de GR, o método de decomposição SGR [4] permite obter a triangularização de uma matriz  $\mathbf{A}$  com um menor número de operações de multiplicações e sem operações de raiz quadrada.

No método SGR representa-se a matriz  $\mathbf{A}_{m \times n}$ , de posto<sup>1</sup>  $n$ , como:

---

<sup>1</sup> Ver Apêndice para definições básicas de Álgebra linear

$$\mathbf{A} = \mathbf{Q}_A \mathbf{D}_U^{-1} \mathbf{U} \quad (5.16)$$

onde:

$$- \mathbf{Q}_A = \mathbf{Q} \mathbf{D}_R \quad (5.17)$$

é uma matriz cujas colunas são uma base ortogonal de  $\mathbf{A}$ ;

$$- \mathbf{U} = \mathbf{D}_R \mathbf{R} \quad (5.18)$$

é a mesma matriz triangular superior obtida pelo método Gaussiano em  $\mathbf{A} = \mathbf{L} \mathbf{U}$ ;

$$- \mathbf{D}_R = \text{diag}(\mathbf{R}) \quad (5.19)$$

$$- \mathbf{D}_U = \text{diag}(\mathbf{U}) = \mathbf{D}_R^2 \quad (5.20)$$

As decomposições QR em (2.18) e (5.16) são equivalentes, conforme demonstrado a seguir.

Substituindo-se  $\mathbf{R} = \mathbf{D}_R^{-1} \mathbf{U}$  a partir de (5.18), em  $\mathbf{A} = \mathbf{Q} \mathbf{R}$ , obtém-se:

$$\mathbf{A} = \mathbf{Q} \mathbf{D}_R^{-1} \mathbf{U} \quad (5.21)$$

$$\mathbf{A} = \left( \mathbf{D}_R \mathbf{Q}^{-1} \right)^{-1} \mathbf{U} \quad (5.22)$$

Substituindo  $\mathbf{Q}^{-1} = \mathbf{D}_R \mathbf{Q}_A^{-1}$  a partir de (5.17):

$$\mathbf{A} = \left( \mathbf{D}_R \mathbf{D}_R \mathbf{Q}_A^{-1} \right)^{-1} \mathbf{U} \quad (5.23)$$

Usando (5.20) obtém-se:

$$\mathbf{A} = \left( \mathbf{D}_U \mathbf{Q}_A^{-1} \right)^{-1} \mathbf{U} \quad (5.24)$$

$$\mathbf{A} = \mathbf{Q}_A \mathbf{D}_U^{-1} \mathbf{U} \quad (5.25)$$

O SGR pode ser entendido como uma translação do espaço vetorial definido pelos vetores  $\mathbf{r}$  e  $\mathbf{a}$  em (5.11) e (5.12) para um espaço vetorial definido por outros vetores  $\mathbf{u}$  e  $\mathbf{v}$ , conforme descrito a seguir.

Para apresentar as equações de aplicação do método SGR considere-se as equações de GR de (5.10) a 5.12). A expressão para o  $k$ -ésimo elemento da linha  $\mathbf{r}$  de uma matriz  $\mathbf{A}$  é:

$$\bar{r}_k = \frac{r_k r_k + a_k a_k}{q} = q \quad (5.26)$$

Assim pode-se reescrever (5.11) como:

$$\bar{r}_k \bar{\mathbf{r}} = r_k \mathbf{r} + a_k \mathbf{a} \quad (5.27)$$

Supondo-se  $r_k \neq 0$  define-se a linha  $\mathbf{u}$  e sua atualização  $\bar{\mathbf{u}}$  como:

$$\mathbf{u} = r_k \mathbf{r} , \quad \bar{\mathbf{u}} = \bar{r}_k \bar{\mathbf{r}} \quad (5.28)$$

Reescreve-se (5.11) novamente como:

$$\bar{\mathbf{u}} = \mathbf{u} + a_k \mathbf{a} \quad (5.29)$$

A linha  $\mathbf{a}$  é reescrita em função de um número escalar  $w > 0$  e de uma linha  $\mathbf{v}$ :

$$\mathbf{a} = w^{1/2} \mathbf{v} \quad (5.30)$$

Então reescrevendo-se (3.12) obtém-se:

$$\bar{\mathbf{a}} = (w^{1/2} r_k / \bar{r}_k) (\mathbf{v} - (v_k / r_k) \mathbf{r}) \quad (5.31)$$

E obtém-se que:

$$\bar{w} = w r_k^2 / \bar{r}_k^2 \quad (5.32)$$

$$\bar{\mathbf{v}} = \mathbf{v} - (v_k / r_k) \mathbf{r} \quad (5.33)$$

Finalmente, reescrevendo as equações apenas em função de  $\mathbf{u}$ ,  $\mathbf{v}$  e  $w$ , obtém-se:

$$\bar{\mathbf{u}} = \mathbf{u} + w v_k^* \mathbf{v} \quad (5.34)$$

$$\bar{\mathbf{v}} = \mathbf{v} - \left(\frac{v_k}{u_k}\right) \mathbf{u} \quad (5.35)$$

$$\bar{w} = w \frac{u_k}{\bar{u}_k} \quad (5.36)$$

Assim em cada operação de rotação, as equações do SGR efetuam uma translação das linhas  $\mathbf{r}$  e  $\mathbf{a}$  da matriz  $\mathbf{A}$  para os espaços  $\mathbf{U}$  e  $\mathbf{V}$ , resultando em um novo valor de  $w$ , indicado por  $\bar{w}$ , e pelas novas linhas  $\bar{\mathbf{u}}$  e  $\bar{\mathbf{v}}$ .

### 5.2.1 Ocorrência de elementos nulos durante o processo

As equações (5.34) a (5.36) foram desenvolvidas com a premissa de que o elemento em  $u_k$  é diferente de zero. Mas para definir completamente o processo de triangularização é preciso solucionar o caso de a matriz a ser processada possuir zeros em  $u_k$ , ou ocorrerem zeros nessas posições durante rotações anteriores, devido a existência de pares de elementos lateralmente iguais, conforme ilustrado a seguir:

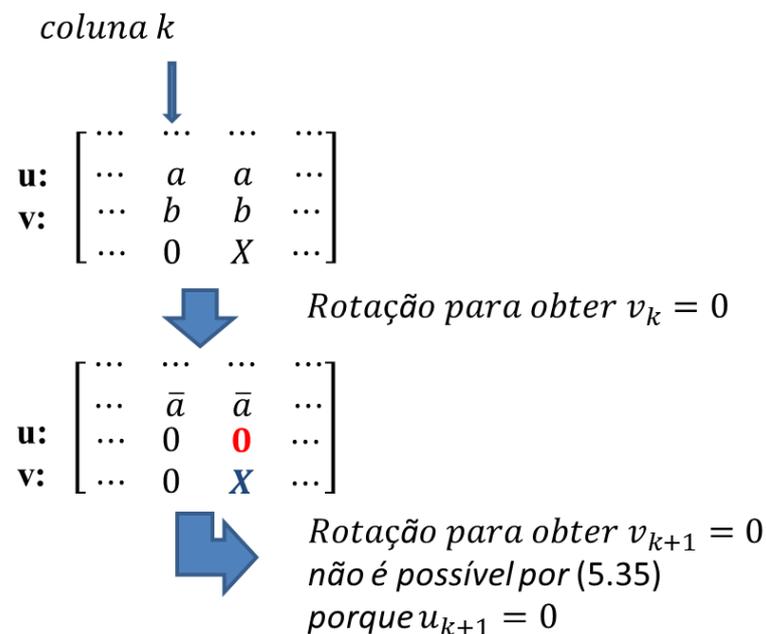


Figura 5.1 Ocorrência de zeros indesejados na matriz a ser triangularizada.

Para permitir que o processo de triangularização continue mesmo com  $u_k$  igual a zero foi proposta em [4] a seguinte solução pivotal:

se  $u_k = 0$  e  $v_k \neq 0$  então

$$\bar{\mathbf{u}} = w v_k^* \mathbf{v} \quad (5.37)$$

$$\mathbf{v} = \text{um valor arbitrário} \quad (5.38)$$

$$\bar{w} = \mathbf{0} \quad (5.39)$$

Em [20] a solução proposta incluiu uma nova solução para  $v_k \neq 0$  e também a situação em que  $v_k = 0$ :

se  $u_k = 0$  e  $v_k \neq 0$  então

$$\bar{\mathbf{u}} = w v_k^* \mathbf{v} \quad (5.40)$$

$$\mathbf{v} = -\mathbf{u} \quad (5.41)$$

$$\bar{w} = w \quad (5.42)$$

se  $u_k = 0$  e  $v_k = 0$  então

$$\bar{\mathbf{u}} = w \mathbf{v} \quad (5.43)$$

$$\mathbf{v} = -\mathbf{u} \quad (5.44)$$

$$\bar{w} = w \quad (5.45)$$

### 5.3 INVERSÃO DA MATRIZ TRIANGULAR OBTIDA COM SGR

Após sucessivas operações rotacionais com SGR sobre a matriz  $\mathbf{A}$  para anular cada elemento  $a_{k,j}$ , com  $k = \{n, n-1, \dots, j+1\}$  e  $j = \{1, \dots, n-1\}$ , obtém-se a matriz triangular superior  $\mathbf{U}$  de  $\mathbf{A} = \mathbf{Q}_A \mathbf{D}_U^{-1} \mathbf{U}$  (5.16). A Fig. 5.2 ilustra um exemplo para uma matriz quadrada de dimensão  $n$  igual a 3.

**SGR para obter U**  
 $j = \{1, \dots, n-1\} = \{1, 2\}$   
 $k = \{n, \dots, j+1\} = \{3, \dots, j+1\}$   
 $\Rightarrow a_{31}, a_{21}$  e  $a_{32}$  são anulados

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{231} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \xrightarrow{\text{SGR}} \mathbf{U} = \begin{bmatrix} u_{11} & u_{12} & u_{13} \\ 0 & u_{22} & u_{23} \\ 0 & 0 & u_{33} \end{bmatrix}$$

Figura 5.2 Ilustração da aplicação do método SGR em uma matriz  $\mathbf{A}_{3 \times 3}$  para obtenção de uma matriz triangular superior  $\mathbf{U}$ .

Obtém-se a matriz inversa  $\mathbf{A}^{-1}$  calculando-se a inversa de (5.16) por meio do produto do inverso de seus dois termos, tal que:

$$\mathbf{A}^{-1} = \mathbf{U}^{-1} \left( \mathbf{Q}_A \mathbf{D}_U^{-1} \right)^{-1} \quad (5.46)$$

A matriz inversa de  $\mathbf{U}$  pode ser calculada por um método clássico de retrosubstituição, utilizando-se as seguintes equações:

$$\mathbf{U}^{-1} = \mathbf{W}_{ij} = \begin{cases} - \left( \sum_{k=1}^{j-1} w_{ik} u_{kj} \right) / u_{jj}, & i < j \\ 1 / u_{jj}, & i = j \\ 0, & i > j \end{cases} \quad (5.47)$$

Para calcular o termo  $(\mathbf{Q}_A \mathbf{D}_U^{-1})^{-1}$  de (5.46), pode-se escrever a partir de (5.16) que:

$$\mathbf{U} = (\mathbf{Q}_A \mathbf{D}_U^{-1})^{-1} \mathbf{A} \quad (5.48)$$

$$\mathbf{M} = (\mathbf{Q}_A \mathbf{D}_U^{-1})^{-1} \quad (5.49)$$

$$\mathbf{U} = \mathbf{M} \mathbf{A} \quad (5.50)$$

Considera-se  $\mathbf{M}$  como uma matriz de transformação que multiplica  $\mathbf{A}$  para obter-se  $\mathbf{U}$ . Assim, enquanto se rotaciona  $\mathbf{A}$  sucessivamente para obter  $\mathbf{U}$ , pode-se rotacionar uma matriz identidade  $\mathbf{I}$  pelos mesmos valores tal que:

$$\mathbf{M} = \mathbf{M} \mathbf{I} \quad (5.51)$$

$$(\mathbf{Q}_A \mathbf{D}_U^{-1})^{-1} = (\mathbf{Q}_A \mathbf{D}_U^{-1})^{-1} \mathbf{I} \quad (5.52)$$

#### 5.4 CÁLCULO MATRICIAL EM BLOCO - COMPLEMENTO DE SCHUR

A complexidade da operação de inversão pode ainda ser reduzida usando-se o método de Complemento de *Schur*, de inversão matricial em bloco. Comumente usam-se algoritmos em bloco para obter um uso mais eficiente da memória dos processadores. Neste caso propõe-se o uso também para reduzir a dimensão da matriz a ser invertida, e assim reduzir os erros

cumulativos nos cálculos, especialmente as divisões que são realizadas por métodos aproximados em plataformas embarcadas, como o método de *Ralp Nelson*.

O método de complemento de Schur consiste em dividir uma matriz em submatrizes para um processo de inversão mais eficiente. Para uma matriz  $\mathbf{A}$ ,  $m \times n$ , dividida em 4 submatrizes  $\mathbf{P}_{m \times m}$ ,  $\mathbf{Q}_{m \times n}$ ,  $\mathbf{R}_{n \times m}$  e  $\mathbf{S}_{n \times n}$ , a inversa de  $\mathbf{A}$  pode ser calculada pelas seguintes equações [21]:

$$\mathbf{A} = \begin{bmatrix} \mathbf{P} & \mathbf{Q} \\ \mathbf{R} & \mathbf{S} \end{bmatrix} \quad (5.53)$$

$$\mathbf{X} = \mathbf{P}^{-1} + \mathbf{P}^{-1} \mathbf{Q} \mathbf{Y}^{-1} \mathbf{R} \mathbf{P}^{-1} \quad (5.54)$$

$$\mathbf{Y} = \mathbf{S} - \mathbf{R} \mathbf{P}^{-1} \mathbf{Q} \quad (5.55)$$

$$\mathbf{A}^{-1} = \begin{bmatrix} \mathbf{X} & -\mathbf{P}^{-1} \mathbf{Q} \mathbf{Y}^{-1} \\ -\mathbf{Y}^{-1} \mathbf{R} \mathbf{P}^{-1} & \mathbf{Y}^{-1} \end{bmatrix} \quad (5.56)$$

A última equação acima é a fórmula de inversão de *Banachiewicz*, e a submatriz  $\mathbf{Y}$  é chamada de complemento de *Schur* da matriz  $\mathbf{A}$  com respeito à  $\mathbf{P}$ , e  $\mathbf{S}$  precisa ser não singular [22].

Para um melhor balanceamento da carga de processamento entre os 4 núcleos do DSP utilizado, escolheu-se lidar com uma dimensão de bloco das submatrizes igual a  $n/2$ . Desta forma, com a utilização do Complemento de *Schur* foi necessário inverter apenas 2 submatrizes com metade da dimensão da matriz original.

## 5.5 MELHORAMENTOS PROPOSTOS

De forma a reduzir a complexidade de inversão e obter mais rapidez sem perdas expressivas em acurácia, o valor escalar  $w$  foi reduzido gradualmente em (5.34):  $\bar{\mathbf{u}} = \mathbf{u} + w v_k^* \mathbf{v}$ , até o limite de zero. Assim modificou-se o método SGR para executar somente rotações no vetor  $\mathbf{v}$ , enquanto foram mantidas todas as demais operações do SGR, e assim obteve-se um método que agrega o SGR e o método de eliminação de Gauss, um método Gauss SGR (GSGR):

$$\bar{\mathbf{u}} = \mathbf{u} \quad (5.41)$$

$$\bar{\mathbf{v}} = \mathbf{v} - \begin{pmatrix} v_k \\ u_k \end{pmatrix} \mathbf{u} \quad (5.42)$$

De acordo com [4], SGR reduz o número de multiplicações do método clássico GR para a metade, enquanto o GSGR proposto reduz novamente esse número à metade, resultando em apenas uma multiplicação a cada operação de rotação para anular o  $k$ -ésimo elemento da matriz.

Desta forma, juntou-se a estabilidade numérica característica do método QR [2] com o menor número de operações da eliminação de Gauss, sem a necessidade do uso de pivôs e sem restrições quando o valor de  $u_k = 0$ .

Para o caso de ocorrência de  $u_k = 0$ , propõe-se ainda uma solução diferente do SGR [4] e do MSGR [20], fazendo:

$$\bar{\mathbf{u}} = \mathbf{u} \quad (5.43)$$

$$\bar{\mathbf{v}} = -\mathbf{u}, \quad \text{se } u_k = 0 \quad (5.44)$$

No cap. 6 mostra-se a partir do resultado de simulações computacionais que as modificações propostas permitem calcular a inversa de matrizes com uma acurácia comparável aos demais métodos, obtendo-se como resultado desvios tão pequenos quanto  $10^{-11}$ , mas com a vantagem de precisar de um número menor de operações quando comparado a outros métodos diretos como o de Gauss (LU), o SGR e o MSGR.

## 6 RESULTADOS DA APLICAÇÃO DO MÉTODO DE INVERSÃO MATRICIAL GSGR A MATRIZES *G.FAST*

Para análise de desempenho e acurácia foram realizadas simulações com a plataforma de *software* MATLAB e também com uma plataforma DSP da Texas Instruments (TI) com um chip SoC TMS320C6670. Este chip é projetado para cálculos com alto desempenho e para sistemas embarcados de propósito geral, com seus 4 núcleos de 1,2-GHz C66x e suporte para operações aritméticas de ponto fixo e ponto flutuante [23].

Para as simulações realizadas no DSP, foram desenvolvidos algoritmos para os métodos LU, MSGR e GSGR, em linguagem C, na plataforma de *software* TI Code Composer Studio (CCS)

Os elementos das matrizes foram representados por números complexos em ponto flutuante com precisão simples, com resolução de 32 bits para a parte real e mais 32 bits para a parte imaginária, em conformidade com o formato IEEE 754 e com o recomendado pela Rec 993.5 [13] sobre *vectoring* – vide seção 4.3.

A comparação do desempenho de algoritmos é sempre uma tarefa difícil. Assim, especial atenção foi tomada para execução do algoritmo GSGR proposto e dos demais algoritmos para comparação objetiva em condições equivalentes. Para comparação de desempenho desses algoritmos foram calculados os números de operações teóricas de ponto flutuante (*flops*) para cada um e o tempo para a execução de uma operação de inversão completa de conjuntos de teste de matrizes.

### 6.1 RESULTADOS DOS TESTES DE DESEMPENHO

A Tab. 6.1 indica as equações polinomiais aproximadas para o número de operações necessárias para completar uma inversão, tendo sido gerada por meio da execução dos algoritmos para valores da dimensão  $n$  desde 4 até 64.

Tabela 6.1 Comparação de complexidade dos algoritmos testados com relação ao número de operações em função da dimensão  $n$  da matriz

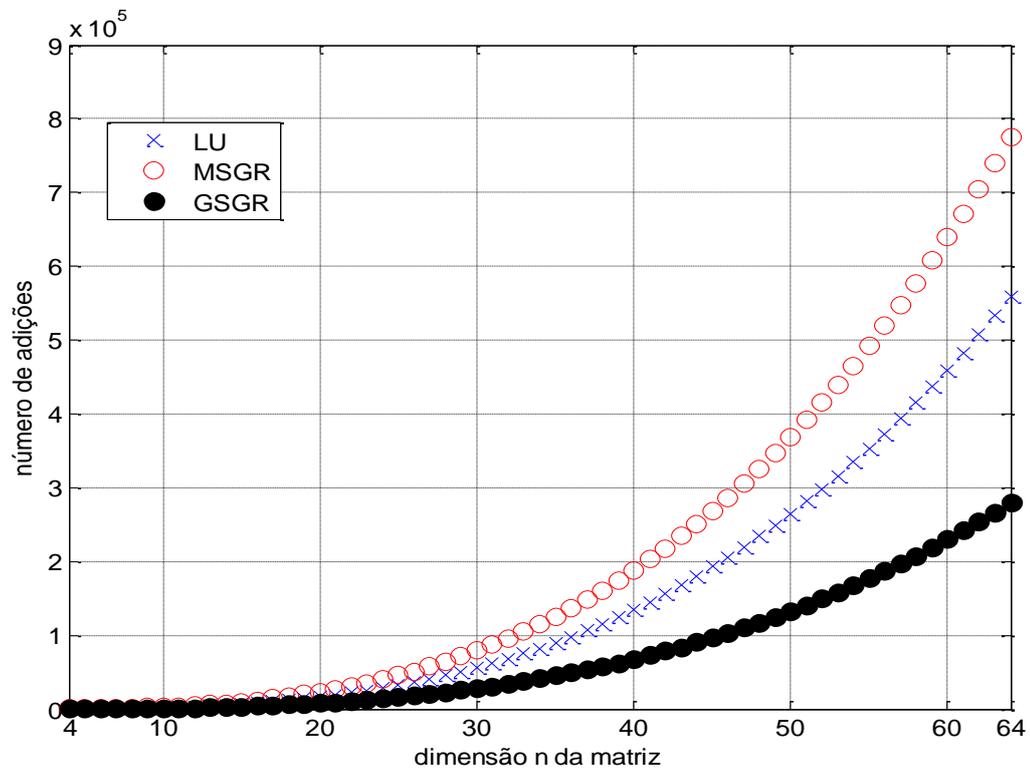
Algoritmo	Número de Adições	Número de Multiplicações	Número de Divisões
LU	$2,17n^3 - 2,5n^2 - 0,33n$	$2,5n^3 - 1,5n^2$	$0,83n^3 + 2,17n - 1$
MSGR	$3n^3 - 2,5n^2 - 0,5n$	$3,33n^3 - n^2 + 0,33n - 1$	$0,33n^3 + 1,67n - 1$
GSGR	$1,08n^3 - 0,75n^2 - 1,83n$	$1,17n^3 + 0,5n^2 - 1,67n$	$0,08n^3 - 0,25n^2 + 2,17n - 2$

A Tab. 6.2 apresenta o número de operações para o caso de uma matriz de dimensão  $n$  igual a 4.

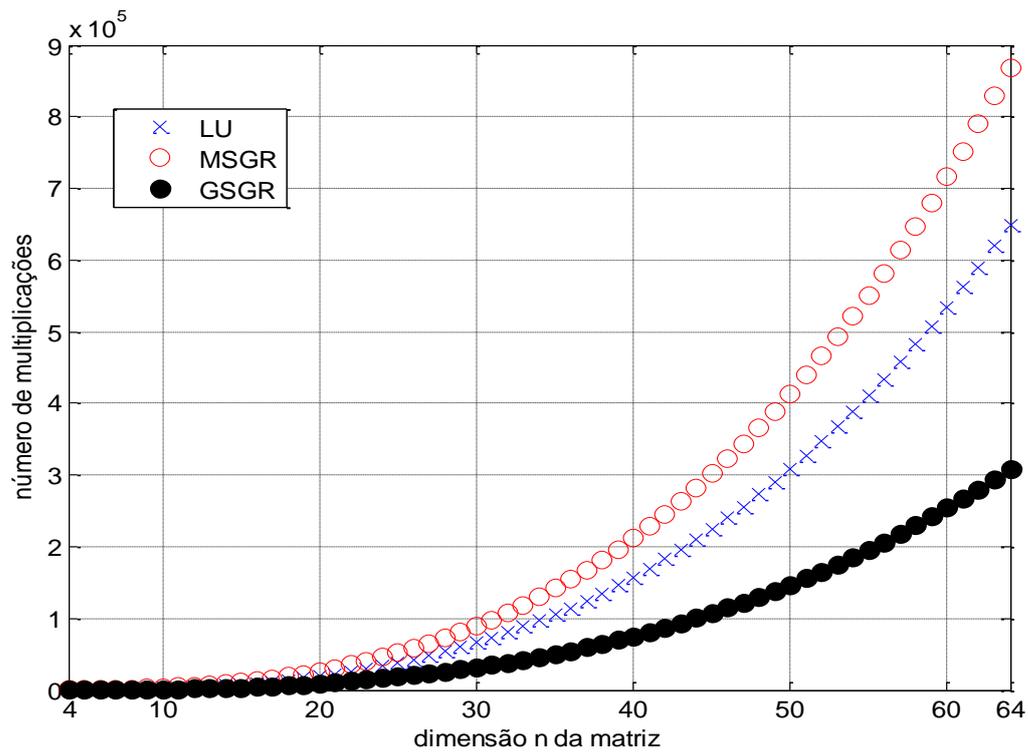
Tabela 6.2 Comparação de complexidade dos algoritmos testados com relação ao número de operações para uma matriz de dimensão  $n = 4$

Algoritmo	Número de Adições	Número de Multiplicações	Número de Divisões
LU	100	136	61
MSGR	150	195	27
GSGR	50	76	12

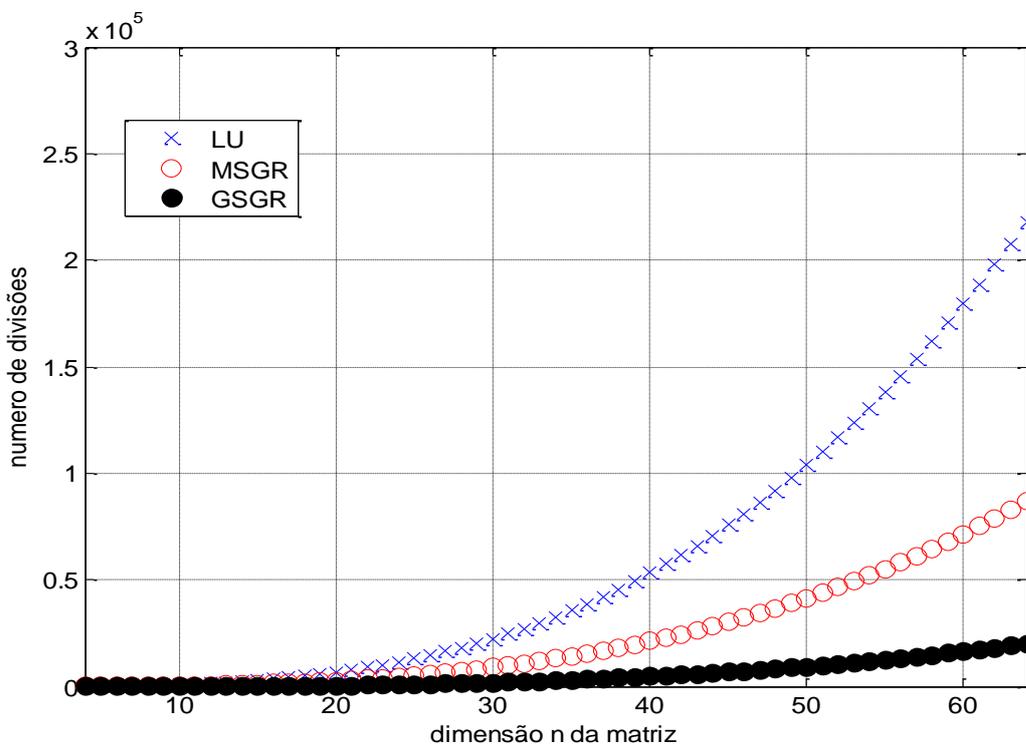
A Fig. 6.1 apresenta gráficos comparativos do número de operações de adições, multiplicações e divisões dos algoritmos LU, MSGR e GSGR.



(a)



(b)



(c)

Figura 6.1 Comparações do número de operações de adição (a), multiplicação (b) e divisão (c) necessário para inverter uma matriz de dimensão  $n$ , usando os algoritmos LU, MSGR e GSGR.

O número teórico de *flops* foi calculado como uma soma ponderada do número de adições, multiplicações e divisões, com diferentes pesos. Estes pesos foram escolhidos proporcionalmente ao número de ciclos necessários para o chip TMS320C6670 executar cada operação matemática [24], assumindo-se sempre o cálculo com números complexos, conforme a Tab. 6.3 abaixo:

Tabela 6.3 Pesos usados para o cálculo teórico do número de operações em ponto flutuante - flops para uma operação de inversão

Operação	Peso
Adição	1
Multiplicação	2
Divisão	10

A Fig. 6.2 mostra o gráfico comparativo do total de *flops* para matrizes de dimensão  $n$  desde 4 até 64.

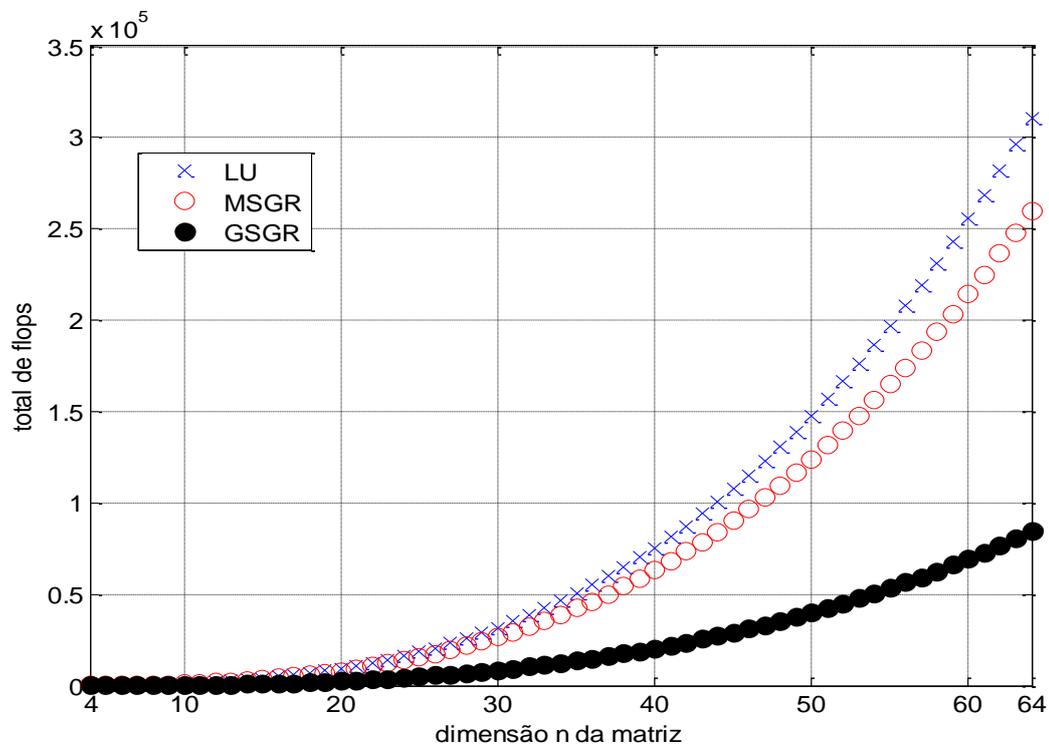


Figura 6.2 Comparação do número total de operações de ponto flutuante teóricas – *flops* necessário para inverter uma matriz de dimensão  $n$ , usando os algoritmos LU, MSGR e GSGR.

A Tab. 6.4 apresenta os números de operações por tipo de operação e o total ponderado de *flops* para execução de uma inversão completa com cada algoritmo, utilizando-se um conjunto de matrizes de teste de dimensão  $n = 64$ .

Tabela 6.4 Total ponderado de operações - *flops* dos algoritmos de inversão testados

Algoritmo	Número de Operações com matrizes de 64x64			Total ponderado de operações - <i>flops</i>
	Adições	Multiplicações	Divisões	
LU	557.760	649.216	218.591	310.930,9
MSGR	776.160	869.695	87.487	260.801,5
GSGR	280.582	307.994	20.740	84.920,6

Na Tab. 6.5 são comparados os totais de *flops* para a inversão de 4096 matrizes de dimensão 64x64 e os percentuais de redução do número de operações quando o LU é substituído pelo MSGR - redução para 84%, e pelo GSGR, redução para 27%. Quando comparado com o MSGR, o GSGR reduz a quantidade de operações para apenas 33%.

Tabela 6.5 Comparação da redução de *flops* obtida pelo método GSGR quando aplicado para inversão de 4096 matrizes de 64x64

Algoritmo	Total ponderado de operações - <i>flops</i>	de	Percentual do total de <i>flops</i> de LU obtido quando ele é substituído:	Percentual do total de <i>flops</i> de MSGR obtido quando ele é substituído:
			LU=>	MSGR =>
LU	310.930,9		100%	119%
MSGR	260.801,5		84%	100%
GSGR	84.920,6		27%	33%

O tempo de execução foi determinado como o tempo máximo em segundos para inverter uma única matriz dentre um conjunto de 20 matrizes complexas quadradas de dimensão 64 na plataforma DSP. Observa-se na Tab. 6.6 que o tempo de execução foi

reduzido para 55% e 38%, quando individualmente comparado com os algoritmos MSGR e o LU respectivamente.

A Fig. 6.3 mostra os tempos de execução para inversão matricial gastos por cada algoritmo sendo executado na plataforma DSP, quando da inversão do mesmo conjunto de matrizes de teste quadradas de dimensão 64. Foram observados períodos de tempo máximos de 2,30 ms para o GSGR, 3,97 ms para o MSGR e 6,76 ms para o LU, conforme já indicado na Tab. 6.6.

Tabela 6.6 Tempo máximo de execução dos algoritmos de inversão testados

Algoritmo	Tempo máximo de execução (ms)	Percentual do total de <i>flops</i> de LU obtido quando ele é substituído:	Percentual do total de <i>flops</i> de MSGR obtido quando ele é substituído:
		LU=>	MSGR =>
LU	5,76	100%	145%
MSGR	3,97	69%	100%
GSGR	2,20	38%	55%

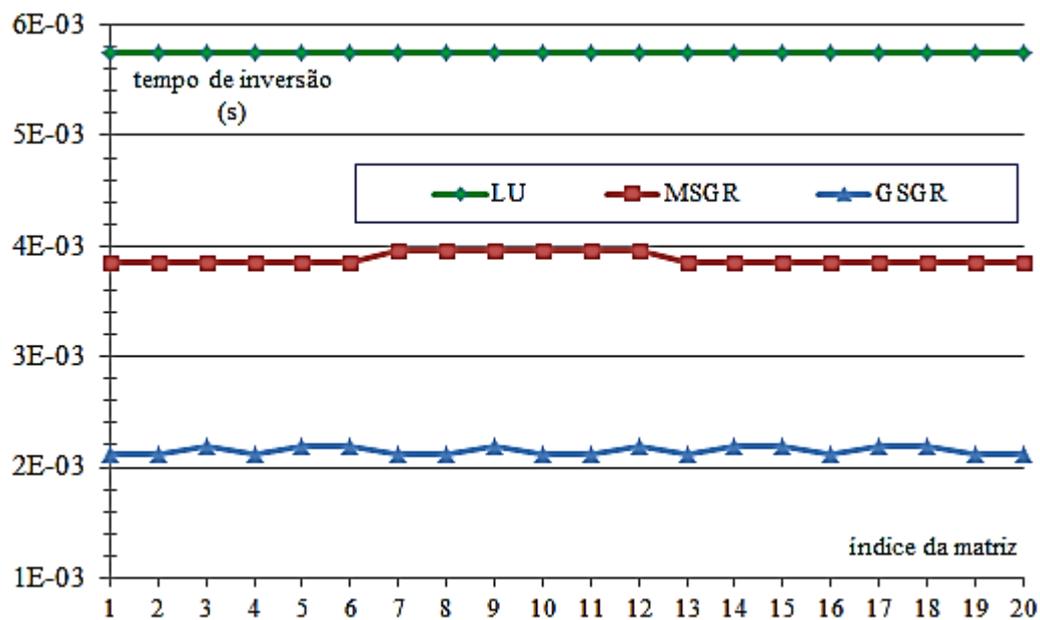


Figura 6.3 Períodos de tempo para inversão de 20 matrizes  $A_{64 \times 64}$  em uma plataforma DSP executando os algoritmos LU, MSGR e GSGR.

## 6.2 RESULTADOS DOS TESTES DE ACURÁCIA

Como mencionado no início do capítulo, a acurácia dos algoritmos foi avaliada usando ambas

as plataformas, Matlab e DSP. Para as simulações em ponto flutuante em precisão dupla no Matlab, os algoritmos LU, *Modified Gram Schmidt - Gram Schmidt* modificado (MGS) [1], MSGR e GSGR foram diretamente comparados com a função inversa do Matlab, que já é uma implementação acurada e otimizada.

Como adotado em trabalhos prévios [20], utilizou-se a métrica  $\mathbf{D}$  em (6.2) para avaliar a acurácia dos algoritmos testados, baseada na propriedade básica das matrizes inversas:

$$\mathbf{A} \mathbf{A}^{-1} = \mathbf{I} \quad (6.1)$$

$$\mathbf{D} = |\mathbf{I} - \mathbf{A} \tilde{\mathbf{A}}^{-1}| \quad (6.2)$$

Em (6.1) e (6.2)  $\mathbf{I}$  é a matriz identidade de mesma dimensão da matriz de teste  $\mathbf{A}$ , e  $\tilde{\mathbf{A}}^{-1}$  é a matriz inversa calculada por um determinado algoritmo. A eq. (6.2) representa assim o desvio  $\mathbf{D}$  da matriz inversa calculado por um algoritmo daquele que seria o resultado correto quando aplicado o cálculo da matriz identidade.

Foram efetuadas operações de inversão com cada um dos algoritmos testados, utilizando-se conjuntos tanto de matrizes geradas aleatoriamente quanto obtidas de medições reais. Em cada operação foi calculada uma matriz  $\mathbf{D}$  por (6.2), e a partir dela foram calculados o desvio médio, o desvio máximo e o desvio padrão dentre todos os seus  $n^2$  elementos.

Os conjuntos de teste utilizados foram de matrizes de números complexos. Nos conjuntos gerados as partes real e imaginária dos elementos foram representadas a partir de uma distribuição estatística Gaussiana com média e desvio padrão iguais a 0,5. Desta forma ficaram similares aos valores dos coeficientes de matrizes *G.fast* reais, considerando-se que no processo de estimação comum os valores de parâmetros das linhas são amostrados, quantizados e normalizados.

Após o processo de triangularização, foram usados os métodos clássicos de substituição e retrossubstituição para obter a inversa no algoritmo para o método LU, e somente retrossubstituição para os algoritmos correspondentes aos métodos MSGR e GSGR.

A Fig. 6.4 apresenta os resultados com relação à acurácia na execução da inversão de matrizes 100 x 100 realizadas no Matlab, geradas por distribuição Gaussiana. Os desvios máximos apresentados pelo algoritmo proposto GSGR foram maiores que os demais, mas

ainda dentro de uma faixa pequena entre  $1,48 \times 10^{-13}$  a  $5,1 \times 10^{-11}$ . Com relação a desvios médios, o GSGR apresentou valores menores que o MSGR em 82% das amostras, e menores que o MGS em 96% das amostras. Deve ser notado na Fig. 6.4 que as matrizes foram ordenadas de acordo com os desvios crescentes, apenas para melhor visualização dos resultados.

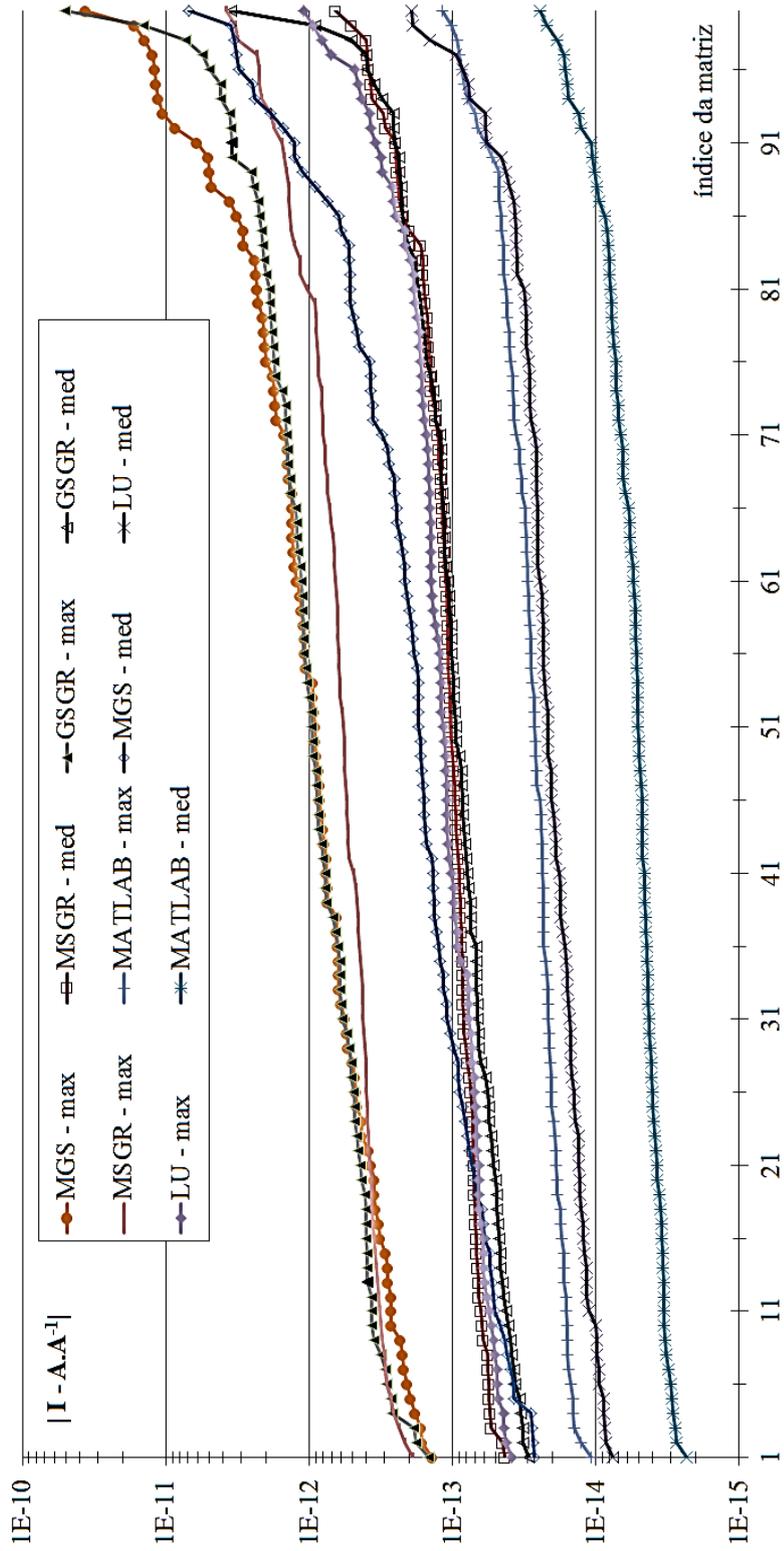


Figura 6.4 Desvios máximo e médio de  $|I - A.A^{-1}|$  dos algoritmos MGS, MSGR, GSGR, LU e da função inversa do Matlab, resultantes da inversão de 100 matrizes de  $100 \times 100$ .

A Fig. 6.5 apresenta os resultados de acurácia na inversão de 20 matrizes 64 x 64 realizadas no DSP, também geradas por distribuição Gaussiana. Deve ser notado que novamente os resultados foram ordenados. Foi observado que os desvios máximos estão dentro da faixa de  $4,93 \times 10^{-5}$  a  $1,46 \times 10^{-3}$  para o GSGR;  $5,71 \times 10^{-5}$  a  $7,7 \times 10^{-4}$  para o MSGR; e  $1,34 \times 10^{-5}$  a  $2,27 \times 10^{-4}$  para o LU. Assim, observou-se que o GSGR aumentou os desvios máximos mas os manteve abaixo de  $1,5 \times 10^{-3}$  e próximos à dimensão dos desvios do MSGR.

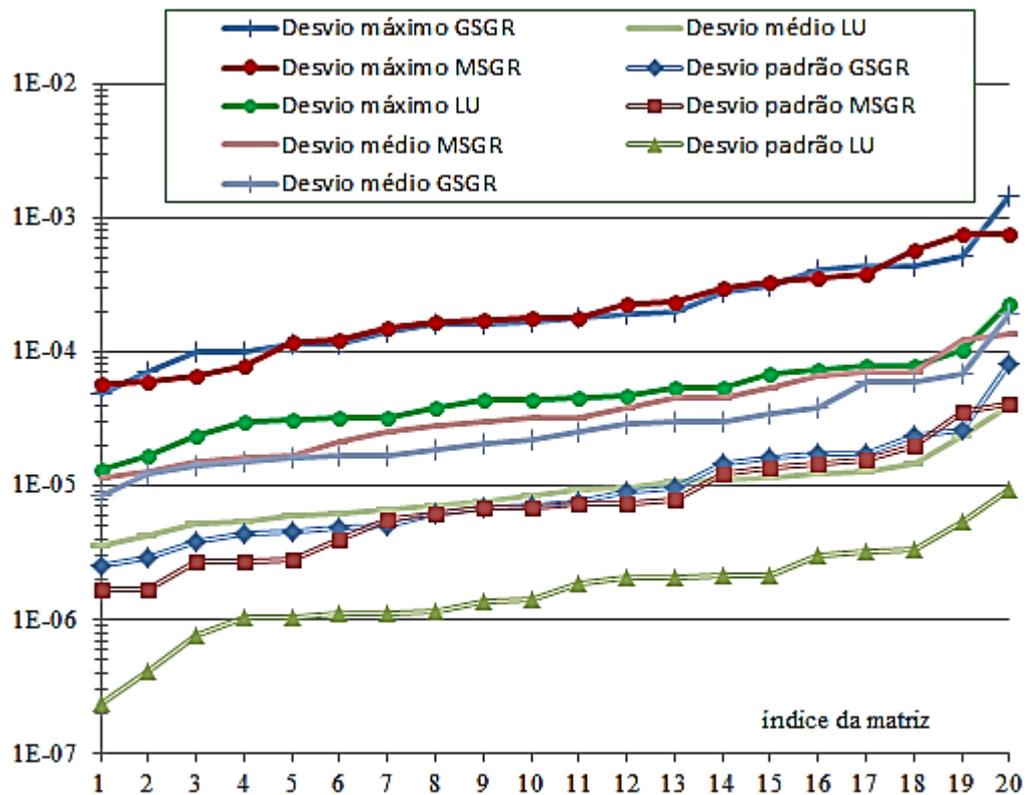


Figura 6.5 Desvios máximo, médio e padrão de  $|\mathbf{I} - \mathbf{A} \cdot \mathbf{A}^{-1}|$  na inversão de 20 matrizes  $\mathbf{A}_{64 \times 64}$  pelos algoritmos GSGR, LU e MSGR em um DSP TMS320C6670.

Para os testes com dados reais, o algoritmo proposto GSGR, os algoritmos MSGR, LU e a função inversa do Matlab foram aplicados a matrizes com dados de medições de canais. As medições são de um cabeamento de cobre de 100 metros de comprimento, com 24 linhas [25], e de um cabeamento de cobre de 30 metros e 10 linhas [26].

A Fig 6.6 apresenta a comparação de acurácia do GSGR em precisão simples e dupla com a função inversa do Matlab de precisão dupla, indicando um aumento no desvio máximo de  $1,1 \times 10^{-2}$  em apenas uma amostra, para o cabeamento de 100 metros e 24 linhas.

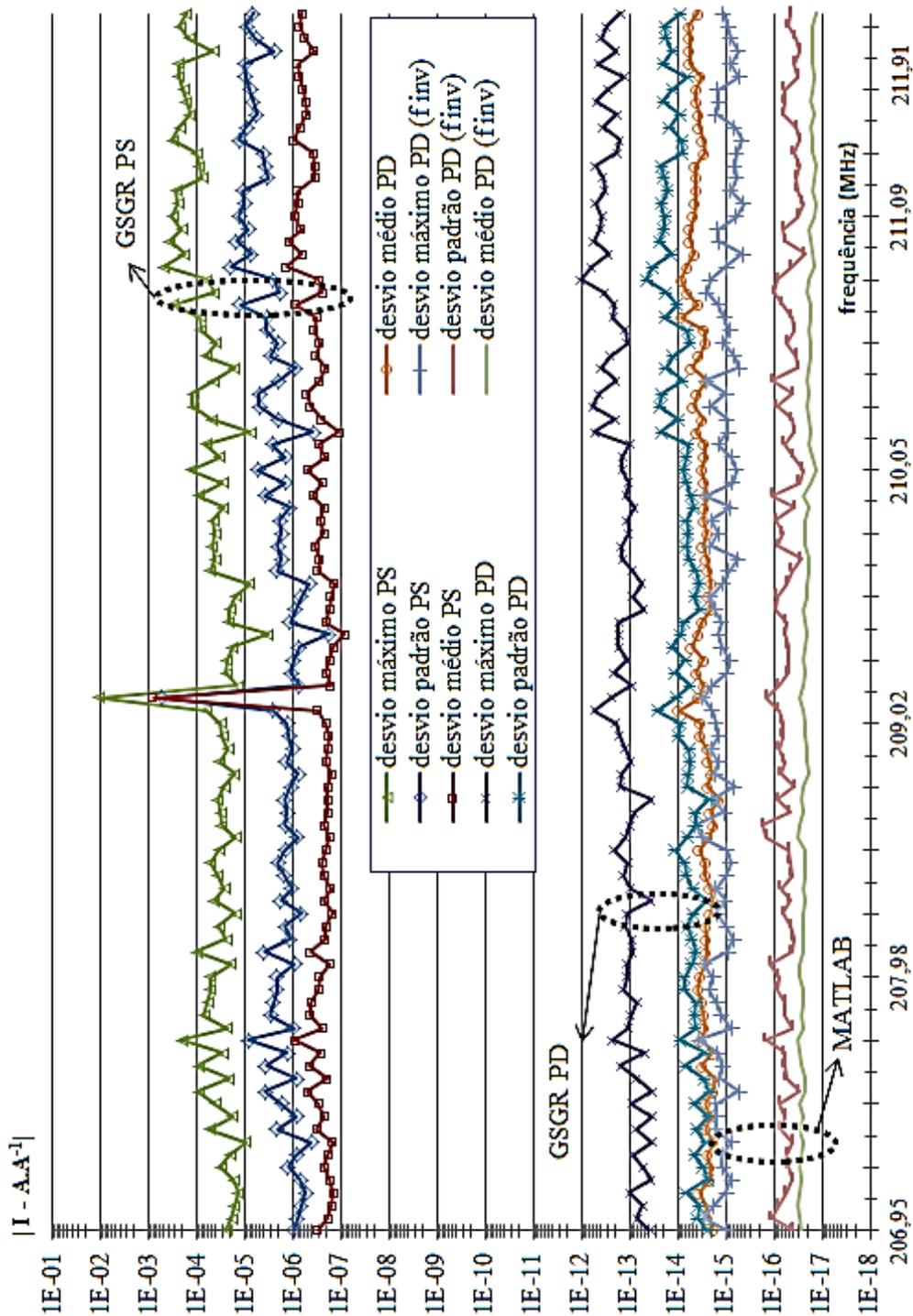


Figura 6.6 Desvios máximo, médio e padrão de  $|\mathbf{I} - \mathbf{A}.\mathbf{A}^{-1}|$  para o algoritmo GSGR e função inversa do Matlab usando precisão simples (PS) e precisão dupla (PD). As matrizes  $\mathbf{A}$  são matrizes de ganho de canal *G.fast* 24 x 24 obtidas de medições realizadas nas frequências de 206,95 MHz a 211,91 MHz.

A Fig 6.7 apresenta a comparação dos desvios máximos do GSGR, do MSGR e do LU em precisão dupla, desde 0 até 211,91 MHz para o cabeamento de 100 metros e 24 linhas. O

desempenho do GSGR foi similar ao do MSGR, com desvios máximos de  $2,026 \times 10^{-12}$  e  $1,657 \times 10^{-12}$ , respectivamente.

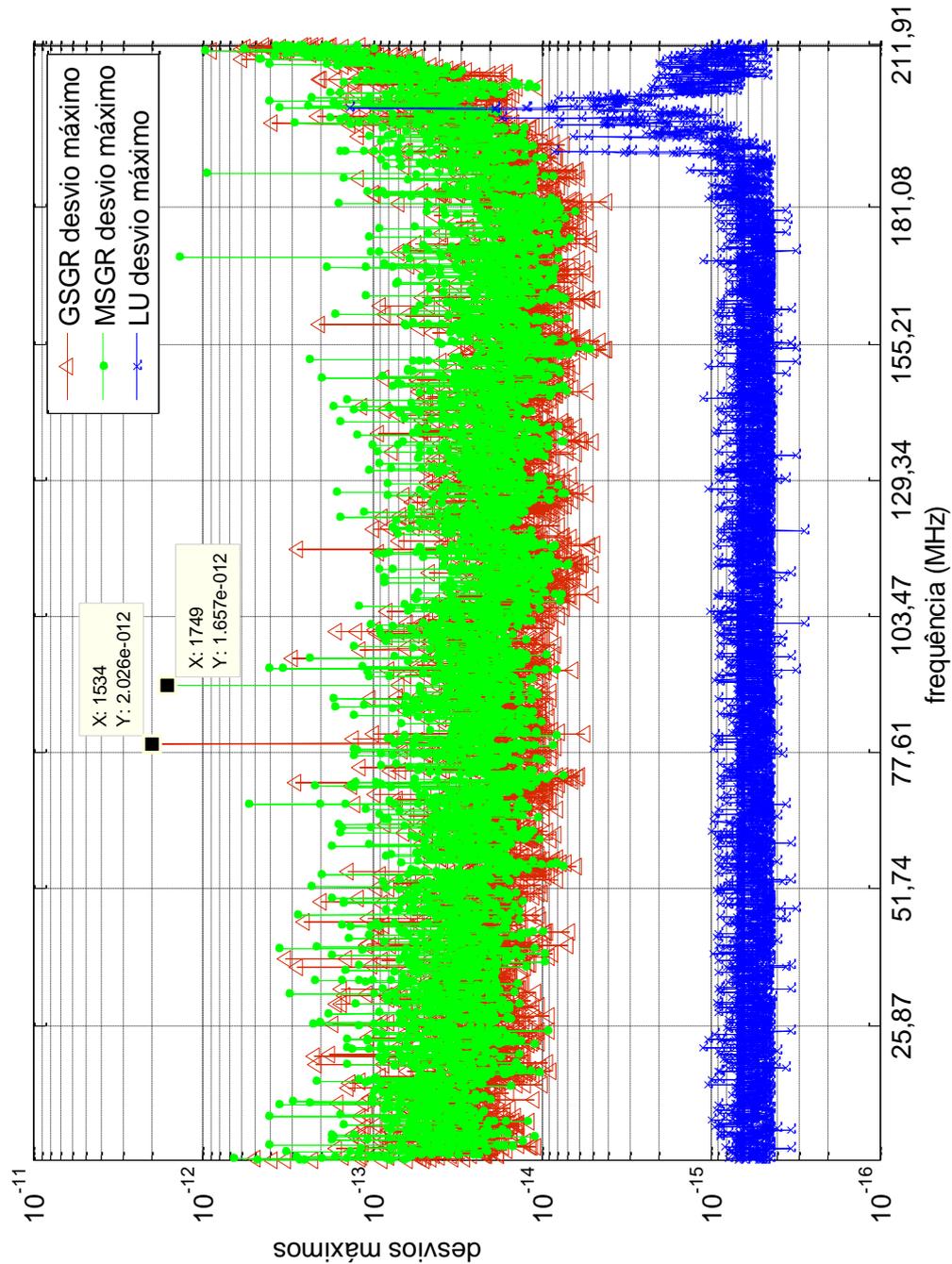


Figura 6.7 Comparação de desvios máximos de  $|\mathbf{I} - \mathbf{A} \cdot \mathbf{A}^{-1}|$  para os algoritmos GSGR, MSGR e LU em precisão dupla (PD). As matrizes  $\mathbf{A}$  usadas nesta simulação são matrizes de ganho de canal de um cabeamento com 24 linhas, obtidas de medições realizadas nas frequências desde 0 até 211,91 MHz.

A Fig 6.8 apresenta a comparação dos desvios máximos do GSGR, do MSGR e do LU em precisão dupla, desde 0 até 211,91 MHz para o cabeamento de 30 metros e 10 linhas. O desempenho do GSGR foi melhor que o do MSGR, com desvios máximos de  $1,126 \times 10^{-13}$  e  $2,026 \times 10^{-12}$ , respectivamente.

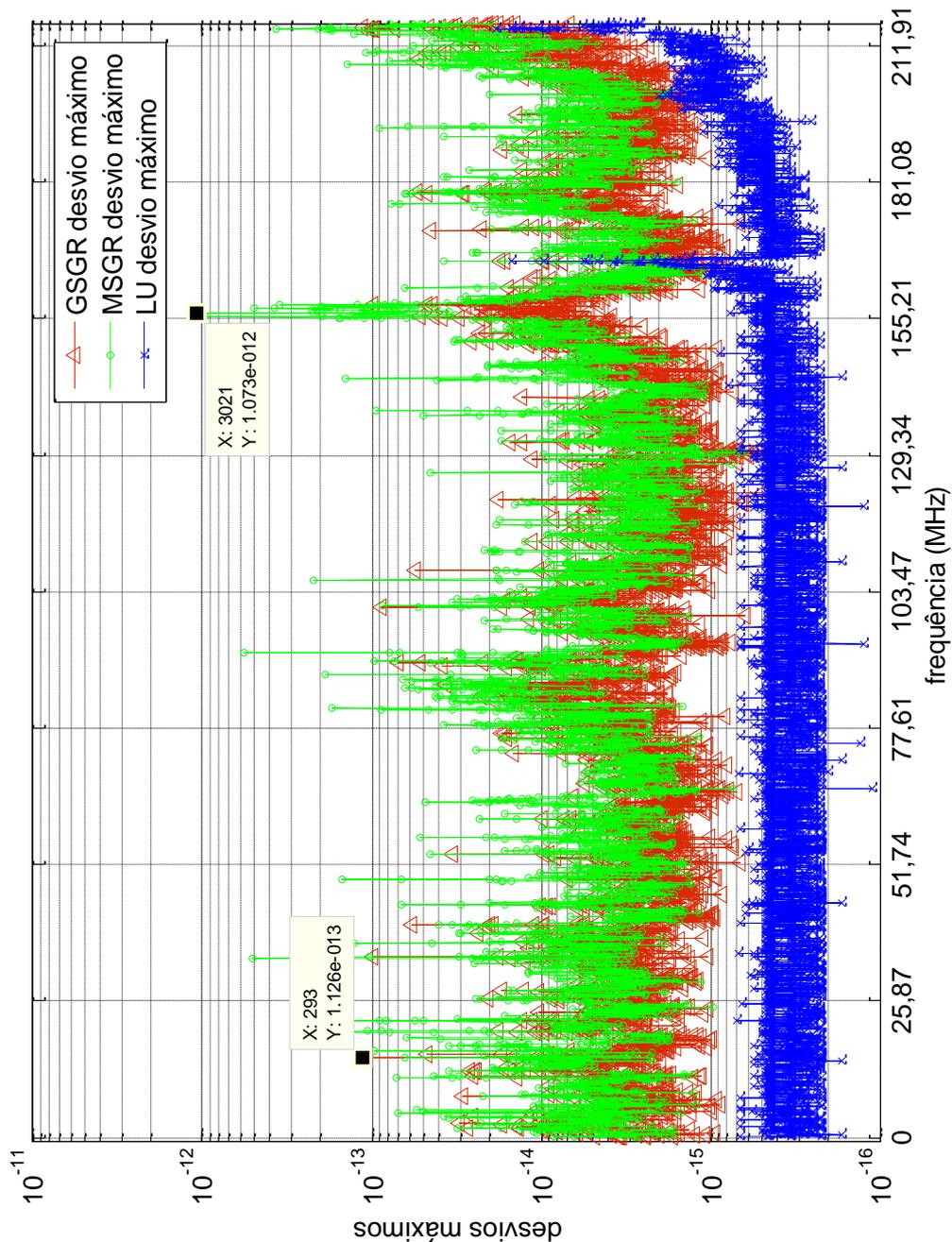


Figura 6.8 Comparação de desvios máximos de  $|\mathbf{I} - \mathbf{A} \cdot \mathbf{A}^{-1}|$  para os algoritmos GSGR, MSGR e LU em precisão dupla (PD). As matrizes  $\mathbf{A}$  usadas nesta simulação são matrizes de ganho de canal de um cabeamento de 30m e com 10 linhas, obtidas de medições realizadas nas frequências desde 0 até 211,91 MHz.

### 6.3 RESULTADOS DE TESTES DE FIGURAS DE MÉRITO

Para melhor caracterizar o desempenho do algoritmo proposto GSGR, quando aplicado a *G.fast* ou a outro sistema de comunicação, foram calculados figuras de mérito comumente relacionadas: a taxa sinal-ruído SNR, a distribuição de potência PSD e a taxa de transmissão de dados alcançável em função do cancelamento de *crosstalk*. Os resultados foram comparados aos obtidos quando aplicada a função de inversão matricial (Inv) do Matlab.

Foram utilizados os seguintes parâmetros na realização das simulações:

- Ruído de fundo: -150 dBm/Hz;
- PSD máxima: -76 dBm/Hz;
- Ganho de código: 5 dB;
- Margem da taxa sinal-ruído: 6 dB;
- *Shannon gap*: 9.8dB;
- Eficiência espectral máxima: 12 bps/Hz;
- Eficiência espectral mínima: 0.5 bps/Hz;
- *Framing overhead*: 12%;
- Largura de banda de frequência entre subportadoras: 51,750 kHz;
- Tom inicial: 45.

As matrizes de canais utilizadas nas simulações são de medições de canais reais, conforme descrito a seguir [25-27].

Os resultados obtidos, conforme apresentado nas Figs. 6.9 a 6.13, indicam que a acurácia do GSGR equivale à da função Inv para esta aplicação em pré-codificação especificamente.

a) Matrizes de um cabeamento de 100 metros e 24 linhas [25]:

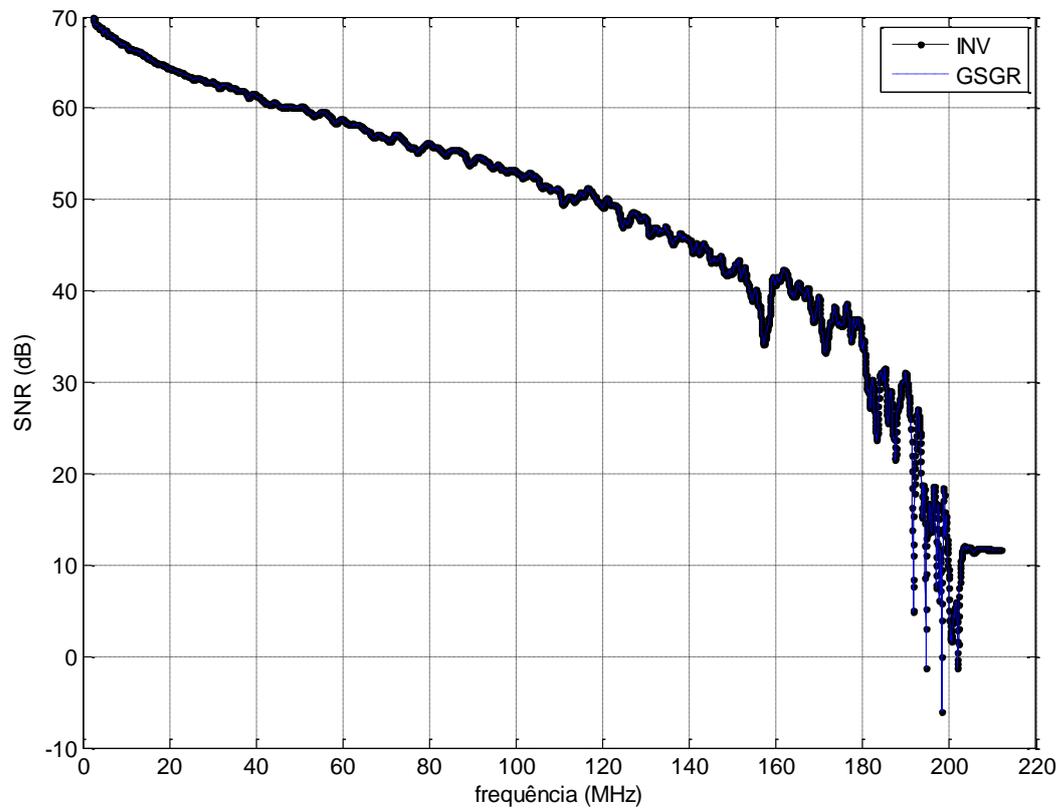


Figura 6.9 Taxas de sinal-ruído (SNR) obtidas com pré-codificação para cancelamento de *crosstalk*, usando o GSGR e INV-Matlab. Dados de um cabeamento de 100 metros e 24 linhas.

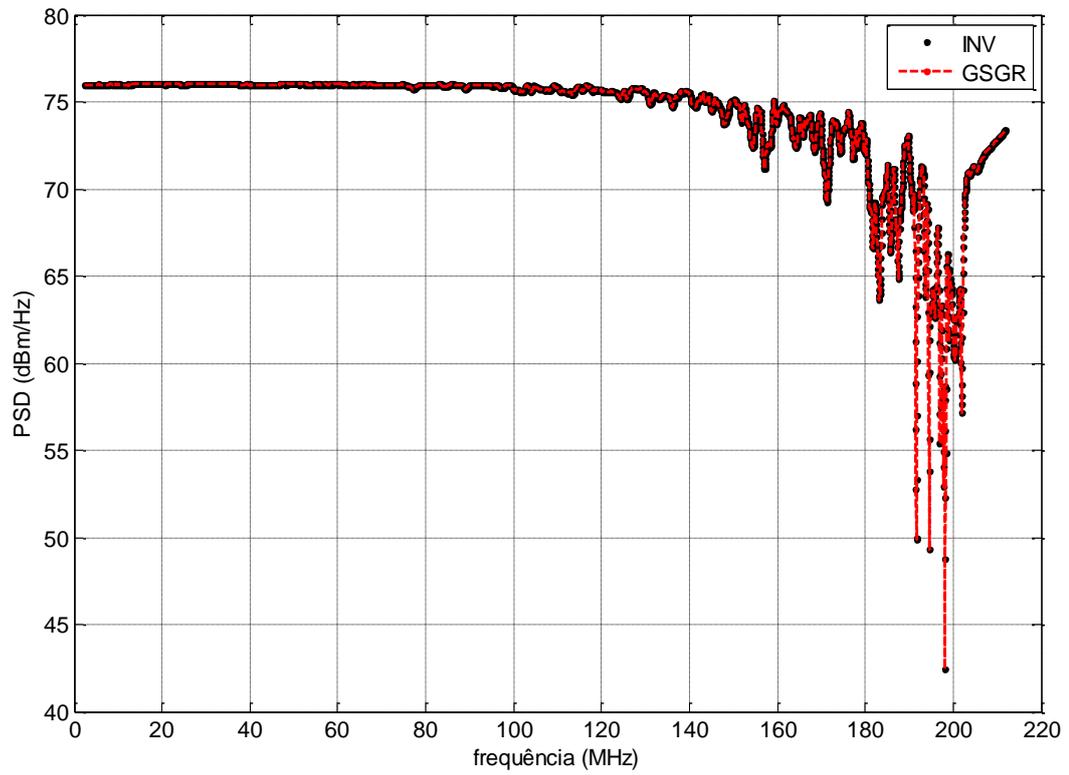


Figura 6.10 Densidade de potência (PSD) média entre linhas após pré-codificação para cancelamento de *crosstalk*, usando o GSGR e INV-Matlab. Dados de um cabeamento de 100 metros e 24 linhas.

b) Matrizes de um cabeamento de 30 metros e 10 linhas [26]:

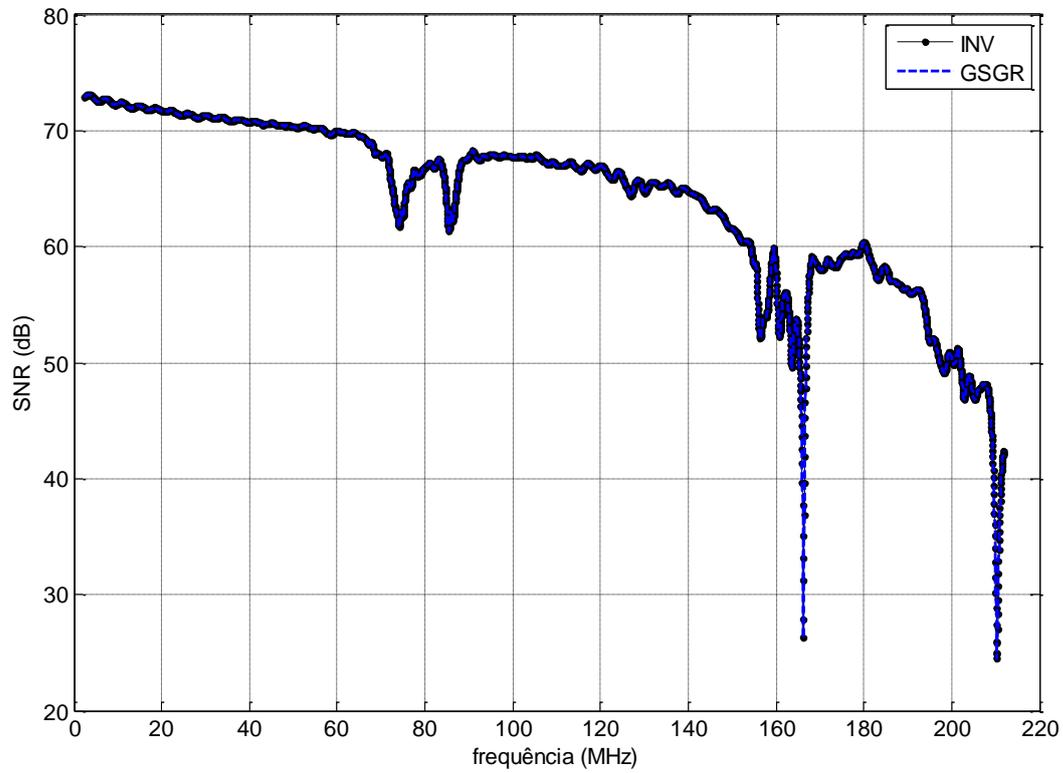


Figura 6.11 Taxas de sinal-ruído (SNR) obtidas com pré-codificação para cancelamento de *crosstalk*, usando o GSGR e INV-Matlab. Dados de um cabeamento de 30 metros e 10 linhas.

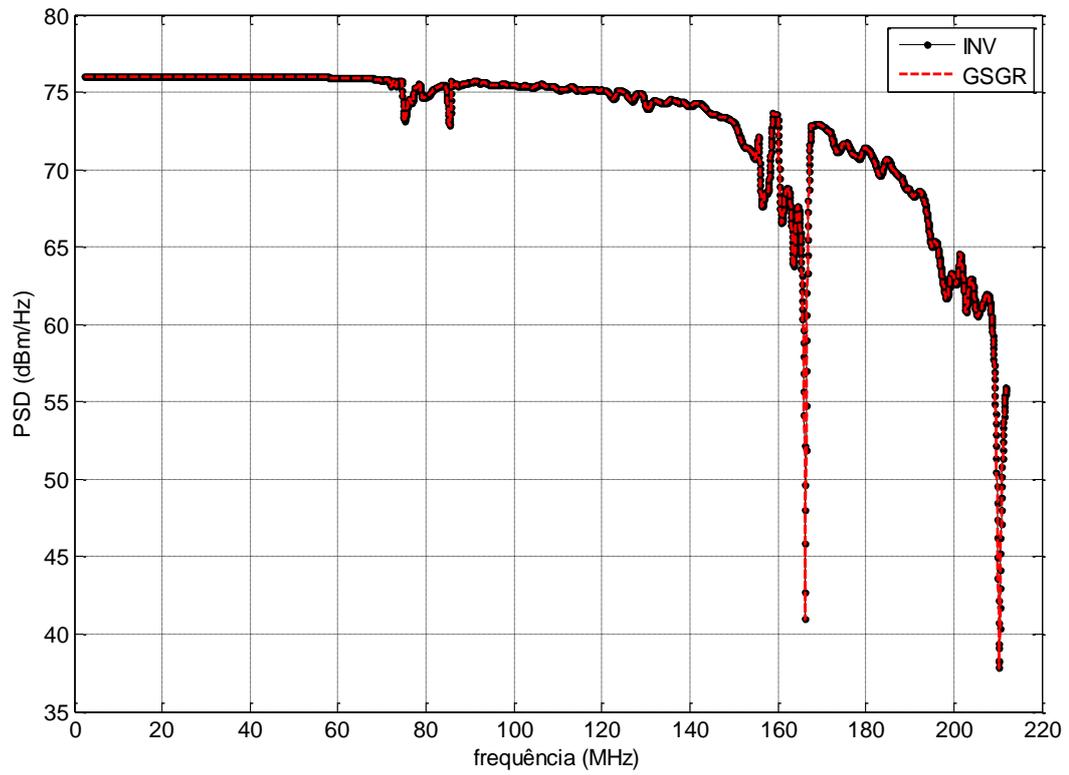


Figura 6.12 Densidade de potência (PSD) média das linhas após pré-codificação para cancelamento de *crosstalk*, usando o GSGR e INV-Matlab. Dados de um cabeamento de 100 metros e 24 linhas.

c) Matrizes de cabeamentos de 30, 100, 150, 200 e 250 metros e 8 linhas [26]:

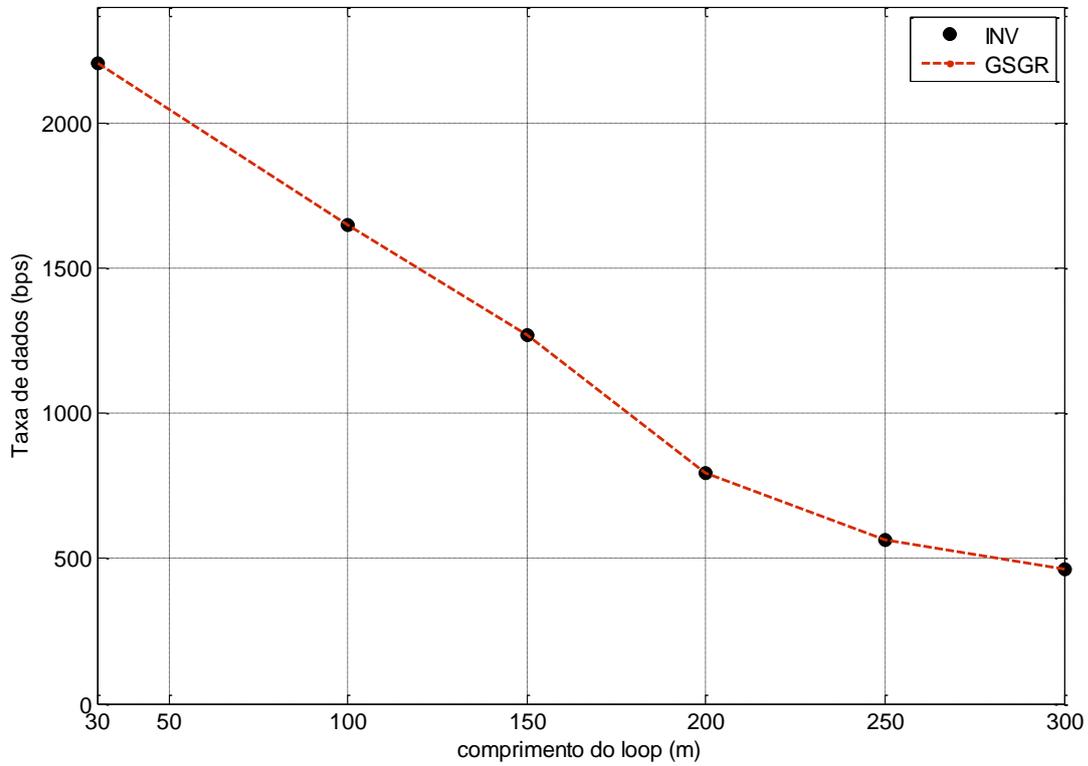


Figura 6.13 Taxa de transmissão de dados das linhas após pré-codificação para cancelamento de *crossstalk*, usando o GSGR e INV-Matlab. Dados de um cabeamento de 30, 100, 150, 200 e 250 metros e 8 linhas.

## 7 CONCLUSÕES E TRABALHOS FUTUROS

Este trabalho propôs um método direto de inversão matricial baseado em decomposição QR e em rotações de *Givens* (GR), chamado de GSGR.

Comparados com os métodos diretos, os métodos iterativos de inversão podem precisar de menos recursos de armazenamento de memória, podem ser eventualmente mais rápidos e podem explorar melhor características de estruturas dominantes das matrizes, como esparsidade [3]. Em contrapartida os métodos diretos calculam uma solução exata após um número fixo e pré-especificado de operações, sendo por isso mais apropriados para processamento de tarefas em paralelo e implementação em *hardwares* embarcados, normalmente com recursos computacionais reduzidos em comparação com plataformas de hardware mais sofisticadas.

Os testes, executados em uma plataforma DSP com um chip SoC TI TMS320C6670, de inversão de matrizes complexas representadas em 32 bits mostraram que quando comparado com um algoritmo de alta velocidade como o MSGR [18] e com um algoritmo de alta acurácia como o LU [1], o novo método GSGR reduziu o número de operações em ponto flutuante para 33% e 27%, e reduziu o tempo de execução para 55% e 38%, respectivamente, com um desvio máximo do resultado ideal de  $1,1 \times 10^{-2}$  com operações em ponto flutuante e precisão simples.

Os testes executados com a plataforma de *software* Matlab e dados de medições de canais reais mostraram que o uso do GSGR produz resultados com acurácia bastante próxima às obtidas com funções de inversão complexas e sofisticadas, quando usado para a aplicação específica de pré-codificação em *Vectoring* para sistemas xDSL, mas com a vantagem de execução mais rápida.

Para futuros trabalhos pode-se observar que dentre as estratégias adotadas para o desenvolvimento do padrão *G.fast* está a operação descontínua das linhas para reduzir o consumo de potência, o que implicará no chaveamento das linhas entre estados normais de operação e estados de baixo consumo de energia. Esse modo de operação descontínua provocará a alteração ainda mais frequente das matrizes de canal. Dessa forma, para a realização da compensação de *crosstalk* - *Vectoring* - serão necessários arranjos e técnicas inovadores para inversão matricial, ainda mais rápidos e inteligentes, para calcular ou aproximar a atualização das matrizes pré-codificadoras. Uma possível abordagem ao problema poderia ser a aplicação do GSGR associado ao método de *Sherman-Morrison* para inversão sucessiva de matrizes.

## 8 REFERÊNCIAS

- [1] G. Golub and C. Van Loan, *Matrix Computations*. Baltimore, MD: The Johns Hopkins Univ. Press, 1996.
- [2] S. Haykin, *Adaptive Filter Theory*, 2<sup>nd</sup> Ed., Prentice-Hall, 1991.
- [3] G. W. Stewart, *Matrix Algorithms*, University of Maryland, College Park, Maryland, vol. 1, 1998.
- [4] R. Dohler, “Squared givens rotations,” *IMA J. Numer. Anal.*, vol. 11, pp. 1–5, 1991.
- [5] D. S. Watkins, *Fundamentals of Matrix Computations*. New York: John Wiley & Sons, Inc, 2<sup>nd</sup> Ed., 2002.
- [6] Institute of Electrical and Electronics Engineers (IEEE), Inc., “Standard for Floating-Point Arithmetic”, IEEE Std-754, Aug. 2008.
- [7] M. Timmers, M. Guenach, C. Nuzman and J. Maes, “*G.fast*: Evolving the Copper Access”, *IEEE Commun. Magazine*, *Telecommunications Standards*, pp. 74-79, 2013.
- [8] P. Ödling, T. Magesacher, S. Höst, P.O. Börjesson, M. Berg and E. Areizaga, “The Fourth Generation Broadband Concept”, *IEEE Commun. Magazine*, vol. 47, no. 1, pp. 63-69, 2009.
- [9] C. Leung, S. Huberman, K. Ho-Van and T. Le-Ngoc “Vectored DSL Potential Implementation Issues and Challenges ”, *IEEE Commun. Surveys & Tutorials*, vol. PP, no. 99, pp. 1-17, 2013.
- [10] P. Golden, H. Dediue, and K. S. Jacobsen, Eds., *Fundamentals of DSL Technology*. Auerbach Publications, 2006.
- [11] V. Oksman, H. Schenk, A. Clausen, J.M. Cioffi, M. Mohseni, G. Ginis, C. Nuzman, J. Maes, M. Peeters, K. Fisher, and P-E. Eriksson “The ITU-T's new g.vector standard proliferates 100Mb/s DSL“, *IEEE Commun. Magazine*, vol.48, pp. 140-148, 2010.
- [12] International Telecommunication Union, “Self-FEXT cancellation (vectoring) for use”, *ITU-T Rec. 993.5*, Amd. 1, Dec. 2011.
- [13] International Telecommunication Union, “Self-FEXT cancellation (vectoring) for use”, *ITU-T Rec. 993.5*, Apr. 2010.
- [14] R. Cendrillon, G. Ginis, E. Van den Bogaert, and M. Moonen, “A near-optimal linear crosstalk precoder for downstream VDSL,” *IEEE Transactions on Communications*, vol. 55, no. 5, pp. 860–863, May 2007.
- [15] R. Cendrillon, G. Ginis, E. Van den Bogaert, and M. Moonen, “A near-optimal linear crosstalk canceller for upstream VDSL,” *IEEE Transactions on Signal Processing*, vol. 54, no. 8, pp. 3136–3146, Aug 2006.
- [16] International Telecommunication Union, “Very high speed digital subscriber line transceivers 2”, *ITU-T Rec. G.993.2*, Dec. 2011.
- [17] International Telecommunication Union, “Handshake procedures for digital subscriber line transceivers”, *ITU-T Rec. G.994.1*, Jun. 2012.
- [18] International Telecommunication Union, “Physical layer management for digital subscriber line transceivers”, *ITU-T Rec. 997.1*, Jun. 2012.

- [19] Givens, Wallace. "Computation of plane unitary rotations transforming a general matrix to triangular form". *J. SIAM* 6(1), pp. 26-50, 1958.
- [20] L. Ma, K. Dickson, J. McAllister, and J. McCanny, "QR Decomposition-Based Matrix Inversion for High Performance Embedded MIMO Receivers", *IEEE Trans. Signal Process.*, vol. 59, no. 4, pp.1858-1867, April, 2011.
- [21] S. Yokoyama, K. Matsumoto, and S. G. Sedukhin, "Matrix Inversion on the Cell/B.E.Process", 11th IEEE International Conference on High Performance Computing and Commun., pp. 148-153, 2009.
- [22] F. Zhang, ed., *The Schur Complement and its Applications*, NY: Springer, 2005.
- [23] Texas Instruments, TMS320C6670 Data Manual "Multicore Fixed and Floating-Point System-on-Chip", lit. no. SPRS689D, Mar. 2012.
- [24] Texas Instruments, TMS320C66x DSP "CPU and Instruction Set – Reference Guide", lit. no. SPRUGH7, Nov. 2010.
- [25] D. Acatauassu, I. Almeida, F. Muller, A. Klautau, C. Lu, K. Ericson, B. Dortschy, "Measurement and modeling techniques for the fourth generation broadband over copper", ch. 13, *Advanced topics in measurements*, Prof. Zahurul Haq (Ed.), Mar. 2012.
- [26] International Telecommunication Union, Deutsche Telekom AG, "G.fast: Cable Measurements for typical DTAG Installation Cable", *Contrib. ITU-T SG15/Q4 2013-05-Q4-040*, May 2013.
- [27] J. Maes, M. Guenach, K. Hooghe and M. Timmers, "Pushing the limits of copper", *IEEE ICC Selected Areas in Commun. Symposium*, pp. 3149-3153, 2012.
- [28] R. Zidane, S. Huberman, C. Leung, and T. Le-Ngoc, "Vectored DSL: Benefits and Challenges for Service Providers," *IEEE Commun. Mag.*, vol. 51, no. 2, pp. 152-157, 2013.
- [29] M. Ylinen, A. Burian, and J. Takala, "Updating matrix inverse in fixed-point representation: Direct versus iterative methods," in *Proc. Int. Symp. System-on-Chip*, pp. 45-48, 2003.
- [30] H. Anton, R. C. Busby, *Álgebra Linear Contemporânea*, Ed. Artmed, 2003.

## A APÊNDICE: CONCEITOS BÁSICOS DE MATRIZES EM ÁLGEBRA LINEAR

Nesta seção são indicados alguns conceitos básicos de álgebra linear relacionados a operações de inversão com matrizes, tais como espaços vetoriais, bases, dimensão, independência, etc. Para uma abordagem matemática mais completa indica-se as referências [1][3] e [4].

### A1. ESPAÇOS VETORIAIS

Para definir um espaço vetorial é necessário antes definir um corpo numérico. Um corpo numérico é um subconjunto  $K$  do conjunto de números complexos  $\mathbf{C}$ , fechado com relação às operações elementares de soma, multiplicação e divisão (com divisor diferente de zero). Ou seja, ao efetuar-se essas operações elementares com elementos de  $K$  obtém-se sempre elementos pertencentes a  $K$ .

Define-se então um espaço vetorial como um conjunto não vazio  $V$  de elementos vetores ou matrizes, sobre um corpo numérico  $K$ , com operações elementares de adição, multiplicação e divisão fechadas com relação às combinações lineares. Assim, o resultado de qualquer combinação linear de um conjunto de vetores ou matrizes pertencentes a um espaço vetorial está contido no próprio espaço vetorial.

### A.2 SUBESPAÇOS

Seja  $S$  um subconjunto não vazio de um espaço vetorial  $V$  sobre um corpo numérico  $K$ . Se  $S$  é também um espaço vetorial sobre  $K$ , então  $S$  é chamado de subespaço vetorial de  $V$ .

Um conjunto não vazio de vetores  $\{\mathbf{a}_1, \dots, \mathbf{a}_n\}$  define um subespaço vetorial  $S$  se ele é fechado com relação às combinações lineares de seus elementos. Assim refere-se ao conjunto de todas as combinações lineares de um conjunto de vetores  $\{\mathbf{a}_1, \dots, \mathbf{a}_n\} \in \mathbf{R}^m$ , como  $span \{\mathbf{a}_1, \dots, \mathbf{a}_n\}$ :

$$span \{a_1, \dots, a_n\} = \left\{ \sum_{j=1}^n \alpha_j a_j, \alpha_j \in \mathbf{R} \right\} \quad (\text{a.1})$$

As matrizes têm subespaços associados a elas. O subespaço coluna de uma matriz é o subespaço abrangido pelo conjunto de todas as combinações lineares dos vetores colunas.

### A3. VETORES LINEARMENTE INDEPENDENTES

Um conjunto de vetores  $\{\mathbf{a}_1, \dots, \mathbf{a}_n\}$  no espaço vetorial finito  $\mathbf{R}^m$ , com  $n < m$ , é dito linearmente independente se o somatório de:

$$\sum_{j=1}^n \alpha_j \mathbf{a}_j = 0 \quad (\text{a.2})$$

implica que todos os números escalares  $\alpha_j$  sejam nulos. Se existe uma combinação não trivial dos vetores  $\mathbf{a}_j$  que também resulte em um somatório nulo, então o conjunto de vetores é dito linearmente dependente.

O subconjunto  $S_i = \{\mathbf{a}_{i1}, \dots, \mathbf{a}_{ik}\}$  é considerado linear maximamente independente de  $S = \{\mathbf{a}_1, \dots, \mathbf{a}_n\}$  se  $S_i$  é linearmente independente e não está contido propriamente em nenhum outro subconjunto linearmente independente de  $S$ .

### A4. BASES E DIMENSÃO

Um subespaço vetorial pode ser representado por um conjunto finito de vetores chamado de base.

Se  $S_i$  é um subconjunto de vetores maximamente linearmente independente de um conjunto  $S$  de vetores, então:

$$\text{span } S_i = \text{span } S \quad (\text{a.3})$$

Portanto  $S_i$  é uma base do  $\text{span}$  de  $S$ .

Para todo subespaço  $S$  contido em  $\mathbf{R}^m$  é possível encontrar subconjuntos  $S_i$  de vetores independentes tal que:

$$S = \text{span } \{\mathbf{a}_{i1}, \dots, \mathbf{a}_{ik}\} \quad (\text{a.4})$$

Todos os subconjuntos bases para um subespaço  $S$  tem o mesmo número de elementos, chamado de dimensão de  $S$ .

### A5. SUBESPAÇOS FUNDAMENTAIS

O subespaço definido pelos vetores coluna de uma matriz  $\mathbf{A}$  e o subespaço nulo são importantes conceitos para análise das características de uma matriz [30].

O subespaço definido pelos vetores coluna  $\{\mathbf{a}_1, \dots, \mathbf{a}_n\}$  de uma matriz  $\mathbf{A}$ , também chamado de imagem ou extensão, é definido por:

$$Col(\mathbf{A}) = \{ \mathbf{y} \in \mathbf{R}^m \mid \mathbf{y} = \mathbf{A}\mathbf{x}, \mathbf{x} \in \mathbf{R}^n \} \quad (\text{a.5})$$

O subespaço nulo de uma matriz  $\mathbf{A}$  é definido por:

$$Nuc(\mathbf{A}) = \{ \mathbf{x} \in \mathbf{R}^n \mid \mathbf{A}\mathbf{x} = \mathbf{0} \} \quad (\text{a.6})$$

#### A6. POSTO DE UMA MATRIZ

O posto de uma matriz  $\mathbf{A}$  é definido como o valor da dimensão do espaço vetorial de imagem de  $\mathbf{A}$ :

$$rank(\mathbf{A}) = dim(ran(\mathbf{A})) \quad (\text{a.7})$$

#### A7. FATORAÇÃO

Qualquer matriz  $\mathbf{A} \in \mathbf{R}^{m \times n}$  pode ser representada por uma fatoração da forma:

$$\mathbf{A} = \mathbf{X} \cdot \mathbf{Y}^T \quad (\text{a.8})$$

onde  $\mathbf{X} \in \mathbf{R}^{m \times k}$  e  $\mathbf{Y} \in \mathbf{R}^{n \times k}$  tem colunas linearmente independentes.

#### A8. INVERSA DE UMA MATRIZ

Se  $\mathbf{A}$  e  $\mathbf{X}$  são matrizes em  $\mathbf{R}^{n \times n}$  e satisfazem a equação  $\mathbf{A} \cdot \mathbf{X} = \mathbf{I}$ , onde  $\mathbf{I}$  é a matriz identidade, então  $\mathbf{X}$  é a matriz inversa de  $\mathbf{A}$ , e é indicada como  $\mathbf{A}^{-1}$ .

Se  $\mathbf{A}^{-1}$  existe,  $\mathbf{A}$  é dita uma matriz não singular. Se  $\mathbf{A}^{-1}$  não existe, diz-se que  $\mathbf{A}$  é singular.