

NOVOS ALGORITMOS DE FILTRAGEM ADAPTATIVA: LMS COM
REUTILIZAÇÃO DE DADOS E RLS RÁPIDO BASEADO EM
DECOMPOSIÇÃO QR

José Antonio Apolinário Junior

TESE SUBMETIDA AO CORPO DOCENTE DA COORDENAÇÃO
DOS PROGRAMAS DE PÓS-GRADUAÇÃO DE ENGENHARIA DA
UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE
DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU
DE DOUTOR EM CIÊNCIAS EM ENGENHARIA ELÉTRICA.

Aprovada por:

Prof. Paulo S. R. Diniz, Ph.D.
(Presidente)

Dr. Mariane R. Petraglia, Ph.D.

Dr. Max Gerken, Dr.Ing.

Dr. Abraham Alcaim, Ph.D.

Dr. Marcello L. R. de Campos, Ph.D.

RIO DE JANEIRO, RJ - BRASIL

JUNHO DE 1998

APOLINÁRIO JUNIOR, JOSÉ ANTONIO

Novos Algoritmos de Filtragem Adaptativa: LMS com Reutilização de Dados e RLS Rápido Baseado em Decomposição QR [Rio de Janeiro] 1998

XV, 137 p. 29,7 cm (COPPE/UFRJ, D.Sc., Engenharia Elétrica, 1998)

Tese - Universidade Federal do Rio de Janeiro, COPPE

1. Filtragem Adaptativa

I. COPPE/UFRJ II. Título (série)

À minha esposa *Ana Luisa*,
à minha filha *Isabela* e ao
meu filho *Eduardo*. O autor
ama muito vocês todos.

Agradecimentos

Por mais de três anos tenho trabalhado em meu programa de doutoramento e durante este período fui muito feliz em conviver com pessoas muito boas e em ter tido o apoio de todas elas.

Em primeiro lugar, o meu muito obrigado ao meu orientador, Dr. Paulo Diniz, pela orientação, incentivo e amizade. Sou muito grato ao meu amigo e mestre (*Jedi*) Diniz que com seu jeito descontraído mas ao mesmo tempo profissional e extremamente competente permitiu-me vencer desafios e ultrapassar barreiras com mais confiança. Divido com ele os louros deste trabalho. Sou também grato ao pessoal da COPPE/UFRJ, aos professores, aos alunos e a todos que lá trabalham e que, de uma maneira ou de outra, me ajudaram nesta difícil tarefa de me tornar um doutor. Agradeço também à Liga da Justiça e aos super-amigos Robin, (Supper) He-Man, A.-M. e, mais recentemente, Fylo-Man. Obrigado Rogério Guedes, Dr. Sérgio Lima Netto, Dr. A.-M. e *tack* ao Mestre em Caipiroskas *in spe* Stefan Werner. Em especial, obrigado ao meu amigo, co-autor e membro da banca Marcello Campos pela sua amizade, parceria e, espero, misericórdia (eu escrevi estes agradecimentos antes do dia D). É também com prazer que agradeço ao Prof. Timo Laakso e sua equipe do *Laboratory of Telecommunications Technology, Helsinki University of Technology*, pela cooperação frutífera durante o período em que lá estive como pesquisador visitante, *kiitos*. Sou também muito grato ao Instituto Militar de Engenharia e ao Exército Brasileiro pela oportunidade que me foi dada de realizar o programa de doutoramento.

Este trabalho foi parcialmente apoiado pela CAPES — Ministério da Educação, Brasil.

Resumo da Tese apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Doutor em Ciências (D.Sc.)

**NOVOS ALGORITMOS DE FILTRAGEM ADAPTATIVA: LMS COM
REUTILIZAÇÃO DE DADOS E RLS RÁPIDO BASEADO EM
DECOMPOSIÇÃO QR**

José Antonio Apolinário Junior

Junho/1998

Orientador: Paulo Sérgio Ramirez Diniz

Programa: Engenharia Elétrica

Este trabalho apresenta dois novos algoritmos de filtragem adaptativa: um deles do tipo mínimo médio quadrático com reutilização de dados e o outro dos mínimos quadrados recursivo rápido baseado em decomposição QR.

A primeira parte deste trabalho apresenta e analisa um novo algoritmo do tipo LMS, o algoritmo LMS binormalizado com reutilização de dados (BNDR-LMS), o qual apresenta resultados favoráveis em relação a outros algoritmos LMS normalizados quando o sinal de entrada é correlacionado. Para este algoritmo, análises na média e na média quadrática são apresentadas e uma fórmula fechada para o

erro médio quadrático é fornecida para sinais de entrada brancos bem como sua extensão para o caso de ruído colorido. Um modelo simples para o vetor sinal de entrada que proporciona simplicidade e tratabilidade para a análise de estatísticas de segunda ordem é completamente descrito. A metodologia é prontamente aplicável a outros algoritmos adaptativos de difícil análise. Resultados de simulações mostram o desempenho do algoritmo BNDR-LMS para diferentes cenários e validam a análise e seus pressupostos. Uma extensão do algoritmo BNDR-LMS para incluir restrições é também derivada de modo a aplicar este algoritmo a um receptor móvel que emprega CDMA (direct-sequence code-division multiple access). Além disto, uma otimização do passo de convergência é proposta para levar a cabo esta aplicação com os requisitos conflitantes de convergência rápida e mínimo erro médio quadrático (MSE) em estado estacionário.

Na segunda parte deste trabalho, os princípios por trás da triangularização via decomposição QR da matriz de dados de entrada ponderada e os tipos de erros usados no processo de atualização são explorados de modo a investigar as relações entre os diferentes membros desta família de algoritmos QR rápidos. Os algoritmos são classificados de acordo com uma estrutura geral e um novo algoritmo QR rápido baseado em rotações de Givens usando erros de predição progressiva *a priori* é apresentado junto com as descrições detalhadas dos quatro algoritmos QR rápidos classificados e duas versões em treliça. Finalmente, uma contribuição à análise em precisão finita dos algoritmos QR rápidos é apresentada.

Abstract of Thesis presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Doctor of Science (D.Sc.)

**NEW ADAPTIVE FILTERING ALGORITHMS: LMS WITH
DATA-REUSING AND FAST RLS BASED ON QR
DECOMPOSITION**

José Antonio Apolinário Junior

May/1998

Advisor: Paulo Sérgio Ramirez Diniz

Department: Electrical Engineering

This work presents two new adaptive filtering algorithms: one of them a least-mean-square type with data-reusing and the other one a fast recursive least-squares based on QR decomposition.

The first part of this work presents and analyzes a new LMS-like algorithm, the binormalized data-reusing least mean-square (BNDR-LMS) algorithm, which compares favorably with other normalized LMS-like algorithms when the input signal is correlated. For this algorithm, convergence analyses in the mean and in the mean-squared are presented and a closed-form formula for the mean-square error

is provided for white input signals as well as its extension to the case of colored input signal. A simple model for the input-signal vector which imparts simplicity and tractability to the analysis of second-order statistics is fully described. The methodology is readily applicable to other adaptation algorithms of difficult analysis. Simulation results show the performance of the BNDR-LMS algorithm for different scenarios and validate the analysis and ensuing assumptions. An extension of the BNDR-LMS algorithm to include constraints is also derived in order to apply this algorithm to a direct-sequence code-division multiple access (DS-CDMA) mobile receiver. Moreover, a step-size optimization is proposed to accomplish this practical application with the conflicting requirements of fast convergence and minimum steady-state mean-square error (MSE).

In the second part of this work, the principles behind the triangularization of the weighted input data matrix via QR decomposition and the type of errors used in the updating process are exploited in order to investigate the relationships among different fast algorithms of the QR decomposition family. The algorithms are classified according to a general framework and a new fast QR algorithm based on Givens rotation using *a priori* forward errors is introduced along with the detailed description of the four classified fast QR algorithms and two lattice versions. Finally, a contribution towards the finite precision analysis of the fast QR algorithms is presented.

Sumário

Ficha Catalográfica	ii
Dedicatória	iii
Agradecimentos	iv
Resumo	v
Abstract	vii
Índice do Texto	ix
Índice de Figuras	xiii
Índice de Tabelas	xiv
Lista de Abreviações	xv
1 Filtragem Adaptativa	1
1.1 Introdução	1
1.2 Conceitos Básicos de Filtragem Adaptativa	3
1.2.1 O Erro Médio Quadrático e os Algoritmos Baseados no LMS	4
1.2.2 Os Mínimos Quadrados e os Algoritmos RLS	5
1.3 Algoritmos LMS, NLMS e com Reutilização de Dados	6
1.4 Introduzindo a Decomposição QR	9
1.5 Contribuições Originais	12

2	O Algoritmo BNDR-LMS	13
2.1	Definição do Problema e Derivação do Algoritmo	14
2.1.1	Versão Simplificada	15
2.1.2	Derivação Geométrica	15
2.2	Análise de Convergência do Vetor de Coeficientes	17
2.3	Análise de Estatística de Segunda Ordem	20
2.3.1	Sinal de Entrada Branco	20
2.3.2	Sinal de Entrada Colorido	24
2.4	Resultados das Simulações	27
2.5	Conclusões	33
3	Uma Aplicação Prática do Algoritmo BNDR-LMS	35
3.1	Introdução	35
3.2	Re-Derivação dos Algoritmos NLMS e BNDR-LMS	36
3.3	O Algoritmo com Restrições	38
3.4	Otimização do Passo do Algoritmo BNDR-LMS	40
3.5	Resultados das Simulações	47
3.6	Conclusões	49
4	Algoritmos QR Rápidos: uma Abordagem Unificada	50
4.1	Introdução	50
4.1.1	O Algoritmo QR Convencional	50
4.1.2	Interpretando as Variáveis Internas	56
4.1.3	O Algoritmo QR Inverso	66
4.2	Classificação dos Algoritmos QR Rápidos	68
4.3	Algoritmos que Usam Triangularização Superior (Atualizando Erros de Predição Progressiva)	70
4.3.1	O Algoritmo FQR_POS_F	72
4.3.2	O Novo Algoritmo FQR_PRI_F	74
4.4	Algoritmos que Usam Triangularização Inferior (Atualizando Erros de Predição Regressiva)	74

4.4.1	O Algoritmo FQR_POS_B	76
4.4.2	O Algoritmo FQR_PRI_B	76
4.5	Conclusões	77
5	As Versões em Treliça dos Algoritmos QR Rápidos	83
5.1	Introdução	83
5.2	Derivando as Versões em Treliça	85
5.3	Resultados das Simulações	88
5.4	Conclusões	89
6	Contribuições à Análise em Precisão Finita dos Algoritmos QR Rápidos	95
6.1	Introdução	95
6.2	Análise em Precisão Infinita	96
6.2.1	Resultados de Precisão Infinita para o Algoritmo FQR_POS_B	96
6.2.2	Resultados de Precisão Infinita para o Algoritmo FQR_PRI_B	99
6.3	Contribuição à Análise em Precisão Finita	100
6.3.1	Modelo do Erro de Quantização de Ponto Fixo	100
6.3.2	O Algoritmo FQR_PRI_B: Valor Médio-Quadrático de $\Delta e_{fq_1}^{(i)}(k)$ e $\Delta d_{fq_2_i}(k)$	101
6.3.3	O Algoritmo FQR_PRI_B: Valor Médio-Quadrático de Δaux_0	104
6.3.4	Refinando as aproximações de $E[1/x^2]$ e $E[1/x^4]$	106
6.4	Resultados das Simulações	108
6.5	Conclusões	110
7	Conclusões e Sugestões	112
7.1	Conclusões	112
7.2	Sugestões para Pesquisa Futura	114

Lista de Figuras

1.1	Configuração básica de um filtro adaptativo	3
1.2	Atualização do vetor de coeficientes:	10
1.3	Os algoritmos RLS.	11
2.1	Excesso de MSE para vetores sinal de entrada paralelos.	24
2.2	Excesso de MSE para vetores sinal de entrada ortogonais.	25
2.3	Excesso de MSE para um vetor sinal de entrada modelado.	26
2.4	MSE para os algoritmos NLMS, NNDR-LMS e BNDR-LMS.	29
2.5	Excesso de MSE para $N = 5$ como função de μ	30
2.6	Excesso de MSE para $N = 10$ como função de μ	31
2.7	Excesso de MSE para $N = 63$ como função de μ	32
2.8	Excesso de MSE para sinais de entrada coloridos.	33
3.1	Seqüências de $\mu(k)$ ótimas para o algoritmo BNDR-LMS.	43
3.2	Seqüência de passos ótima e duas classes de seqüências aproximadas.	44
3.3	Curvas de aprendizado para passo fixo, passo variável ótimo e suas duas aproximações.	45
3.4	Comparando as curvas de aprendizagem para o caso de sinal de entrada colorido.	46
3.5	Curvas de aprendizagem dos algoritmos LMS, NLMS e BNDR-LMS com restrições.	48
4.1	As diferentes triangularizações de $\mathbf{U}(k)$: (a) SUPERIOR e (b) INFERIOR.	55
5.1	Um estágio da estrutura em treliça do algoritmo FQR_POS_B.	87

5.2	Um estágio da estrutura em treliça do algoritmo FQR_PRLB.	88
5.3	Desempenho dos algoritmos num ambiente de precisão finita (varian- do B, o número de bits da mantissa).	89
5.4	MSE em db para diferentes valores de λ	90

Lista de Tabelas

2.1	O algoritmo “Binormalized Data-Reusing LMS”	17
2.2	Excesso de Erro Médio Quadrático	29
2.3	Comparação de complexidade computacional (para $L=1$)	34
3.1	Algoritmo para computar a seqüência de passo variável ótima.	42
4.1	As equações do algoritmo QR convencional.	57
4.2	As equações do algoritmo QR inverso.	69
4.3	Classificação dos algoritmos QR rápidos.	70
4.4	As equações do algoritmo FQR_POS_F.	78
4.5	As equações do algoritmo FQR_PRI_F.	79
4.6	As equações do algoritmo FQR_POS_B.	80
4.7	As equações do algoritmo FQR_PRI_B.	81
4.8	Comparação de complexidade computacional.	82
5.1	Variáveis usadas nos algoritmos FQR_POS_B e FQR_PRI_B.	84
5.2	A versão em treliça do algoritmo FQR_POS_B.	91
5.3	A versão em treliça do algoritmo FQR_PRI_B.	93
6.1	Comparação de desempenho das novas expressões.	108
6.2	Valor médio-quadrático de $\Delta e_{f_{q_1}}^{(i)}(k)$	109
6.3	Valor médio-quadrático de $\Delta d_{f_{q_2}}(k)$	110
6.4	Valor médio-quadrático de Δaux_0	110

Lista de Abreviações

BNDR-LMS	binormalized data-reusing LMS
DR-LMS	data-reusing LMS
DS-CDMA	direct sequence code-division multiple access
FIR	finite-duration impulse response
FQR	fast QR
FQR_POS_B	fast QR updating <i>a posteriori</i> backward prediction errors
FQR_POS_F	fast QR updating <i>a posteriori</i> forward prediction errors
FQR_PRL_B	fast QR updating <i>a priori</i> backward prediction errors
FQR_PRL_F	fast QR updating <i>a priori</i> forward prediction errors
FQR-L	fast QR lattice
FTRLS	fast transversal RLS
GSC	general sidelobe canceler
IIR	infinite-duration impulse response
IQR	inverse QR
LMS	least mean-square
LRLS	lattice RLS
LS	least-squares
MSE	mean-square error
NLMS	normalized LMS
NNDR-LMS	normalized new data-reusing LMS
QRD	QR decomposition
RLS	recursive least-squares
UNDR-LMS	unnormalized new data-reusing LMS

Capítulo 1

Filtragem Adaptativa

1.1 Introdução

Nas últimas décadas, o campo do processamento digital de sinais, e particularmente o de processamento adaptativo de sinais, tem se desenvolvido enormemente devido à tecnologia crescentemente disponível para a implementação de algoritmos modernos. Estes algoritmos têm sido aplicados a um extensivo número de problemas incluindo cancelamento de ruído e de eco, equalização de canais, predição de sinais, arranjos (*arrays*) adaptativos bem como muitos outros. Uma aplicação particular exemplificada no Capítulo 3 é a detecção multiusuária adaptativa num receptor móvel DS-CDMA que mostra a utilidade de técnicas de filtragem adaptativa em sistemas móveis de comunicações.

Um filtro adaptativo pode ser definido como um filtro digital auto-modificável que ajusta seus coeficientes de modo a minimizar uma função de erro. Esta função de erro ou função custo é obtido da diferença entre o sinal “referência” ou “desejado” e a saída do filtro. Algoritmos de filtragem adaptativa são de fato intimamente relacionados às técnicas de otimização clássicas embora nestas últimas todos cálculos são executados “offline”. Além disto, de um filtro adaptativo é algumas vezes esperado que ele acompanhe o filtro ótimo (ou filtro de Wiener como é chamado o filtro ótimo no sentido médio quadrático para um ambiente estacionário) num ambiente ligeiramente variante no tempo.

De modo a comparar a grande variedade de algoritmos disponíveis na literatura de filtragem adaptativa, os seguintes aspectos devem ser levados em consideração [1].

- Estrutura. A maneira pela qual o algoritmo é implementado pode ser basicamente dividida em dois tipos para os filtros adaptativos FIR (finite impulse response): filtro transversal (ou tapped-delay line como é conhecido no idioma inglês) e estrutura em treliça (lattice em inglês). Filtros adaptativos IIR (infinite impulse response) constituem um outro campo dentro da filtragem adaptativa onde podemos encontrar um grande número de realizações.
- Taxa de convergência, desajuste e acompanhamento (tracking). Supondo uma situação onde inexistente ruído de medição ou de modelagem, os coeficientes de um filtro adaptativo podem convergir rapidamente ou lentamente para a solução ótima. Os coeficientes, em geral, não alcançarão os valores ótimos mas permanecerão próximos dos mesmos. Desajuste é a medida de quão próximos estes coeficientes (os estimados e os ótimos) estão em estado estacionário. Pode ser tomado como regra geral que para um dado algoritmo o mais rápido você o faz convergir, maior será seu desajuste. Em ambientes não-estacionários, o algoritmo deve ser rápido o suficiente para acompanhar os coeficientes ótimos variantes no tempo.
- Aspectos computacionais. Podem ser incluídos aqui a complexidade computacional bem como o desempenho do algoritmo num ambiente em precisão finita. O esforço em se obter versões rápidas dos algoritmos mais complexos resulta do desejo de reduzir a carga computacional a um número mínimo de operações e reduzir o tamanho da memória necessária para poder rodar estes algoritmos em aplicações em tempo real. Por outro lado, um ambiente em precisão finita gera erros de quantização os quais chamam a atenção de projetistas para estabilidade numérica, precisão numérica e robustez do algoritmo.

1.2 Conceitos Básicos de Filtragem Adaptativa

A definição da função custo dá espaço a um grande número de diferentes algoritmos de filtragem adaptativa. O erro médio quadrático é usado nos algoritmos conhecidos como mínimo (erro) médio quadrático ou “least mean-square” (LMS) e baseados no LMS, enquanto que os mínimos quadrados conduzem aos esquemas dos mínimos quadrados recursivos ou “recursive least-squares” (RLS). Os algoritmos RLS podem ser subdivididos em filtros adaptativos RLS convencional, em treliça, transversal rápido e baseado em decomposição QR.

A configuração básica de um filtro adaptativo é ilustrada na Figura 1.1. O sinal de entrada é denotado por $x(k)$ onde k é o número de iteração. O sinal de referência $d(k)$ pode ser visto (como num problema de identificação de sistema FIR) como o sinal desejado mais um ruído de observação, ou seja, $\mathbf{x}^T(k)\mathbf{w}_0 + n(k)$ onde \mathbf{w}_0 é o vetor de coeficientes ótimos e $\mathbf{x}(k)$ é o vetor $[x(k) \ x(k-1) \ \dots \ x(k-N)]^T$, com N sendo a ordem do filtro adaptativo. O sinal de erro é $e(k) = d(k) - y(k)$, onde $y(k)$ é a saída do filtro adaptativo. Este erro será usado no algoritmo de adaptação para atualizar o vetor de coeficientes $\mathbf{w}(k)$ do filtro adaptativo.

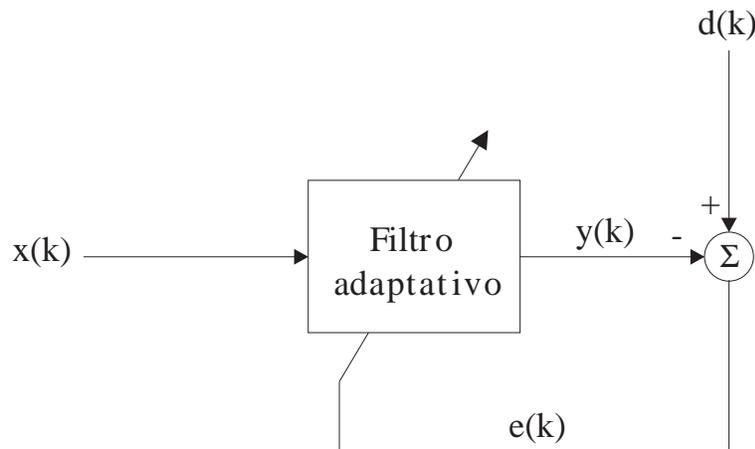


Figura 1.1: Configuração básica de um filtro adaptativo

1.2.1 O Erro Médio Quadrático e os Algoritmos Baseados no LMS

O erro médio quadrático ou “mean-square error” (MSE) é definido por

$$\xi(k) = E[e^2(k)] = E[(d(k) - y(k))^2] \quad (1.1)$$

onde $y(k) = \sum_{i=0}^N w_i(k)x(k-i) = \mathbf{x}^T(k)\mathbf{w}(k)$, com $\mathbf{w}(k) = [w_0(k)w_1(k) \cdots w_N(k)]^T$ sendo o vetor de coeficientes.

O vetor gradiente do MSE em relação ao vetor dos coeficientes é dado por

$$\nabla_{\mathbf{w}(k)}\xi(k) = -2\mathbf{p} + 2\mathbf{p}\mathbf{w}(k) \quad (1.2)$$

onde $\mathbf{p} = E[d(k)\mathbf{x}(k)]$ é o vetor de correlação cruzada entre os sinais desejado e de entrada, e $\mathbf{R} = E[\mathbf{x}(k)\mathbf{x}^T(k)]$ é a matriz de autocorrelação do sinal de entrada. A solução de Wiener é obtida igualando-se o vetor gradiente a zero e assumindo que \mathbf{R} é não-singular, sendo dada por

$$\mathbf{w}_0 = \mathbf{R}^{-1}\mathbf{p} \quad (1.3)$$

Nós podemos abordar a solução de Wiener buscando na direção da estimativa do vetor gradiente (algoritmo baseado na descendente máxima ou “steepest-descent-based algorithm” como é usualmente conhecido em inglês) usando um passo μ . Uma solução possível e muito simples é obtida usando-se estimativas instantâneas de \mathbf{R} e \mathbf{p} dadas por $\mathbf{x}(k)\mathbf{x}^T(k)$ e $d(k)\mathbf{x}(k)$. O algoritmo baseado no gradiente resultante é conhecido como algoritmo “least mean-square” (LMS).

O algoritmo LMS é muito popular e tem sido largamente usado devido a sua simplicidade. Sua velocidade de convergência, entretanto, é altamente dependente da razão de autovalores (número de condicionamento ou “condition number”) da matriz de auto-correlação do sinal de entrada [1, 2]. Esquemas alternativos que tentam melhorar este desempenho ao custo de uma complexidade computacional adicional mínima têm sido propostos e extensivamente discutidos no passado [2]–[5].

Uma abordagem que tem sido empregada com sucesso em situações onde as estatísticas do sinal são desconhecidas é o cálculo “online” do fator de convergência

o qual toma parte na atualização dos coeficientes do filtro [4, 6]. O algoritmo LMS normalizado (NLMS) pode ser incluído nesta categoria [4, 7]. Também pertencendo a esta categoria estão os algoritmos que reutilizam dados os quais serão descritos mais tarde em maiores detalhes.

1.2.2 Os Mínimos Quadrados e os Algoritmos RLS

Uma outra função objetivo determinística e conveniente de ser usada em ambiente estacionário é conhecida como mínimos quadrados ou “least-squares” (LS) sendo dada por $\frac{1}{k+1} \sum_{i=0}^k e^2(i)$. A computação dos mínimos quadrados numa forma recursiva resultou numa família de algoritmos conhecidos como mínimos quadrados recursivos ou “recursive least-squares” (RLS). Os algoritmos RLS são conhecidos por possuir uma taxa de convergência rápida que é independente da razão de autovalores da matriz de correlação do sinal de entrada. Eles são também bastante úteis em aplicações onde o ambiente é ligeiramente variante. O preço de todos estes benefícios deste algoritmo é um considerável aumento na complexidade computacional.

A função objetivo desta classe de algoritmos é dada por

$$\xi(k) = \sum_{i=0}^k \lambda^{k-i} e^2(i) = \sum_{i=0}^k \lambda^{k-i} [d(i) - \mathbf{x}^T(i) \mathbf{w}(k)]^2 \quad (1.4)$$

onde $e(i)$ é o erro de saída *a posteriori* no instante i e λ é o fator de “esquecimento”.

A solução ótima no sentido dos mínimos quadrados é dada após derivarmos $\xi(k)$ em relação a $\mathbf{w}(k)$ e igualarmos o resultado a zero. O resultado é dado pelo seguinte produto do inverso de uma matriz por um vetor.

$$\mathbf{w}(k) = \left[\sum_{i=0}^k \lambda^{k-i} \mathbf{x}(i) \mathbf{x}^T(i) \right]^{-1} \left[\sum_{i=0}^k \lambda^{k-i} \mathbf{x}(i) d(i) \right] \quad (1.5)$$

A simples computação da equação acima resulta num algoritmo com uma complexidade computacional da ordem de N^3 multiplicações ou $O[N^3]$. Entretanto, o cálculo da inversa com $O[N^3]$ multiplicações pode ser evitado pelo uso do chamado *lema de inversão de matriz* [2]. A relação resultante é usada no algoritmo RLS convencional cuja complexidade computacional é da ordem de N^2 . Esta complexidade

computacional pode cair para $O[N]$ quando o vetor de entrada consiste de versões com retardo de um mesmo sinal. Um número de $O[N]$ ou algoritmos **rápidos** estão disponíveis na literatura incluindo diferentes versões de algoritmos RLS em treliça ou “lattice” RLS [8] e o algoritmo RLS transversal rápido (FTRL) [9] o qual é considerado o mais rápido (no sentido de que um mínimo número de operações é necessário) embora não estável.

1.3 Algoritmos LMS, NLMS e com Reutilização de Dados

Como ressaltado anteriormente, o algoritmo LMS usa estimativas da matriz de correlação do sinal de entrada e vetor de correlação cruzada baseados nos sinais desejado e de entrada atuais. O algoritmo LMS com reutilização de dados (DR-LMS) [3] usa os sinais desejado e de entrada atuais repetidamente entre cada iteração de modo a melhorar sua velocidade de convergência. Pode ser mostrado facilmente que, no limite de infinitas reutilizações de dados, os algoritmos DR-LMS e LMS normalizado (NLMS) conduziram à mesma solução [5]. Com os algoritmos denominados “novos LMS normalizado e não-normalizado com reutilização de dados” (normalized e un-normalized new data-reusing LMS algorithms ou NNDR-LMS e UNDR-LMS) [10], recentemente propostos, o desempenho pode ser melhorado quando dados de iterações prévias são também usados.

Em [10], uma descrição gráfica dos algoritmos NNDR-LMS e UNDR-LMS é apresentada e foi mostrado que esta nova classe de algoritmos tem uma perspectiva de desempenho melhor que o algoritmo NLMS em termos de velocidade de convergência. A descrição gráfica também esclareceu porque a melhora é obtida quando o número de reusos é aumentado.

Para o algoritmo LMS, o vetor de coeficientes \mathbf{w} é atualizado na direção oposta ao vetor gradiente ($\nabla_w[\cdot]$) obtido a partir do erro de saída quadrático instantâneo

[2, 11], i.e.,

$$\mathbf{w}_{LMS}(k+1) = \mathbf{w}_{LMS}(k) - \frac{\mu}{2} \nabla_w [e^2(k)] \quad (1.6)$$

onde

$$e(k) = d(k) - \mathbf{x}^T(k) \mathbf{w}_{LMS}(k) \quad (1.7)$$

é o erro de saída, $d(k)$ é o sinal desejado, $\mathbf{x}(k)$ é o vetor sinal de entrada contendo as $N + 1$ amostras mais recentes do sinal de entrada, i.e.,

$$\mathbf{x}(k) = [x(k) \ x(k-1) \ \cdots \ x(k-N)]^T \quad (1.8)$$

e μ é o fator de convergência. A equação de atualização dos coeficientes é

$$\mathbf{w}_{LMS}(k+1) = \mathbf{w}_{LMS}(k) + \mu e(k) \mathbf{x}(k) \quad (1.9)$$

O algoritmo NLMS normaliza o tamanho do passo (step-size) tal que a relação

$$\mathbf{x}^T(k) \mathbf{w}_{NLMS}(k+1) = d(k) \quad (1.10)$$

é sempre satisfeita, i.e.,

$$\mathbf{w}_{NLMS}(k+1) = \mathbf{w}_{NLMS}(k) + \frac{e(k)}{\|\mathbf{x}(k)\|^2 + \epsilon} \mathbf{x}(k) \quad (1.11)$$

onde ϵ , teoricamente igual a zero para satisfazer (1.10), é feito na prática um número muito pequeno usado para evitar a ocorrência de divisão por zero.

Para o algoritmo DR-LMS com L reutilizações de dados, os coeficientes são atualizados por

$$\mathbf{w}_{i+1}(k) = \mathbf{w}_i(k) + \mu e_i(k) \mathbf{x}(k) \quad (1.12)$$

para $i = 0, \dots, L$, onde

$$e_i(k) = d(k) - \mathbf{x}^T(k) \mathbf{w}_i(k) \quad (1.13)$$

$$\mathbf{w}_0(k) = \mathbf{w}_{DR-LMS}(k) \quad (1.14)$$

e

$$\mathbf{w}_{DR-LMS}(k+1) = \mathbf{w}_{L+1}(k) \quad (1.15)$$

Observe que se $L = 0$ estas equações correspondem ao algoritmo LMS convencional.

Para o algoritmo NNDR-LMS com L reutilizações de dados, os coeficientes são atualizados por

$$\mathbf{w}_{i+1}(k) = \mathbf{w}_i(k) + \frac{e_i(k)}{\|\mathbf{x}(k-i)\|^2 + \epsilon} \mathbf{x}(k-i) \quad (1.16)$$

para $i = 0, \dots, L$, onde

$$e_i(k) = d(k-i) - \mathbf{x}^T(k-i)\mathbf{w}_i(k) \quad (1.17)$$

$$\mathbf{w}_0(k) = \mathbf{w}_{NNDR-LMS}(k) \quad (1.18)$$

e

$$\mathbf{w}_{NNDR-LMS}(k+1) = \mathbf{w}_{L+1}(k) \quad (1.19)$$

A Figura 1.2 ilustra geometricamente a atualização do vetor de coeficientes num problema bidimensional para todos os algoritmos discutidos anteriormente, começando com um $\mathbf{w}(k)$ arbitrário. Uma vez que estamos interessados em comparar algoritmos de complexidade similar, foi considerado o caso de uma única reutilização, i.e., $L = 1$. $\mathcal{S}(k)$ denota o hiperplano que contém todos vetores \mathbf{w} tal que $\mathbf{x}^T(k)\mathbf{w} = d(k)$. Numa situação de modelagem exata e livre de ruído, $\mathcal{S}(k)$ conteria o vetor de coeficientes ótimo, \mathbf{w}_o . Pode ser facilmente mostrado que $\mathbf{x}(k)$ e, conseqüentemente, $\nabla_w[e^2(k)]$ são ortogonais ao hiperplano $\mathcal{S}(k)$.

O algoritmo LMS convencional toma um passo simples na direção de $\mathcal{S}(k)$ conduzindo à solução $\mathbf{w}_{LMS}(k+1)$, representada pelo ponto 2 na Figura 1.2, que é mais próximo de $\mathcal{S}(k)$ do que $\mathbf{w}_{LMS}(k)$. O algoritmo DR-LMS se aproxima iterativamente de $\mathcal{S}(k)$ tomando passos sucessivos na direção dada por $\mathbf{x}(k)$. A solução $\mathbf{w}_{DR-LMS}(k+1)$ é representada pelos pontos 2 e 3 na Figura 1.2. Pode ser mostrado que $\mathbf{w}_{DR-LMS}(k+1)$ alcançaria $\mathcal{S}(k)$ no limite onde o número de reutilizações de

dados fosse infinito [10]. O algoritmo NLMS faz uma busca em linha na direção de $\mathbf{x}(k)$ para produzir a solução $\mathbf{w}_{NLMS}(k+1)$ num único passo, representado pelo ponto 4 na Figura 1.2, o qual pertence a $\mathcal{S}(k)$.

Os algoritmos apresentados em [10] usam mais que um hiperplano, i.e., usam pares de dados prévios $(\mathbf{x}(k-i), d(k-i))$, $i > 0$, de modo a produzir as soluções $\mathbf{w}_{UNDR-LMS}(k+1)$ e $\mathbf{w}_{NNDR-LMS}(k+1)$ que são mais próximas de \mathbf{w}_o que as soluções obtidas somente com o par de dados atual $(\mathbf{x}(k), d(k))$. As soluções obtidas com os algoritmos UNDR-LMS e NNDR-LMS são representadas pelos pontos 5 e 6 na Figura 1.2, respectivamente. A posição 7 da Figura 1.2 corresponde ao novo algoritmo LMS binormalizado com reutilização de dados ou “Binormalized Data-Reusing LMS” (BNDR-LMS) o qual será derivado e analisado no próximo capítulo.

Para uma situação de modelagem com ordem exata e sem ruído, \mathbf{w}_o estaria na interseção dos $N + 1$ hiperplanos construídos com vetores de sinal de entrada linearmente independentes. Neste caso, o *algoritmo das projeções ortogonais* [12] produziria a solução ótima \mathbf{w}_o em $N + 1$ iterações. Este algoritmo pode ser visto como um algoritmo ortogonal normalizado que reutiliza $N + 1$ pares de dados (\mathbf{x}, d) , pois ele executa uma busca em linha exata em $(N + 1)$ direções ortogonais construídas de pares de dados atual e passados.

1.4 Introduzindo a Decomposição QR

Um outro método alternativo para a implementação dos mínimos quadrados recursivos (RLS) é o uso da decomposição QR. Em (1.5), a matriz que necessita ser invertida é usualmente chamada de matriz de correlação de dados determinística $\mathbf{R}_D = \mathbf{X}^T(k)\mathbf{X}(k)$ onde $\mathbf{X}(k)$ é a matriz de entrada de dados como será definida mais tarde. A idéia básica desta dita família de algoritmos QR é a triangularização da matriz de entrada de dados por meio do uso de técnicas de decomposição QR.

Vale a pena mencionarmos que a matriz $\mathbf{X}(k)$ é $(k+1) \times (N+1)$ o que significa que a mesma cresce de ordem com o número de iterações. O processo de decomposição QR faz uso da matriz ortonormal $\mathbf{Q}(k)$ de ordem $(k+1) \times (k+1)$ de modo a reduzir

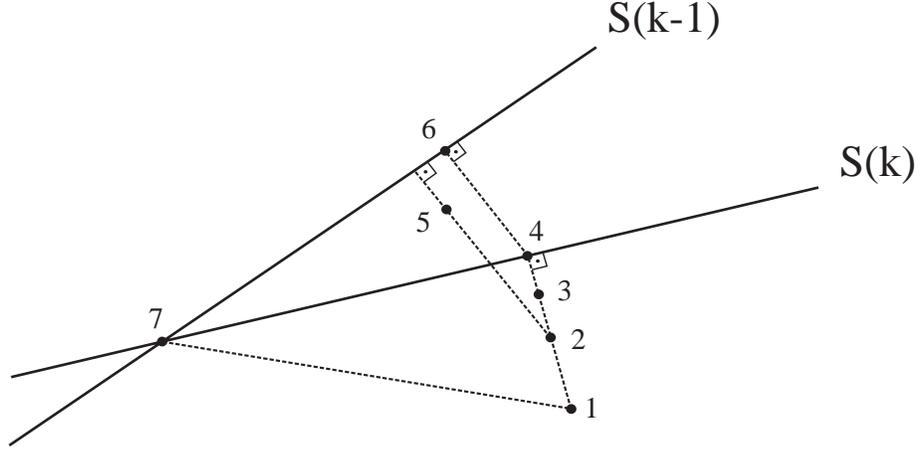


Figura 1.2: Atualização do vetor de coeficientes:

- Posição 1. $\mathbf{w}(k)$;
- Posição 2. $\mathbf{w}_{LMS}(k+1)$ e primeiro passo em direção a $\mathbf{w}_{DR-LMS}(k+1)$ e $\mathbf{w}_{UNDR-LMS}(k+1)$;
- Posição 3. $\mathbf{w}_{DR-LMS}(k+1)$;
- Posição 4. $\mathbf{w}_{NLMS}(k+1)$ e primeiro passo em direção a $\mathbf{w}_{NNDR-LMS}(k+1)$;
- Posição 5. $\mathbf{w}_{UNDR-LMS}(k+1)$;
- Posição 6. $\mathbf{w}_{NNDR-LMS}(k+1)$;
- Posição 7. $\mathbf{w}_{BNDR-LMS}(k+1)$.

$\mathbf{X}(k)$ a uma matriz triangular $\mathbf{U}(k)$ de ordem $(N+1) \times (N+1)$ tal que

$$\mathbf{Q}(k)\mathbf{X}(k) = \begin{bmatrix} \mathbf{O} \\ \mathbf{U}(k) \end{bmatrix} \quad (1.20)$$

onde \mathbf{O} é a matriz de ordem $(k-N) \times (N+1)$ com todos seus elementos nulos.

A matriz $\mathbf{Q}(k)$ representa o processo de triangularização completo e pode ser implementada de diferentes maneiras. Nesta tese, as rotações de Givens serão usadas por apresentarem bom condicionamento numérico embora outras técnicas, tal como a transformação de Householder [13, 14], são disponíveis. As principais vantagens obtidas com os algoritmos RLS que usam a decomposição QR (QRD-RLS) são a possibilidade de implementação em arrays sistólicos e seus melhores comportamentos numéricos em ambiente de precisão finita. O algoritmo QR-RLS convencional tem uma exigência de carga computacional da ordem de N^2 multiplicações ou $O[N^2]$. Um número de algoritmos alternativos tais como o algoritmo QR inverso [15], os

chamados algoritmos QR rápidos ou “fast¹ QR algorithms” (FQR) [16]–[20] e os algoritmos QR rápidos em treliça ou “fast QR-Lattice algorithms” (FQR-L) [21, 22] são também disponíveis na literatura técnica.

Os algoritmos baseados em RLS estão resumidos na Figura 1.3. Estes algoritmos podem ser utilizados quando uma convergência rápida é necessária para sinais de entrada com elevado espalhamento de autovalores e o acréscimo na carga computacional é tolerável. Exceto para o algoritmo FTRLs, cuja versão básica é instável, todos os outros algoritmos RLS rápidos não possuem o vetor de coeficientes na realização em forma direta disponível em toda iteração.

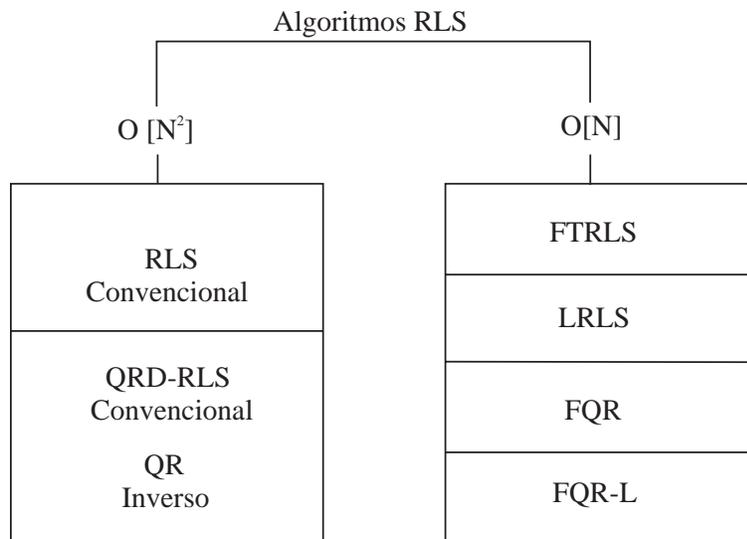


Figura 1.3: Os algoritmos RLS.

O nosso objetivo aqui é o estudo dos algoritmos RLS rápidos baseados em decomposição QR os quais estão entre aqueles algoritmos adaptativos com ambas características de complexidade computacional de $O[N]$ e robustez numérica associada ao uso de rotações de Givens. Embora versões multicanais e complexas da maioria dos algoritmos adaptativos existam [1], esta tese restringir-se-á somente aos casos de um só canal e sinal de entrada real.

¹“Fast” aqui significa de $O[N]$

1.5 Contribuições Originais

Muitos dos algoritmos apresentados e discutidos nas seções anteriores ainda são objetos de pesquisa onde simplicidade, rapidez de convergência, estabilidade e robustez são os tópicos de maior interesse.

No próximo capítulo, o novo algoritmo BNDR-LMS será descrito. Este algoritmo combina reutilização de dados, projeções ortogonais de duas direções de gradientes e normalização de modo a obter convergência mais rápida que outros algoritmos do tipo LMS. Por outro lado, o novo algoritmo é mais simples com relação a complexidade computacional e mais robusto que o algoritmo das projeções ortogonais. A cada iteração, o algoritmo BNDR-LMS produz a solução $\mathbf{w}(k+1)$ a qual está na interseção dos hiperplanos $\mathcal{S}(k)$ e $\mathcal{S}(k-1)$, e a uma distância mínima de $\mathbf{w}(k)$ (veja 7 na Figura 1.2). O algoritmo pode também ser visto como uma versão simplificada do algoritmo das projeções ortogonais que utiliza somente duas direções consecutivas. O Capítulo 2 trata também da análise dos primeiro e segundo momentos do vetor de coeficientes e de resultados de simulações.

No Capítulo 3, uma seqüência ótima de passo é proposta e uma aplicação deste algoritmo no campo das comunicações móveis é desenvolvida.

O desenvolvimento do novo algoritmo QR rápido é feito no Capítulo 4 onde uma abordagem unificada é usada para classificar os membros desta família de algoritmos QR rápidos.

No Capítulo 5, uma versão em treliça de dois daqueles algoritmos QR rápidos é totalmente descrita de acordo com a notação usada neste trabalho.

O Capítulo 6 trata da análise em precisão finita dos algoritmos QR rápidos usando atualização de erros de predição retrógrados (“backward prediction errors”).

As conclusões desta tese bem como sugestões para pesquisas posteriores estão resumidas no Capítulo 7.

Capítulo 2

O Algoritmo BNDR-LMS

O algoritmo LMS binormalizado com reutilização de dados (BNDR-LMS) [23] descrito neste capítulo e brevemente introduzido em [24] e [25] emprega normalização em duas direções ortogonais obtidas de pares de dados consecutivos entre cada iteração. Em todas simulações realizadas com sinais de entrada coloridos, este algoritmo apresentou convergência mais rápida que outros algoritmos com reutilização de dados para o caso de dois pares de dados, ou, equivalentemente, uma reutilização de dados.

Uma minuciosa análise de convergência na média e média quadrática do vetor de coeficientes é apresentada e os limites de estabilidade para o fator de convergência bem como fórmulas fechadas para o erro médio quadrático (MSE) após convergência são obtidos desta análise. Sendo o pressuposto de independência (termo mais conhecido em inglês por “independence assumption”) [26] inadequado para análises de algoritmos que reutilizam dados [10], é adotado um modelo simplificado para o vetor sinal de entrada o qual é consistente com os primeiro e segundo momentos e proporciona uma análise tratável [4, 27]. Esta análise pode ser estendida a outros algoritmos que reutilizam dados tais como os algoritmos NNDR-LMS e UNDR-LMS [10].

2.1 Definição do Problema e Derivação do Algoritmo

Visando uma boa definição do problema, notamos que a solução que pertence a $\mathcal{S}(k)$ e $\mathcal{S}(k-1)$ a uma distância mínima de $\mathbf{w}(k)$ é aquela que resolve

$$\min_{\mathbf{w}(k+1)} \|\mathbf{w}(k+1) - \mathbf{w}(k)\|^2 \quad (2.1)$$

sujeito a

$$\mathbf{x}^T(k)\mathbf{w}(k+1) = d(k) \quad (2.2)$$

e

$$\mathbf{x}^T(k-1)\mathbf{w}(k+1) = d(k-1) \quad (2.3)$$

A função a ser minimizada é, portanto,

$$\begin{aligned} f[\mathbf{w}(k+1)] = & \|\mathbf{w}(k+1) - \mathbf{w}(k)\|^2 + \lambda_1[d(k) - \mathbf{x}^T(k)\mathbf{w}(k+1)] \\ & + \lambda_2[d(k-1) - \mathbf{x}^T(k-1)\mathbf{w}(k+1)] \end{aligned} \quad (2.4)$$

a qual, para vetores de sinal de entrada linearmente independentes $\mathbf{x}(k)$ e $\mathbf{x}(k-1)$, tem a solução única

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \frac{\lambda_1}{2}\mathbf{x}(k) + \frac{\lambda_2}{2}\mathbf{x}(k-1) \quad (2.5)$$

onde

$$\frac{\lambda_1}{2} = \frac{[d(k) - \mathbf{x}^T(k)\mathbf{w}(k)]\|\mathbf{x}(k-1)\|^2 - [d(k-1) - \mathbf{x}^T(k-1)\mathbf{w}(k)]\mathbf{x}^T(k-1)\mathbf{x}(k)}{\|\mathbf{x}(k)\|^2\|\mathbf{x}(k-1)\|^2 - [\mathbf{x}^T(k)\mathbf{x}(k-1)]^2} \quad (2.6)$$

e

$$\frac{\lambda_2}{2} = \frac{[d(k-1) - \mathbf{x}^T(k-1)\mathbf{w}(k)]\|\mathbf{x}(k)\|^2 - [d(k) - \mathbf{x}^T(k)\mathbf{w}(k)]\mathbf{x}^T(k-1)\mathbf{x}(k)}{\|\mathbf{x}(k)\|^2\|\mathbf{x}(k-1)\|^2 - [\mathbf{x}^T(k)\mathbf{x}(k-1)]^2} \quad (2.7)$$

2.1.1 Versão Simplificada

A derivação apresentada acima é válida para qualquer $\mathbf{w}(k)$, o qual pode ou não pertencer a $\mathcal{S}(k-1)$. Entretanto, se passos de otimização sucessivos são tomados para $\mathbf{w}(k)$ para todo k , então, de (2.2), temos

$$\mathbf{x}^T(k-1)\mathbf{w}(k) = d(k-1) \quad (2.8)$$

e um conjunto simplificado de equações de atualização para o algoritmo resulta:

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \frac{\lambda'_1}{2}\mathbf{x}(k) + \frac{\lambda'_2}{2}\mathbf{x}(k-1) \quad (2.9)$$

onde

$$\frac{\lambda'_1}{2} = \frac{[d(k) - \mathbf{x}^T(k)\mathbf{w}(k)]\|\mathbf{x}(k-1)\|^2}{\|\mathbf{x}(k)\|^2\|\mathbf{x}(k-1)\|^2 - [\mathbf{x}^T(k)\mathbf{x}(k-1)]^2} \quad (2.10)$$

e

$$\frac{\lambda'_2}{2} = \frac{-[d(k) - \mathbf{x}^T(k)\mathbf{w}(k)]\mathbf{x}^T(k-1)\mathbf{x}(k)}{\|\mathbf{x}(k)\|^2\|\mathbf{x}(k-1)\|^2 - [\mathbf{x}^T(k)\mathbf{x}(k-1)]^2} \quad (2.11)$$

2.1.2 Derivação Geométrica

O algoritmo BNDR-LMS pode ser alternativamente derivado a partir de um raciocínio puramente geométrico. O primeiro passo é obter uma solução preliminar, $\mathbf{w}_1(k)$, a qual pertence a $\mathcal{S}(k)$ e está a uma distância mínima de $\mathbf{w}(k)$, representada pelo ponto 4 na Figura 1.2. Isto é obtido pelo algoritmo NLMS partindo de $\mathbf{w}(k)$, i.e.,

$$\mathbf{w}_1(k) = \mathbf{w}(k) + \frac{e(k)}{\|\mathbf{x}(k)\|^2}\mathbf{x}(k) \quad (2.12)$$

Num segundo passo, $\mathbf{w}_1(k)$ é atualizado numa direção ortogonal à usada previamente, pertencendo portanto a $\mathcal{S}(k)$, até a interseção com $\mathcal{S}(k-1)$ ser alcançada. Isto é obtido pelo algoritmo NLMS partindo de $\mathbf{w}_1(k)$ e seguindo a direção de $\mathbf{x}_1^\perp(k)$ o qual é a projeção de $\mathbf{x}(k-1)$ em $\mathcal{S}(k)$, i.e.,

$$\mathbf{w}(k+1) = \mathbf{w}_1(k) + \frac{e_1(k)}{\|\mathbf{x}_1^{\perp T}(k)\|^2}\mathbf{x}_1^\perp(k) \quad (2.13)$$

onde

$$\mathbf{x}_1^\perp(k) = \left[\mathbf{I} - \frac{\mathbf{x}(k)\mathbf{x}^T(k)}{\|\mathbf{x}(k)\|^2} \right] \mathbf{x}(k-1) \quad (2.14)$$

e

$$e_1(k) = d(k-1) - \mathbf{x}^T(k-1)\mathbf{w}_1(k) \quad (2.15)$$

O uso de $\mathbf{x}_1^\perp(k)$ obtido de $\mathbf{x}(k-1)$ garante que o caminho da distância mínima seja escolhido. Note que o requisito de independência linear dos vetores sinal de entrada consecutivos $\mathbf{x}(k)$ e $\mathbf{x}(k-1)$, necessário para garantir a existência da solução, é também manifestado aqui.

Como será visto na análise (c.f. Seção 2.3) e será verificado por simulações (c.f. Tabela 2.2), o excesso de erro médio quadrático (MSE) para o algoritmo BNDR-LMS tal como em (2.5)–(2.7) ou em (2.9)–(2.11) é próximo à variância do ruído de observação quando não existe erro de modelagem. Tal desempenho é esperado dos algoritmos normalizados. Portanto, de modo a controlar este excesso de MSE, um passo (fator de convergência) μ pode ser introduzido. Embora a taxa de convergência máxima é usualmente obtida com $\mu = 1$, o uso de um passo menor pode ser necessário em aplicações onde o erro de medida é muito elevado. Neste caso, devemos enfatizar que a solução $\mathbf{w}(k+1)$ obtida a cada iteração não está na interseção dos hiperplanos $\mathcal{S}(k-1)$ e $\mathcal{S}(k)$ e, portanto a versão simplificada do algoritmo dada por (2.9)–(2.11) não deve ser usada.

Se $\mathbf{x}(k)$ e $\mathbf{x}(k-1)$ forem linearmente dependentes, então $\mathcal{S}(k)$ é paralelo a $\mathcal{S}(k-1)$, $\mathbf{x}_1^\perp(k)$ é o vetor nulo e $\mathbf{w}(k+1) = \mathbf{w}_1(k)$, que corresponde ao algoritmo NLMS para qualquer valor de passo. Particularmente quando $\mu = 1$, é correto dizer que $\mathbf{w}(k)$ já está no hiperplano $\mathcal{S}(k-1)$.

O algoritmo BNDR-LMS está resumido na Tabela 2.1.

Tabela 2.1: O algoritmo “Binormalized Data-Reusing LMS”

BNDR-LMS
$\epsilon = \text{valor positivo e pequeno}$ for $k = 1, 2, \dots$ { $\alpha = \mathbf{x}^T(k)\mathbf{x}(k-1)$ $\rho(k) = \mathbf{x}^T(k)\mathbf{x}(k)$ $y_1 = \mathbf{x}^T(k)\mathbf{w}(k)$ $e_1 = d(k) - y_1$ $den = \rho(k)\rho(k-1) - \alpha^2$ if $den < \epsilon$ { $\mathbf{w}(k+1) = \mathbf{w}(k) + \mu e_1 \mathbf{x}(k) / \rho(k)$ } else { $y_2 = \mathbf{x}^T(k-1)\mathbf{w}(k)$ $e_2 = d(k-1) - y_2$ $\frac{\lambda_1}{2} = (e_1\rho(k-1) - e_2\alpha) / den$ $\frac{\lambda_2}{2} = (e_2\rho(k) - e_1\alpha) / den$ $\mathbf{w}(k+1) = \mathbf{w}(k) + \mu[\frac{\lambda_1}{2}\mathbf{x}(k) + \frac{\lambda_2}{2}\mathbf{x}(k-1)]$ } } }

2.2 Análise de Convergência do Vetor de Coeficientes

Nesta seção, nós assumimos que um filtro FIR desconhecido com vetor de coeficientes dado por \mathbf{w}_o está para ser identificado por um filtro adaptativo da mesma ordem empregando o algoritmo BNDR-LMS, i.e., $d(k)$ pode ser modelado como

$$d(k) = \mathbf{x}^T(k)\mathbf{w}_o + n(k) \tag{2.16}$$

onde $n(k)$ é o ruído de medição. É também assumido que o sinal de entrada e o ruído de medição são tomados de processos brancos, de média zero, identicamente distribuídos e independentes com variâncias σ_x^2 e σ_n^2 , respectivamente.

Estamos interessados em analisar o comportamento do vetor de coeficientes du-

rante a convergência em termos de um valor de passo μ . Seja

$$\Delta \mathbf{w}(k) = \mathbf{w}(k) - \mathbf{w}_o \quad (2.17)$$

o vetor erro nos coeficientes do filtro adaptativo em relação ao vetor de coeficientes ideal. Para o algoritmo BNDR-LMS assim como descrito em (2.5)–(2.7), $\Delta \mathbf{w}(k+1)$ é dado por

$$\Delta \mathbf{w}(k+1) = \Delta \mathbf{w}(k) + \mu \left[\frac{\lambda_1}{2} \mathbf{x}(k) + \frac{\lambda_2}{2} \mathbf{x}(k-1) \right] \quad (2.18)$$

De (2.16) e (2.5)–(2.7), temos

$$\Delta \mathbf{w}(k+1) = [\mathbf{I} + \mu \mathbf{A}] \Delta \mathbf{w}(k) + \mu \mathbf{b} \quad (2.19)$$

onde

$$\begin{aligned} \mathbf{A} = & \frac{\mathbf{x}(k) \mathbf{x}^T(k) \mathbf{x}(k-1) \mathbf{x}^T(k-1) + \mathbf{x}(k-1) \mathbf{x}^T(k-1) \mathbf{x}(k) \mathbf{x}^T(k)}{\|\mathbf{x}(k)\|^2 \|\mathbf{x}(k-1)\|^2 - [\mathbf{x}^T(k) \mathbf{x}(k-1)]^2} \\ & - \frac{\|\mathbf{x}(k-1)\|^2 \mathbf{x}(k) \mathbf{x}^T(k) + \|\mathbf{x}(k)\|^2 \mathbf{x}(k-1) \mathbf{x}^T(k-1)}{\|\mathbf{x}(k)\|^2 \|\mathbf{x}(k-1)\|^2 - [\mathbf{x}^T(k) \mathbf{x}(k-1)]^2} \end{aligned} \quad (2.20)$$

e

$$\begin{aligned} \mathbf{b} = & \frac{n(k) \|\mathbf{x}(k-1)\|^2 - n(k-1) \mathbf{x}^T(k) \mathbf{x}(k-1)}{\|\mathbf{x}(k)\|^2 \|\mathbf{x}(k-1)\|^2 - [\mathbf{x}^T(k) \mathbf{x}(k-1)]^2} \mathbf{x}(k) \\ & + \frac{n(k-1) \|\mathbf{x}(k)\|^2 - n(k) \mathbf{x}^T(k-1) \mathbf{x}(k)}{\|\mathbf{x}(k)\|^2 \|\mathbf{x}(k-1)\|^2 - [\mathbf{x}^T(k) \mathbf{x}(k-1)]^2} \mathbf{x}(k-1) \end{aligned} \quad (2.21)$$

Tomando-se o valor esperado de ambos lados de (2.19), para $n(k)$ e $x(k)$ amostras de processos aleatórios independentes e de média nula, temos

$$E[\mathbf{b}] = 0 \quad (2.22)$$

e

$$\begin{aligned} E[\Delta \mathbf{w}(k+1)] &= E[(\mathbf{I} + \mu \mathbf{A}) \Delta \mathbf{w}(k)] \\ &= E \left(\left\{ \mathbf{I} + \mu \left[\frac{\mathbf{x}(k) \mathbf{x}^T(k) \mathbf{x}(k-1) \mathbf{x}^T(k-1)}{\|\mathbf{x}(k)\|^2 \|\mathbf{x}(k-1)\|^2 - [\mathbf{x}^T(k) \mathbf{x}(k-1)]^2} \right. \right. \right. \\ &\quad + \frac{\mathbf{x}(k-1) \mathbf{x}^T(k-1) \mathbf{x}(k) \mathbf{x}^T(k) - \|\mathbf{x}(k-1)\|^2 \mathbf{x}(k) \mathbf{x}^T(k)}{\|\mathbf{x}(k)\|^2 \|\mathbf{x}(k-1)\|^2 - [\mathbf{x}^T(k) \mathbf{x}(k-1)]^2} \\ &\quad \left. \left. \left. - \frac{\|\mathbf{x}(k)\|^2 \mathbf{x}(k-1) \mathbf{x}^T(k-1)}{\|\mathbf{x}(k)\|^2 \|\mathbf{x}(k-1)\|^2 - [\mathbf{x}^T(k) \mathbf{x}(k-1)]^2} \right] \right\} \Delta \mathbf{w}(k) \right) \end{aligned} \quad (2.23)$$

A expressão (2.23) pode ser adicionalmente simplificada se os seguintes pressupostos forem feitos:

1. $\Delta \mathbf{w}(k)$ é estatisticamente independente de $\mathbf{x}(k)\mathbf{x}^T(k)$ (“independence assumption” [26]),
2. $E[num/den] \approx E[num]/E[den]$, onde num e den são os elementos no numerador e denominador de (2.23), respectivamente, que implica na independência entre num e den bem como uma aproximação de primeira ordem¹ de $E[1/den]$.

Além disto, as seguintes relações podem ser facilmente verificadas quando os elementos de $\mathbf{x}(k)$ são amostras de um processo branco Gaussiano (veja o Apêndice A):

1.

$$E\{[\mathbf{x}^T(k)\mathbf{x}(k-1)]^2\} = (N+1)(\sigma_x^2)^2 \quad (2.24)$$

2.

$$E\{\|\mathbf{x}(k)\|^2\|\mathbf{x}(k-1)\|^2 - [\mathbf{x}^T(k)\mathbf{x}(k-1)]^2\} = N(N+3)(\sigma_x^2)^2 \quad (2.25)$$

3.

$$\{E[\mathbf{x}(k-1)\mathbf{x}^T(k-1)\mathbf{x}(k)\mathbf{x}^T(k)]\}_{ij} = \begin{cases} (\sigma_x^2)^2, & i = j \text{ or } i = j - 2 \\ 0, & \text{caso contrário} \end{cases} \quad (2.26)$$

para $[\cdot]_{ij}$ o elemento (i, j) da matriz $[\cdot]$.

4.

$$E[\|\mathbf{x}(k-1)\|^2\mathbf{x}(k)\mathbf{x}^T(k)] = (N+3)(\sigma_x^2)^2\mathbf{I} \quad (2.27)$$

5.

$$\mathbf{x}^T(k-1)\Delta \mathbf{w}(k) = (1-\mu)\mathbf{x}^T(k-1)\Delta \mathbf{w}(k-1) + \mu n(k-1) \quad (2.28)$$

¹Para uma discussão em maior profundidade, veja [4, 6].

Com base nestes pressupostos e relações, (2.23) pode ser re-escrita como

$$\begin{aligned}
E[\Delta \mathbf{w}(k+1)] &\approx E \left(\left\{ \mathbf{I} + \mu \left[\frac{\mathbf{x}(k-1)\mathbf{x}^T(k-1)\mathbf{x}(k)\mathbf{x}^T(k)}{N(N+3)(\sigma_x^2)^2} \right. \right. \right. \\
&\quad \left. \left. \left. - \frac{\|\mathbf{x}(k-1)\|^2 \mathbf{x}(k)\mathbf{x}^T(k)}{N(N+3)(\sigma_x^2)^2} \right] \right\} \Delta \mathbf{w}(k) \right. \\
&\quad \left. + \mu(1-\mu) \left[\frac{\mathbf{x}(k)\mathbf{x}^T(k)\mathbf{x}(k-1)\mathbf{x}^T(k-1)}{N(N+3)(\sigma_x^2)^2} \right. \right. \\
&\quad \left. \left. - \frac{\|\mathbf{x}(k)\|^2 \mathbf{x}(k-1)\mathbf{x}^T(k-1)}{N(N+3)(\sigma_x^2)^2} \right] \Delta \mathbf{w}(k-1) \right) \\
&\approx \left(1 - \frac{\mu}{N} \right) E[\Delta \mathbf{w}(k)] - \frac{\mu(1-\mu)}{N} E[\Delta \mathbf{w}(k-1)]
\end{aligned} \tag{2.29}$$

A última relação de (2.29) foi obtida considerando-se $\|\mathbf{x}(k-1)\|^2$ estatisticamente independente de $\Delta \mathbf{w}(k)$ e fazendo-se uma aproximação de primeira ordem no cálculo do numerador com a ajuda das relações (2.26) a (2.28). De (2.29), é claro que a convergência na média do algoritmo BNDR-LMS para uma solução não polarizada é garantida para valores de passo μ tais que todos elementos de $E[\Delta \mathbf{w}(k+1)]$ em (2.29) tendam a zero com $k \rightarrow \infty$. Isto é obtido se os pólos da equação de diferenças de segunda ordem estiverem estritamente dentro do círculo unitário, i.e.,

$$|z_{1,2}| = \left| \frac{1 - \frac{\mu}{N} \pm \sqrt{\left(1 - \frac{\mu}{N}\right)^2 - \frac{4\mu(1-\mu)}{N}}}{2} \right| < 1 \tag{2.30}$$

o que é sempre verdadeiro para $N > 1$ e μ satisfazendo

$$0 < \mu < 2 \tag{2.31}$$

2.3 Análise de Estatística de Segunda Ordem

2.3.1 Sinal de Entrada Branco

Embora $\Delta \mathbf{w}(k)$ venha a convergir em média para zero quando k vai para infinito, fato este que caracteriza uma estimativa não polarizada, a consistência dos coeficientes estimados, ou seja, erros instantâneos desprezíveis nestes coeficientes, somente é obtida em casos de ξ_{min} muito pequeno ou valor de μ próximo de zero. Em geral,

um excesso de MSE, o qual depende das estatísticas de segunda ordem do vetor $\Delta\mathbf{w}(k)$, estará presente. O excesso de MSE é definido por [1, 2]

$$\xi_{exc} = \lim_{k \rightarrow \infty} \xi(k) - \xi_{min} \quad (2.32)$$

onde $\xi(k) = E[e^2(k)]$ e ξ_{min} é o erro médio quadrático mínimo devido a uma modelagem não exata ou à presença de ruído aditivo, ou ambos [2].

A diferença $\Delta\xi(k) = \xi(k) - \xi_{min}$ é conhecida como excesso no MSE [2] e pode ser expressa por

$$\begin{aligned} \Delta\xi(k) &= E\{[n(k) - \Delta\mathbf{w}^T(k)\mathbf{x}(k)]^2\} - \xi_{min} \\ &= E[\Delta\mathbf{w}^T(k)\mathbf{R}\Delta\mathbf{w}(k)] \\ &= \text{tr}\{\mathbf{R}\text{cov}[\Delta\mathbf{w}(k)]\} \end{aligned} \quad (2.33)$$

assumindo $\Delta\mathbf{w}(k)$ independente de $\mathbf{x}(k)$.

É necessário, portanto, derivarmos uma expressão para a matriz de covariância do vetor de erro dos coeficientes $\text{cov}[\Delta\mathbf{w}(k+1)]$. De (2.19),

$$\begin{aligned} \text{cov}[\Delta\mathbf{w}(k+1)] &= E[\Delta\mathbf{w}(k+1)\Delta\mathbf{w}^T(k+1)] \\ &= E\{[\mathbf{I} + \mu\mathbf{A}]\Delta\mathbf{w}(k)\Delta\mathbf{w}^T(k)[\mathbf{I} + \mu\mathbf{A}]\} \\ &\quad + E\{\mu[\mathbf{I} + \mu\mathbf{A}]\Delta\mathbf{w}(k)\mathbf{b}^T\} + E\{\mu\mathbf{b}\Delta\mathbf{w}^T(k)[\mathbf{I} + \mu\mathbf{A}]\} \\ &\quad + E[\mu^2\mathbf{b}\mathbf{b}^T] \end{aligned} \quad (2.34)$$

Lembrando (2.20) e (2.21), podemos antever a enorme complexidade para avaliarmos (2.34) mesmo com um número razoável de pressupostos. Uma alternativa interessante é o uso de um modelo simplificado para o vetor sinal de entrada $\mathbf{x}(k)$ o qual seja consistente com as estatísticas de primeira e segunda ordens de um sinal de entrada genérico, mas que tenha um número contável de direções de excitação. Este modelo foi introduzido em [28] e foi empregado anteriormente com sucesso em [4] e [27]. O vetor sinal de entrada para o modelo é

$$\mathbf{x}(k) = s_k r_k \mathbf{V}_k \quad (2.35)$$

onde:

- s_k é ± 1 com probabilidade de ocorrência igual a $1/2$;
- r_k^2 tem a mesma função de distribuição de probabilidade de $\|\mathbf{x}(k)\|^2$, ou, para o caso de interesse, é uma amostra de um processo independente com distribuição χ -quadrado de $(N + 1)$ graus de liberdade, $E[r_k^2] = (N + 1)\sigma_x^2$;
- \mathbf{V}_k é igual a um dos $N + 1$ autovetores ortonormais de \mathbf{R} , denotados \mathcal{V}_i , $i = 1, \dots, N + 1$. Assumiremos também para um sinal de entrada branco Gaussiano que \mathbf{V}_k é uniformemente distribuído e, conseqüentemente, se $P(\cdot)$ denota a probabilidade de ocorrência do evento (\cdot) , então

$$P(\mathbf{V}_k = \mathcal{V}_i) = \frac{1}{N + 1} \quad (2.36)$$

Para o modelo de sinal de entrada dado, podemos expressar $\Delta\xi(k + 1)$ como

$$\begin{aligned} \Delta\xi(k + 1) &= \Delta\xi(k + 1) |_{\mathbf{x}(k)\|\mathbf{x}(k-1)} \times P[\mathbf{x}(k) \parallel \mathbf{x}(k - 1)] \\ &\quad + \Delta\xi(k + 1) |_{\mathbf{x}(k)\perp\mathbf{x}(k-1)} \times P[\mathbf{x}(k) \perp \mathbf{x}(k - 1)] \end{aligned} \quad (2.37)$$

As condições $\mathbf{x}(k) \parallel \mathbf{x}(k - 1)$ e $\mathbf{x}(k) \perp \mathbf{x}(k - 1)$ no modelo adotado são equivalentes a $\mathbf{V}_k = \mathbf{V}_{k-1}$ e $\mathbf{V}_k \neq \mathbf{V}_{k-1}$, respectivamente, tal que \mathbf{V}_k e \mathbf{V}_{k-1} podem somente ser paralelos ou ortogonais entre si.

Como comentado anteriormente, o algoritmo BNDR-LMS comporta-se exatamente como o algoritmo NLMS quando os vetores sinal de entrada nos instantes k e $k - 1$ são paralelos. Neste caso o excesso de MSE é dado por [4]

$$\Delta\xi(k + 1)_{\parallel} = \left[1 + \frac{\mu(\mu - 2)}{N + 1} \right] \Delta\xi(k) + \frac{\mu^2}{(N + 2 - \nu_x)} \sigma_n^2 \quad (2.38)$$

onde $\nu_x = E[x^4(k)/\sigma_x^4]$ é conhecido como *kurtosis* do sinal de entrada, a qual varia de 1 para uma distribuição binária a 3 para uma distribuição Gaussiana a ∞ para uma distribuição Cauchy [4, 29]. Deve ser ressaltado, entretanto, que (2.38) é válida somente para $\nu_x \ll N + 1$ [4].

Para o caso onde $\mathbf{x}(k)$ e $\mathbf{x}(k - 1)$ são sempre ortogonais, de (2.33) e (2.34) nós temos, para $\mathbf{R} = \sigma_x^2 \mathbf{I}$, i.e., sinais de entrada ruído branco (veja Apêndice B),

$$\begin{aligned} \Delta\xi(k + 1)_{\perp} &= \left[1 + \frac{\mu(\mu - 2)}{N + 1} \right] \Delta\xi(k) + \frac{\mu(1 - \mu)^2(\mu - 2)}{N + 1} \Delta\xi(k - 1) \\ &\quad + \frac{\mu^2(\mu - 2)^2}{N + 2 - \nu_x} \sigma_n^2 \end{aligned} \quad (2.39)$$

Uma expressão final para o excesso no MSE pode agora ser obtida de (2.38) e (2.39) combinadas e corretamente ponderadas, como sugerido em (2.37). Para um sinal de entrada branco, as probabilidades de $\mathbf{V}_k = \mathbf{V}_{k-1}$ e $\mathbf{V}_k \neq \mathbf{V}_{k-1}$ são iguais a $\frac{1}{N+1}$ e $\frac{N}{N+1}$, respectivamente. O excesso no MSE é, portanto, dado por

$$\begin{aligned} \Delta\xi(k+1) = & \left[1 + \frac{\mu(\mu-2)}{N+1}\right] \Delta\xi(k) + \frac{N\mu(1-\mu)^2(\mu-2)}{(N+1)^2} \Delta\xi(k-1) \\ & + \frac{\mu^2 [1 + N(\mu-2)^2]}{(N+1)(N+2-\nu_x)} \sigma_n^2 \end{aligned} \quad (2.40)$$

Experimento 1: De modo a confrontar o comportamento do algoritmo BNDR-LMS com (2.38) para diferentes valores de μ um experimento simples foi conduzido onde os vetores sinal de entrada em instantes consecutivos de tempo são sempre paralelos. O esquema foi construído de modo que uma planta desconhecida de décima ordem deve ser identificada por um filtro adaptativo também de ordem $N = 10$ para um vetor sinal de entrada $\mathbf{x}(k) = s_k \mathbf{V}$ com \mathbf{V} um vetor constante igual a $[1 \ 0 \ \dots \ 0]^T$. Neste caso, a *kurtosis* ν_x é igual a 1 e o valor em regime estacionário de $\Delta\xi(k)$, ξ_{exc} , é

$$\xi_{exc} = \frac{\mu\sigma_n^2}{2-\mu} \quad (2.41)$$

Os resultados, mostrados na Figura 2.1, mostram uma comparação entre simulações tomadas em média de 50 rodadas e valores teóricos previstos por (2.41). Da análise deste experimento, fica claro que, para o caso especial onde o algoritmo BNDR-LMS comporta-se como o algoritmo NLMS, (2.41) está em excelente concordância com os resultados das simulações.

Experimento 2: Um segundo experimento foi conduzido onde vetores sinal de entrada em instantes de tempo consecutivos são sempre ortogonais. O esquema foi similar ao do Experimento 1, exceto pelo vetor sinal de entrada que foi escolhido como $\mathbf{x}(k) = s_k \mathbf{V}_k$ com \mathbf{V}_k sempre diferente de \mathbf{V}_{k-1} e igual a um dos vetores que formam a base canônica. A *kurtosis* do sinal de entrada é também igual a 1. Os resultados teóricos e os resultados obtidos numa simulação com 50 rodadas estão

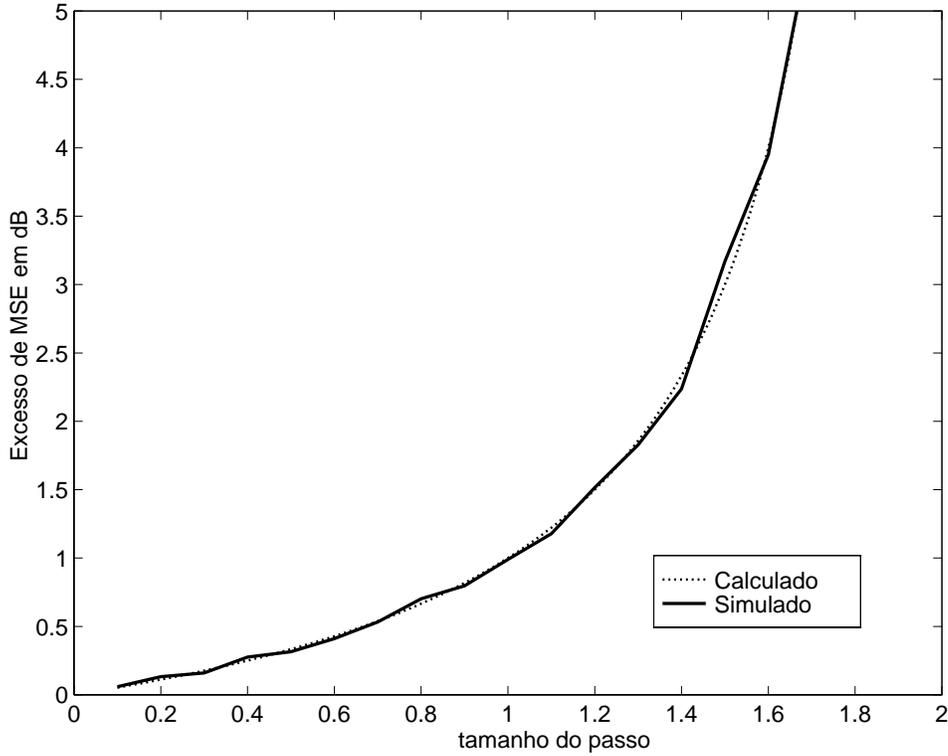


Figura 2.1: Excesso de MSE para vetores sinal de entrada paralelos.

apresentados na Figura 2.2. Tais resultados mostram que a expressão em (2.39) é precisa para os pressupostos feitos.

Experimento 3: Um terceiro experimento foi conduzido onde os vetores sinal de entrada foram escolhidos aleatoriamente entre os componentes da base canônica, tal que \mathbf{V}_k e \mathbf{V}_{k-1} podem ser paralelos ou ortogonais com probabilidades $\frac{1}{N+1}$ e $\frac{N}{N+1}$, respectivamente. Os resultados de uma simulação com 50 rodadas e aqueles de (2.40) são descritos na Figura 2.3 que mostra a precisão da análise ao menos para o modelo de sinal de entrada usado.

2.3.2 Sinal de Entrada Colorido

Usando o modelo para o vetor sinal de entrada dado em (2.35), podemos agora estender a análise para sinais de entrada coloridos. A distribuição angular de $\mathbf{x}(k)$

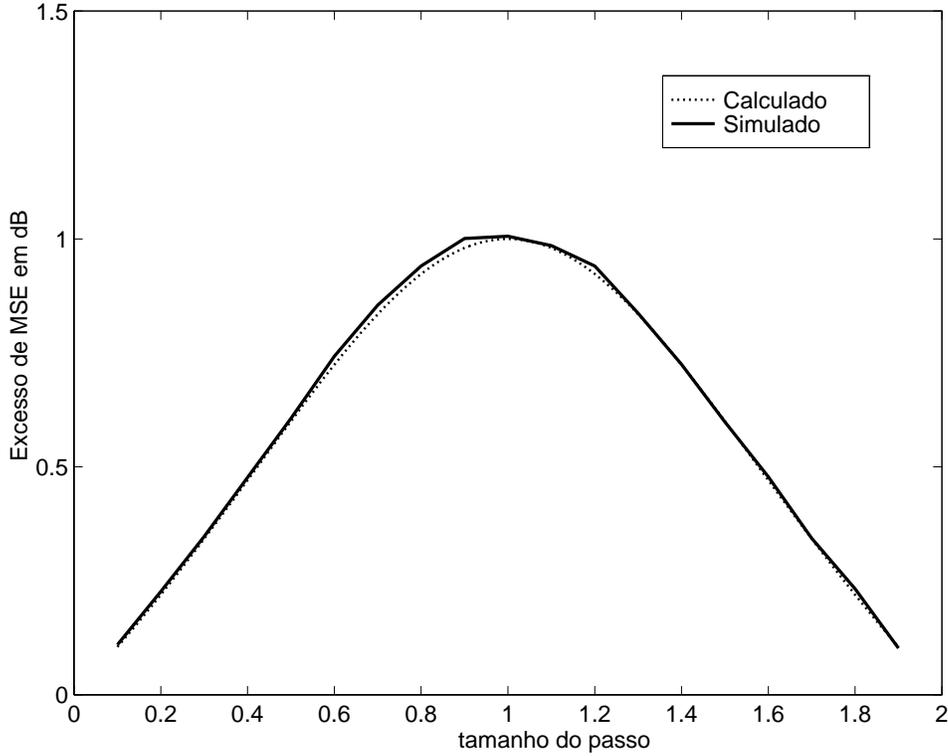


Figura 2.2: Excesso de MSE para vetores sinal de entrada ortogonais.

necessita ser mudada de modo a incorporar probabilidades diferentes para as direções dadas pelos $(N + 1)$ autovetores de \mathbf{R} . Em outras palavras, (2.37)–(2.39) são mantidas e somente as probabilidades $P[\mathbf{x}(k) \parallel \mathbf{x}(k - 1)]$ e $P[\mathbf{x}(k) \perp \mathbf{x}(k - 1)]$ necessitam ser recalculadas. Cada autovetor de \mathbf{R} , denotado por \mathcal{V}_i , $i = 1, \dots, N + 1$ terá agora a seguinte probabilidade de ocorrência [4]

$$P(\mathbf{V}_k = \mathcal{V}_i) = \frac{\lambda_i}{tr(\mathbf{R})} \quad (2.42)$$

onde λ_i é o autovalor associado ao autovetor \mathcal{V}_i . Para uma simples associação entre $P[\mathbf{x}(k) \parallel \mathbf{x}(k - 1)]$ e a correlação do sinal de entrada, suponhamos que o sinal de entrada $x(k)$ é correlacionado por um filtro só pólos como em

$$x(k) = \gamma x(k - 1) + (1 - \gamma)\eta(k), \quad 0 \leq \gamma \leq 1 \quad (2.43)$$

onde $\eta(k)$ é uma amostra de um processo de média zero independente com variância dada por σ_η^2 . A matriz de autocorrelação para este sinal de entrada pode ser facil-

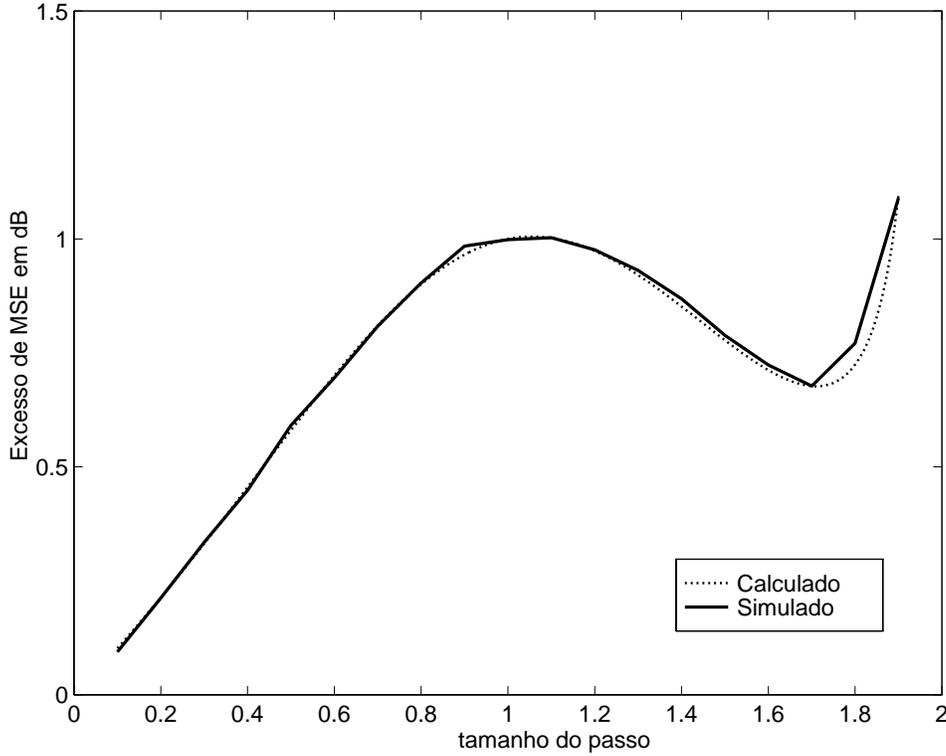


Figura 2.3: Excesso de MSE para um vetor sinal de entrada modelado.

mente derivada e expressa por como

$$\mathbf{R} = \frac{1 - \gamma}{1 + \gamma} \sigma_{\eta}^2 \begin{bmatrix} 1 & \gamma & \gamma^2 & \dots & \gamma^N \\ \gamma & 1 & \gamma & \dots & \gamma^{N-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \gamma^N & \gamma^{N-1} & \gamma^{N-2} & \dots & 1 \end{bmatrix} \quad (2.44)$$

De (2.44) temos todos os autovalores e autovetores necessários tal que podemos computar

$$\begin{aligned} P[\mathbf{x}(k) \parallel \mathbf{x}(k-1)] &= P[\mathbf{V}_k \parallel \mathbf{V}_{k-1}] \\ &= P[\mathbf{V}_k \parallel \mathbf{V}_{k-1} \mid \mathbf{V}_{k-1} = \mathbf{V}_1] \times P[\mathbf{V}_{k-1} = \mathbf{V}_1] + \dots \\ &\quad + P[\mathbf{V}_k \parallel \mathbf{V}_{k-1} \mid \mathbf{V}_{k-1} = \mathbf{V}_{N+1}] \times P[\mathbf{V}_{k-1} = \mathbf{V}_{N+1}] \\ &= \sum_{i=1}^{N+1} \left(\frac{\lambda_i}{\text{tr}(\mathbf{R})} \right)^2 \end{aligned} \quad (2.45)$$

e

$$P[\mathbf{x}(k) \perp \mathbf{x}(k-1)] = 1 - P[\mathbf{x}(k) \parallel \mathbf{x}(k-1)] \quad (2.46)$$

As equações (2.45) e (2.46) estão de acordo com a situação de termos entrada branca, pois este caso corresponde a $\gamma = 0$ e todos autovalores iguais a σ_x^2 tal que $P(\mathbf{V}_k = \mathcal{V}_i) = \frac{1}{N+1}$ como já anteriormente descrito. Quando o sinal de entrada é correlacionado por um filtro só pólos de primeira ordem e modelado com (2.35) e (2.42), o excesso de MSE é dado por (2.37)–(2.39) com probabilidades dadas por (2.45) e (2.46). Embora (2.38)–(2.39) tenham sido obtidas baseado num modelo branco Gaussiano para o sinal de entrada, as simulações mostraram que nosso raciocínio é válido quando o sinal de entrada é gerado de acordo com (2.35) com probabilidades dadas por (2.42) e λ_i obtido de (2.44). Além disto, para $\mu = 1$ e um sinal de entrada modelado onde somente vetores sinal de entrada paralelos ou perpendiculares podem ocorrer, o algoritmo BNDR-LMS degrada para o algoritmo NLMS e o MSE em estado estacionário torna-se independente da distribuição radial de $\mathbf{x}(k)$ [4]. Isto é perfeitamente descrito por (2.38)–(2.39), suportando a validade deste raciocínio.

2.4 Resultados das Simulações

De modo a testar o algoritmo BNDR-LMS em situações mais práticas, foram feitas simulações para vários problemas de identificação de sistemas com sinais de entrada não restritos a uma dada função de distribuição angular como nos experimentos da seção anterior. Inicialmente, a ordem do sistema foi ajustada para $N = 10$, o sinal de entrada foi um ruído colorido com um número de condicionamento (da matriz de autocorrelação) em torno de 187 e uma razão de sinal para ruído de observação (SNR) de 60dB e 150dB. As curvas de aprendizagem (MSE em dB) para os algoritmos NLMS, NNDR-LMS (uma reutilização) e BNDR-LMS estão mostradas na Figura 2.4, correspondendo a uma média de 200 realizações. Neste primeiro experimento, o passo μ foi ajustado para 1 de modo a obtermos a taxa de convergência

mais rápida do algoritmo BNDR-LMS. A escolha de uma reutilização para o algoritmo NNDR-LMS proporcionou complexidades computacionais similares para os algoritmos testados.

Neste exemplo podemos claramente verificar o desempenho superior do algoritmo BNDR-LMS em termos de velocidade de convergência quando comparado aos algoritmos NLMS e NNDR-LMS (uma reutilização simples) para o caso de elevada relação sinal-ruído. Esta vantagem torna-se mais evidente para casos de sinais de entrada ainda mais correlacionados. Simulações para o algoritmo LMS convencional e para o algoritmo DR-LMS foram também desenvolvidas mas seus desempenhos foram, como esperados, muito inferiores ao obtido pelo algoritmo NLMS e seus resultados foram omitidos da Figura 2.4. Em relação a este exemplo, vale a pena mencionarmos que para o algoritmo NNDR-LMS, um desempenho similar poderia ser obtido se tivéssemos usado no mínimo quatro reutilizações ($L = 4$) com os mesmos dois pares de dados (ou em torno de duas reutilizações se tivéssemos aumentado a informação com um par extra de dados). Isto significa que mais que o dobro do esforço computacional do algoritmo BNDR-LMS seria necessário para o algoritmo NNDR-LMS, no caso da mesma quantidade de memória, para ter uma taxa de convergência similar.

Num segundo experimento, ainda com $\mu = 1$ e $N = 10$, o excesso de MSE (ξ_{exc} em dB) foi medido de modo a testar o desempenho do algoritmo BNDR-LMS em termos de erro médio quadrático após convergência. O ξ_{min} , neste caso a variância do ruído de medida, foi ajustado para 10^{-6} enquanto que o sinal de entrada foi feito um processo ruído branco Gaussiano de média zero. Os resultados estão resumidos na Tabela 2.2 onde podemos também observar o excesso de MSE em dB para um ambiente não estacionário. Neste caso, o ruído de observação foi feito zero e os coeficientes do sistema (planta) foram variados de acordo com um modelo de passeio aleatório (random walk) generalizado, $\mathbf{w}(k) = \mathbf{w}(k-1) + \mathbf{v}$, onde \mathbf{v} era um vetor aleatório com elementos de média zero e variância igual a 10^{-6} . Como podemos ver da Tabela 2.2, em ambos ambientes estacionário e não estacionário, o algoritmo BNDR-LMS obteve um desempenho próximo aos algoritmos NLMS e NNDR-LMS.

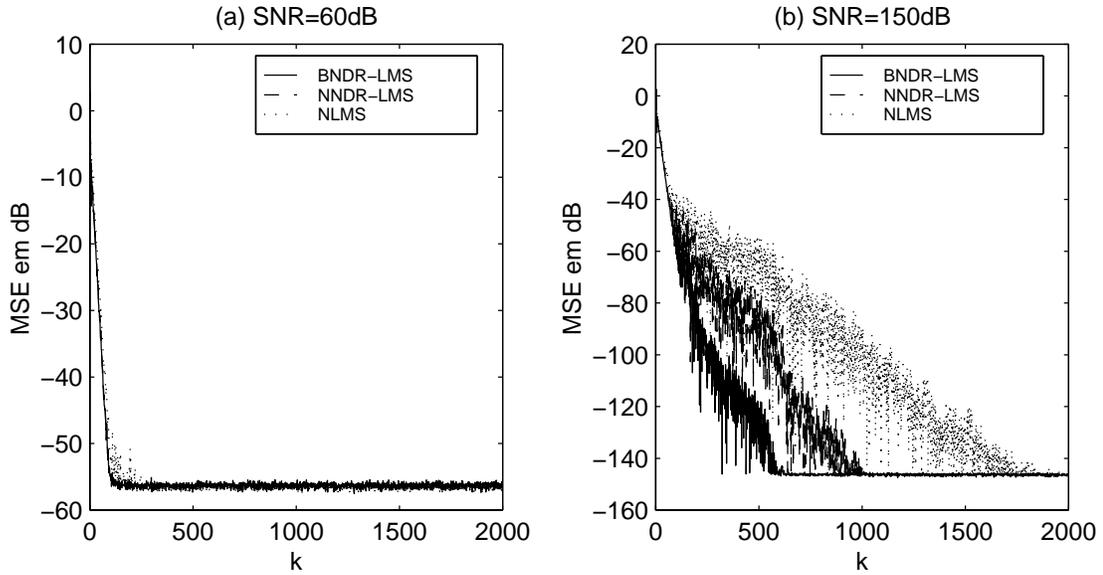


Figura 2.4: MSE para os algoritmos NLMS, NNDR-LMS e BNR-LMS.

Novamente o tamanho de passo μ foi ajustado para 1 neste experimento.

Tabela 2.2: Excesso de Erro Médio Quadrático

Tipo de Algoritmo	$(\xi_{exc})_{dB}$	
	Estacionário	Não estacionário
NLMS	-59,09	-39,15
NNDR-LMS	-59,40	-39,42
BNDR-LMS	-58,60	-39,45

Outros experimentos foram realizados de modo a testar os resultados teóricos obtidos da análise de convergência. O sinal de entrada neste caso foi feito ruído branco e o excesso de MSE foi medido para diferentes valores de passos (μ variando de 0,1 a 1,9). Uma vez que foi mostrado que a ordem do filtro (N) tem uma grande influência nos resultados teóricos, o experimento foi repetido para $N = 5$, $N = 10$, e $N = 63$. Os resultados são mostrados nas Figuras 2.5, 2.6 e 2.7 respectivamente, onde podemos ver que a curva teórica é mais próxima da curva experimental a

medida que N é aumentado. Além disto, a medida que N é aumentado, a probabilidade de ocorrer $\mathbf{V}_k = \mathbf{V}_{k-1}$ torna-se menor e as curvas se aproximam da obtida no Experimento 2 da seção anterior.

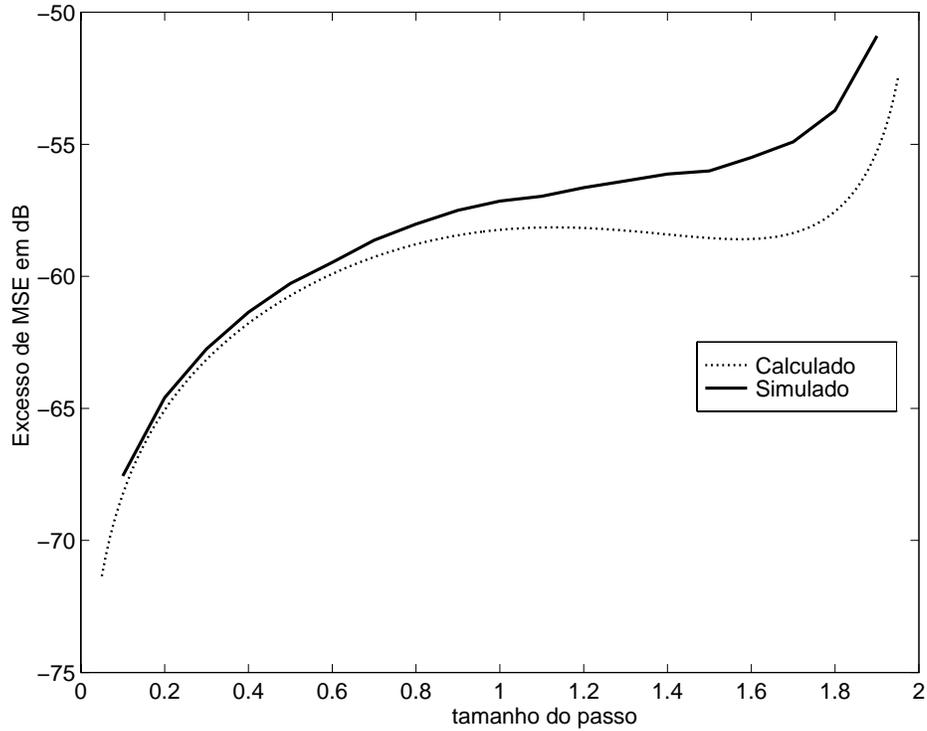


Figura 2.5: Excesso de MSE para $N = 5$ como função de μ .

Um último experimento foi planejado para testar a influência de sinais coloridos no excesso de MSE e a precisão das expressões derivadas na análise. Quatro situações foram consideradas correspondendo a sinais de entrada tendo diferentes características (todas com $N = 10$). Nas duas primeiras situações, os sinais foram obtidos de seqüências Gaussianas brancas e de média zero filtradas por filtros IIR só pólos de primeira ordem com pólos em 0,8 e 0,9, levando a matrizes de autocorrelação com razões de autovalores de 50,85 e 145,44, respectivamente. Nas outras duas situações, os vetores sinal de entrada foram gerados com distribuições de probabilidades radiais discretas e matrizes de autocorrelação com dispersão de autovalores também iguais a 50,85 e 145,44, respectivamente. O excesso de MSE em dB para

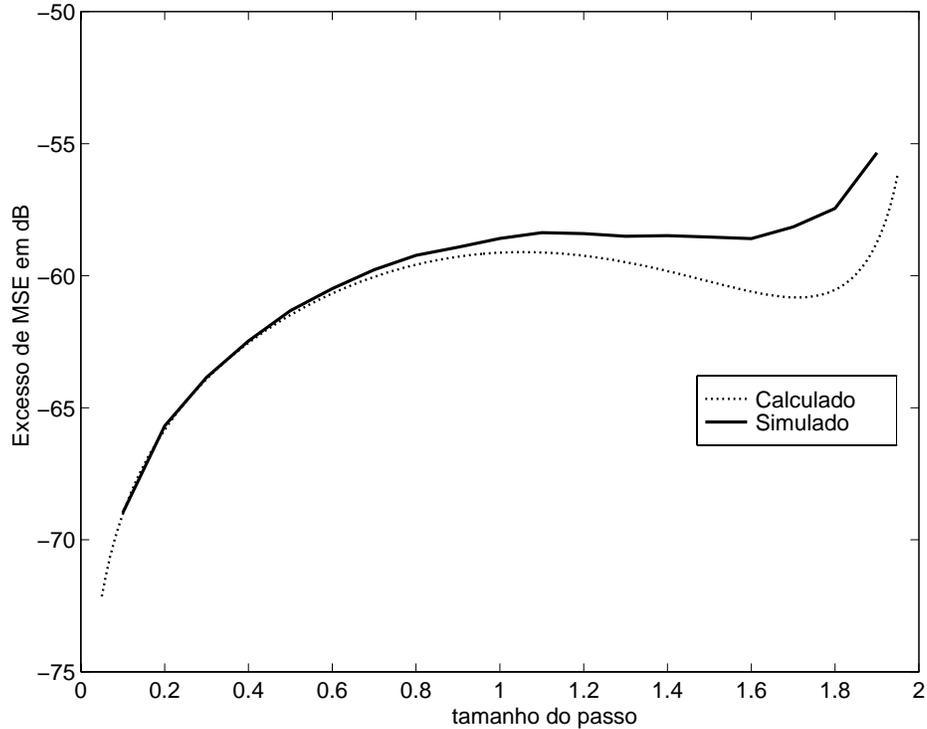


Figura 2.6: Excesso de MSE para $N = 10$ como função de μ .

estas simulações está mostrado na Figura 2.8 onde as curvas teóricas e resultantes de simulações são confrontadas. Os valores teóricos foram calculados usando (2.37)–(2.39) com probabilidades dadas por (2.45) e (2.46). A análise para sinais de entrada coloridos apresentou uma concordância muito boa com as simulações realizadas com os vetores sinal de entrada com distribuições de probabilidades angulares discretas. Para os sinais obtidos pela filtragem de sequências brancas Gaussianas por filtros IIR só pólos de primeira ordem, somente uma descrição qualitativa razoável da evolução do excesso de MSE em relação ao passo pôde ser observada. Isto pode ser explicado pelo fato das expressões terem sido derivadas para a situação de sinal de entrada branco. Além disto, observamos que na faixa de interesse ($0 < \mu \leq 1$) a diferença entre as curvas simulada e teórica é menor que 3 dB. A faixa de valores de μ entre 1 e 2 não é usada na prática pois tais valores iriam comprometer o desempenho do algoritmo elevando o desajuste sem proporcionar melhoria de velocidade

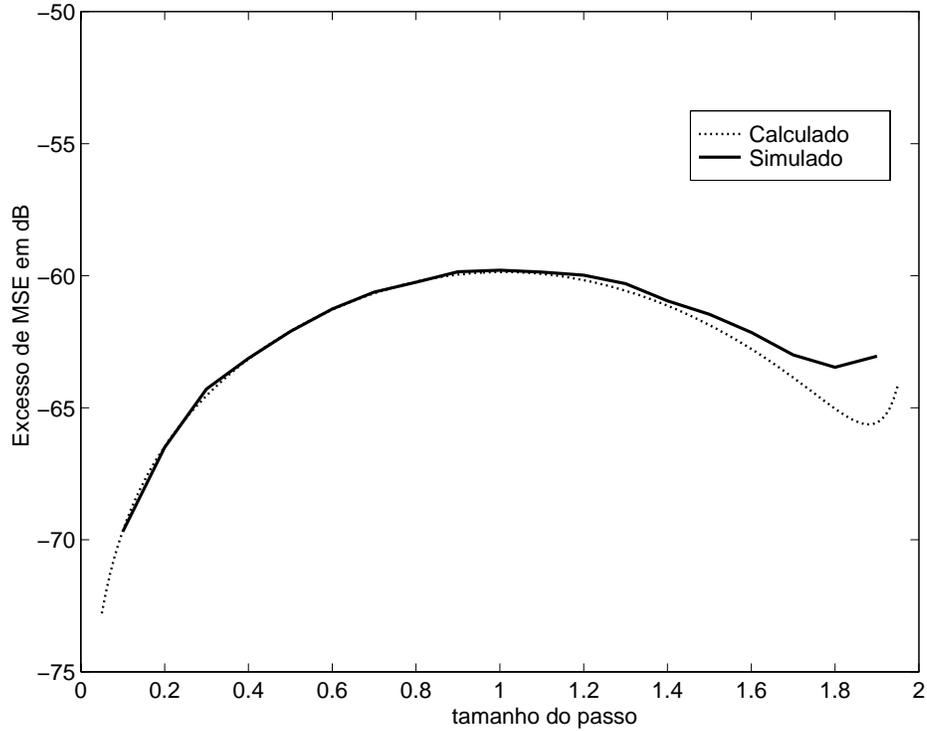


Figura 2.7: Excesso de MSE para $N = 63$ como função de μ .

de convergência que é máxima para $\mu = 1$.

Em termos de complexidade computacional, a Tabela 2.3 mostra as comparações entre os três algoritmos normalizados mencionados anteriormente. Note que $p = N + 1$ é o número de coeficientes. Olhando para esta tabela, podemos concluir que a carga computacional do algoritmo BNDR-LMS é ligeiramente superior a do algoritmo NNDR-LMS (que é igual a $L + 1 = 2$ vezes a complexidade do algoritmo NLMS). Ressaltamos ainda que a referida tabela é relativa a um único re-uso ($L=1$) e que para um desempenho do algoritmo NNDR-LMS igual ao do algoritmo BNDR-LMS no primeiro experimento, sua complexidade seria de $L + 1 = 5$ vezes a complexidade do algoritmo NLMS.

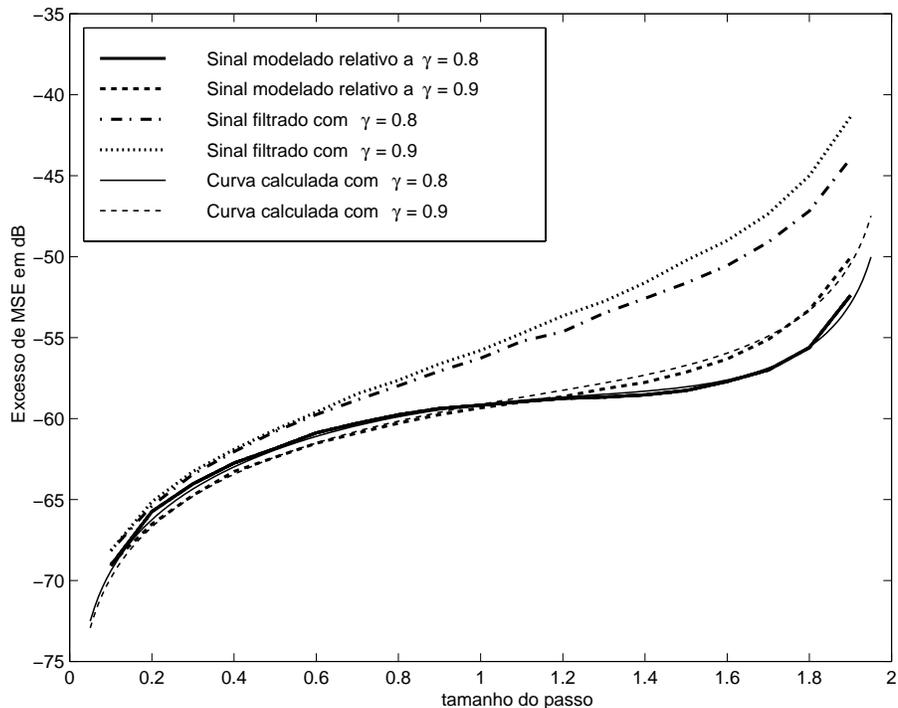


Figura 2.8: Excesso de MSE para sinais de entrada coloridos.

2.5 Conclusões

Este capítulo apresentou o algoritmo BNDR-LMS junto com sua análise de convergência e erro médio quadrático. Uma interpretação geométrica do algoritmo foi também proporcionada mostrando que os coeficientes são atualizados em dois passos normalizados seguindo direções ortogonais. A relação entre o algoritmo BNDR-LMS e o algoritmo das projeções ortogonais foi esclarecida.

Simulações realizadas numa aplicação de identificação de sistemas mostrou que o algoritmo BNDR-LMS apresentou resultados favoráveis em relação a outros algoritmos normalizados em termos de velocidade de convergência. Além disto, quanto mais correlacionado era o sinal de entrada, melhor era o desempenho do novo algoritmo quando comparado com os demais algoritmos do tipo LMS. Esta melhora é mais claramente observada em casos de pequeno ruído de medida.

Análises na média e na covariância foram desenvolvidas sendo a última basea-

Tabela 2.3: Comparação de complexidade computacional (para $L=1$)

ALGORITMO	SOMA	MULTIPLICAÇÃO	DIVISÃO
NLMS	$3p-1$	$3p$	1
NNDR-LMS	$6p-2$	$6p$	2
BNDR-LMS	$6p+1$	$6p+8$	2

da num modelo simplificado para o sinal de entrada o qual resultou em expressões tratáveis para o complexo problema de analisar algoritmos que reutilizam dados passados. Consistência com os dois primeiros momentos do sinal de entrada é mantida pelo modelo. Para sinais brancos, a análise de erro médio quadrático foi realizada com excelente concordância com as simulações. Limites para convergência na média e covariância do vetor de coeficientes foram estabelecidas. Além disso, uma expressão em forma fechada para o excesso de MSE como uma função do passo foi derivada para o caso de sinais de entrada brancos. A aplicabilidade desta expressão para o caso de sinais de entrada coloridos foi também explorada. O modelo e as análises podem ser prontamente estendidas a outros algoritmos que reutilizam dados que não foram considerados para análise no passado devido à complexidade excessiva.

Capítulo 3

Uma Aplicação Prática do Algoritmo BNDR-LMS

3.1 Introdução

Visando ilustrar uma aplicação prática do algoritmo BNDR-LMS, este capítulo apresenta uma versão com restrições deste algoritmo bem como propõe uma seqüência de passos ótima a qual permite convergência rápida e desajuste mínimo. O novo algoritmo é aplicado a um receptor digital móvel que usa espalhamento de espectro do tipo seqüência direta com acesso múltiplo por divisão de código ou “direct-sequence code-division multiple access” (DS-CDMA), tendo mostrado como resultado uma considerável melhoria da taxa de convergência quando comparada à abordagem tradicional usando o algoritmo LMS.

Um filtro adaptativo com restrições possui muitos campos de aplicação tais como processamento de “array” de antenas e cancelamento de interferência em sistemas de comunicações móveis que usam DS-CDMA. Os dois métodos tradicionais usados nestas aplicações são a chamada abordagem de Frost [30] e a abordagem conhecida em inglês como “the general sidelobe canceler” (GSC) [31]. A abordagem GSC converte o problema com restrições num outro sem restrições. O esquema de Frost é provavelmente a técnica mais largamente utilizada devido à simplicidade do algoritmo LMS. Contudo, a principal desvantagem do algoritmo LMS está também

presente no esquema Frost; isto é, seu desempenho é fortemente dependente do espalhamento de autovalores da matriz de autocorrelação do sinal de entrada. Uma abordagem alternativa é o uso de técnicas de mínimos quadrados rápidas como proposto em [32]. Sendo o algoritmo usado por Frost uma projeção do resultado do algoritmo LMS convencional no hiperplano definido pelas restrições, um passo natural seria o uso de algoritmos da família dos LMS normalizados convencionais [2, 33] seguidos de uma projeção tal como no caso LMS Frost. Esta abordagem resulta numa taxa de convergência superior comparada ao algoritmo LMS Frost quando os sinais de entrada são fortemente correlacionados. Esta abordagem intuitiva, entretanto, carece de um critério de otimização e apresenta desempenho pior que o obtido da estrutura GSC em alguns casos.

Foi observado em nossos experimentos que o algoritmo LMS com restrições em ambas implementações, esquemas Frost e GSC, apresenta resultados idênticos[31] quando aplicados ao mesmo conjunto de dados. Nosso objetivo neste capítulo é a derivação de um algoritmo BNDR-LMS com restrições (“Constrained BNDR-LMS”), estrutura do tipo Frost, tal que apresente resultado idêntico ao obtido empregando-se o algoritmo BNDR-LMS sem restrições na estrutura GSC.

Este capítulo é organizado como segue. A Seção 3.2 apresenta uma alternativa à derivação dos algoritmos NLMS e BNDR-LMS convencionais. Na Seção 3.3 os algoritmos normalizados com restrições são derivados com base na abordagem usada na Seção 3.2. Na Seção 3.4 a seqüência de passos ótima é derivada. A Seção 3.5 mostra alguns resultados de simulações no campo de aplicação proposto aqui, seguida pelas conclusões.

3.2 Re-Derivação dos Algoritmos NLMS e BNDR-LMS

Nesta seção nós mostramos uma maneira alternativa de derivarmos os algoritmos NLMS e BNDR-LMS. Começemos pelo algoritmo LMS normalizado. Vamos supor

que temos um algoritmo tipo LMS que atualiza o vetor de coeficientes de acordo com a seguinte expressão.

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \mu_k \mathbf{x}(k) \quad (3.1)$$

onde $\mathbf{w}(k)$ é o vetor de coeficientes (de tamanho $(N+1) \times 1$ onde N é a ordem do filtro adaptativo) no instante k , $\mathbf{x}(k)$ é o vetor sinal de entrada e μ_k é o passo variável (ou fator de convergência variável) o qual deve ser escolhido com o objetivo de proporcionar uma máxima velocidade de convergência. A estratégia usada aqui é reduzir o erro ao quadrado instantâneo tanto quanto possível pois esta é uma boa e simples estimativa do erro médio quadrático (MSE) [2]. Uma vez que o erro instantâneo é dado por $e(k) = d(k) - \mathbf{x}^T(k)\mathbf{w}(k)$, o erro quadrado instantâneo no instante k após a *atualização* do vetor de coeficientes pode ser escrito como

$$\begin{aligned} e^{*2}(k) &\triangleq [d(k) - \mathbf{x}^T(k)\mathbf{w}(k+1)]^2 \\ &= [d(k) - \mathbf{x}^T(k)(\mathbf{w}(k) + \mu_k \mathbf{x}(k))]^2 \end{aligned} \quad (3.2)$$

onde a *estrela* (*) indica o erro *a posteriori*. De modo a aumentar a taxa de convergência por escolher um passo apropriado, tomamos a derivada parcial de $e^{*2}(k)$ em relação a μ_k e a fazemos igual a zero, obtendo

$$\mu_k = \frac{d(k) - \mathbf{x}^T(k)\mathbf{w}(k)}{\mathbf{x}^T(k)\mathbf{x}(k)} \quad (3.3)$$

o qual corresponde, como esperado, ao algoritmo LMS normalizado tradicional.

Para o algoritmo BNDR-LMS, nós atualizamos o vetor de coeficientes somando os vetores sinal de entrada $\mathbf{x}(k)$ e $\mathbf{x}(k-1)$ ponderados por dois passos, μ_{1k} e μ_{2k} , respectivamente.

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \mu_{1k}\mathbf{x}(k) + \mu_{2k}\mathbf{x}(k-1) \quad (3.4)$$

Neste caso, nós minimizamos a função custo $F(k)$, a qual corresponde ao erro quadrado instantâneo no instante k mais o erro quadrado instantâneo no instante $k-1$ calculado com o vetor de coeficientes do instante k ou $F(k) = [d(k) - \mathbf{x}^T(k)\mathbf{w}(k)]^2 + [d(k-1) - \mathbf{x}^T(k-1)\mathbf{w}(k)]^2$. Em seguida definimos $F^*(k)$ como $F(k)$ calculada com

o vetor de coeficientes atualizado.

$$\begin{aligned}
F^*(k) &\triangleq [d(k) - \mathbf{x}^T(k)(\mathbf{w}(k) + \mu_{1k}\mathbf{x}(k) + \mu_{2k}\mathbf{x}(k-1))]^2 \\
&+ [d(k-1) - \mathbf{x}^T(k-1)(\mathbf{w}(k) + \mu_{1k}\mathbf{x}(k) + \mu_{2k}\mathbf{x}(k-1))]^2
\end{aligned} \tag{3.5}$$

Num próximo passo, tomamos as derivadas parciais de $F^*(k)$ em relação a μ_{1k} e μ_{2k} e as fazemos iguais a zero. Após algumas manipulações algébricas, obtemos

$$\begin{aligned}
e_1 &= d(k) - \mathbf{x}^T(k)\mathbf{w}(k) \\
e_2 &= d(k-1) - \mathbf{x}^T(k-1)\mathbf{w}(k) \\
den &= \mathbf{x}^T(k)\mathbf{x}(k)\mathbf{x}^T(k-1)\mathbf{x}(k-1) - (\mathbf{x}^T(k-1)\mathbf{x}(k))^2 \\
\mu_{1k} &= \frac{e_1\mathbf{x}^T(k-1)\mathbf{x}(k-1) - e_2\mathbf{x}^T(k-1)\mathbf{x}(k)}{den} \\
\mu_{2k} &= \frac{e_2\mathbf{x}^T(k)\mathbf{x}(k) - e_1\mathbf{x}^T(k-1)\mathbf{x}(k)}{den}
\end{aligned} \tag{3.6}$$

a qual junto com (3.4) correspondem ao algoritmo LMS binormalizado com reutilização de dados (BNDR-LMS).

3.3 O Algoritmo com Restrições

Numa filtragem adaptativa linear com restrições, as J restrições são representadas pelo seguinte sistema linear.

$$\mathbf{C}^T \mathbf{w}(k) = \mathbf{f} \tag{3.7}$$

onde \mathbf{C} é uma matriz $(N+1) \times J$ contendo os vetores de restrições, e \mathbf{f} é um vetor de J elementos contendo os valores das restrições (uma restrição simples significa que \mathbf{C} é um vetor e \mathbf{f} é um escalar).

No caso do algoritmo LMS (estrutura Frost), o algoritmo resultante é dado pela projeção do vetor de coeficientes — $\mathbf{w}(k+1)$ sem restrições — no hiperplano definido por (3.7). O vetor de coeficientes com restrições é obtido projetando-se inicialmente a solução sem restrições no hiperplano homogêneo $\mathbf{C}^T \mathbf{w}(k) = \mathbf{0}$ com o auxílio da matriz de projeção $\mathbf{P} = \mathbf{I} - \mathbf{C}(\mathbf{C}^T \mathbf{C})^{-1} \mathbf{C}^T$. Finalmente, o vetor resultante é movido

de volta ao hiperplano das restrições pela adição do vetor $\mathbf{F} = \mathbf{C}(\mathbf{C}^T \mathbf{C})^{-1} \mathbf{f}$.

$$\begin{aligned} \mathbf{w}(k+1) &= \mathbf{P}\mathbf{w}_{LMS}(k+1) + \mathbf{F} \\ &= \mathbf{P}[\mathbf{w}(k) + \mu e(k)\mathbf{x}(k)] + \mathbf{F} \end{aligned} \quad (3.8)$$

onde $e(k) = d(k) - \mathbf{x}^T(k)\mathbf{w}(k)$.

Nossa abordagem aqui para ambos algoritmos NLMS e BNDR-LMS é a projeção da solução sem restrições seguida de uma otimização do(s) passo(s) similar ao que foi feito na seção anterior. Começamos pelo algoritmo NLMS tomando $\mathbf{w}_{NLMS}(k+1)$ como em 3.1 onde μ_k é o passo variável que desejamos obter.

$$\begin{aligned} \mathbf{w}(k+1) &= \mathbf{P}\mathbf{w}_{NLMS}(k+1) + \mathbf{F} \\ &= \mathbf{P}[\mathbf{w}(k) + \mu_k \mathbf{x}(k)] + \mathbf{F} \end{aligned} \quad (3.9)$$

Se lembrarmos que $\mathbf{w}(k)$ foi forçado a satisfazer a restrição em (3.7), o que significa que $\mathbf{P}\mathbf{w}(k) + \mathbf{F} = \mathbf{w}(k)$, segue que (3.9) pode ser escrita como

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \mu_k \mathbf{P}\mathbf{x}(k) \quad (3.10)$$

Se agora compararmos (3.1) e (3.10) podemos ver que elas descrevem o mesmo problema se substituirmos o vetor de entrada por um vetor rotacionado $\mathbf{x}'(k) = \mathbf{P}\mathbf{x}(k)$. Além disto, lembrando que $\mathbf{P}^2 = \mathbf{P}$, segue da mesma abordagem usada na seção prévia que

$$\begin{aligned} e(k) &= d(k) - \mathbf{x}^T(k)\mathbf{w}(k) \\ \mathbf{w}(k+1) &= \mathbf{P} \left[\mathbf{w}(k) + \frac{e(k)\mathbf{x}(k)}{\mathbf{x}^T(k)\mathbf{P}\mathbf{x}(k)} \right] + \mathbf{F} \end{aligned} \quad (3.11)$$

que corresponde ao algoritmo NLMS com restrições [34].

A mesma abordagem pode ser aplicada ao algoritmo BNDR-LMS se fizermos

$$\begin{aligned} \mathbf{w}(k+1) &= \mathbf{P}\mathbf{w}_{BNDR-LMS}(k+1) + \mathbf{F} \\ &= \mathbf{P}[\mathbf{w}(k) + \mu_{1k}\mathbf{x}(k) + \mu_{2k}\mathbf{x}(k-1)] + \mathbf{F} \\ &= \mathbf{w}(k) + \mu_{1k}\mathbf{P}\mathbf{x}(k) + \mu_{2k}\mathbf{P}\mathbf{x}(k-1) \end{aligned} \quad (3.12)$$

e compararmos com (3.4). As equações do algoritmo BNDR-LMS com restrições ou “constrained BNDR-LMS” são obtidas como

$$\begin{aligned}
e_1 &= d(k) - \mathbf{x}^T(k)\mathbf{w}(k) \\
e_2 &= d(k-1) - \mathbf{x}^T(k-1)\mathbf{w}(k) \\
\mathbf{P} &= \mathbf{I} - \mathbf{C}(\mathbf{C}^T\mathbf{C})^{-1}\mathbf{C}^T \\
\mathbf{F} &= \mathbf{C}(\mathbf{C}^T\mathbf{C})^{-1}\mathbf{f} \\
den &= \mathbf{x}^T(k)\mathbf{P}\mathbf{x}(k)\mathbf{x}^T(k-1)\mathbf{P}\mathbf{x}(k-1) - (\mathbf{x}^T(k-1)\mathbf{P}\mathbf{x}(k))^2 \\
\mu_{1k} &= \frac{e_1\mathbf{x}^T(k-1)\mathbf{P}\mathbf{x}(k-1) - e_2\mathbf{x}^T(k-1)\mathbf{P}\mathbf{x}(k)}{den} \\
\mu_{2k} &= \frac{e_2\mathbf{x}^T(k)\mathbf{P}\mathbf{x}(k) - e_1\mathbf{x}^T(k-1)\mathbf{P}\mathbf{x}(k)}{den} \\
\mathbf{w}(k+1) &= \mathbf{P}[\mathbf{w}(k) + \mu_{1k}\mathbf{x}(k) + \mu_{2k}\mathbf{x}(k-1)] + \mathbf{F}
\end{aligned} \tag{3.13}$$

Vale a pena mencionar que estes dois algoritmos com restrições apresentam resultados idênticos aos obtidos com os algoritmos NLMS e BNDR-LMS sem restrições usados em estruturas GSC. É também interessante ressaltar que (3.11) e (3.13) podem ser simplificadas admitindo-se que $\mathbf{P}\mathbf{w}(k) + \mathbf{F} = \mathbf{w}(k)$. Esta simplificação, entretanto, pode produzir acumulação de erro de arredondamento quando o algoritmo é implementado em precisão finita [30].

3.4 Otimização do Passo do Algoritmo BNDR-LMS

Vimos que o algoritmo BNDR-LMS oferece uma convergência mais rápida que vários outros algoritmos LMS normalizados para sinais de entrada altamente correlacionados ao custo de uma pequena complexidade adicional. O MSE após a convergência para este algoritmo é controlado pelo parâmetro tamanho do passo μ . Para $\mu = 1$, temos a convergência mais rápida e também o maior MSE se comparado ao caso onde o valor do passo é próximo de zero. No capítulo anterior, foi mostrado que o algoritmo BNDR-LMS converge se o passo está na faixa de zero a dois. Por razões

práticas, o valor de μ é mantido entre zero e um pois foi observado que o MSE em estado estacionário era maior e a convergência mais lenta quando o passo era tomado entre um e dois.

Nesta seção, a expressão para o MSE desenvolvida no Capítulo 2 é usada para propor uma seqüência de passos ótima que proporciona convergência rápida e desajuste mínimo. A expressão final para o comportamento de convergência do algoritmo BNDR-LMS é rescrita aqui em termos do excesso no MSE.

$$\begin{aligned}\Delta\xi(k+1) &= \left[1 + \frac{\mu(\mu-2)}{N+1}\right] \Delta\xi(k) \\ &+ \frac{N\mu(1-\mu)^2(\mu-2)}{(N+1)^2} \Delta\xi(k-1) \\ &+ \frac{(1+N(\mu-2)^2)\mu^2}{(N+1)(N+2-\nu_x)} \sigma_n^2\end{aligned}\quad (3.14)$$

Da expressão de $\Delta\xi(k+1)$ acima, seguiremos uma abordagem similar à usada em [4] e começamos por rescrever (3.14) assumindo que até o instante k nós temos a seqüência ótima $\mu^*(0)$ a $\mu^*(k-1)$ já disponível e também as quantidades ótimas $\Delta\xi^*(k)$ e $\Delta\xi^*(k-1)$.

$$\begin{aligned}\Delta\xi(k+1) &= \left[1 + \frac{\mu(k)(\mu(k)-2)}{N+1}\right] \Delta\xi^*(k) \\ &+ \frac{N\mu(k)(1-\mu(k))^2(\mu(k)-2)}{(N+1)^2} \Delta\xi^*(k-1) \\ &+ \frac{(1+N(\mu(k)-2)^2)\mu(k)^2}{(N+1)^2} \sigma_n^2\end{aligned}\quad (3.15)$$

Se agora computarmos a derivada de $\Delta\xi(k+1)$ com relação a $\mu(k)$ e igualarmos a zero, obtemos após algumas manipulações algébricas

$$\begin{aligned}\mu^*(k) &= 1 - \sqrt{1 - \frac{\Delta\xi^*(k) + \Delta\xi^*(k-1)}{2(\Delta\xi^*(k-1) + \sigma_n^2)}} \\ &= 1 - \sqrt{1 - \frac{\xi^*(k) + \xi^*(k-1) - 2\sigma_n^2}{2\xi^*(k-1)}}\end{aligned}\quad (3.16)$$

É válido mencionarmos que (3.16) está de acordo com a situação encontrada após a convergência ser alcançada, ou seja, termos $\xi^*(k) = \xi^*(k-1) = \sigma_n^2$ e portanto termos $\mu^*(k) = 0$ como esperado. Além disto, se tivermos $\sigma_n^2 = 0$ o valor de $\mu^*(k)$

será próximo a um (admitindo que $\Delta\xi^*(k) \approx \Delta\xi^*(k-1)$) mesmo após convergência, o que implica em termos máxima velocidade de convergência com mínimo desajuste se o ruído é zero.

Para o algoritmo LMS normalizado (NLMS), a fórmula recursiva para $\mu^*(k)$ em termos de $\mu^*(k-1)$ e da ordem N foi obtida em [4]. No caso do algoritmo BNDR-LMS, uma expressão recursiva simples não foi obtida e um pequeno algoritmo foi usado para produzir a seqüência de passo variável ótima. Este algoritmo é apresentado na Tabela 3.1 ¹ e possui um importante parâmetro de inicialização com uma forte influência no comportamento de $\mu^*(k)$. Este parâmetro é a razão $\frac{\sigma_d^2}{\sigma_n^2}$ onde o numerador é a variância do sinal de referência.

Tabela 3.1: Algoritmo para computar a seqüência de passo variável ótima.

$\mu(k)$ do algoritmo BNDR-LMS
$\Delta\xi(0) = \Delta\xi(-1) = \sigma_d^2$
$\sigma_n^2 =$ variância do ruído
$N =$ ordem do filtro adaptativo
$\mu(0) = 1$
for $k = 1, 2, \dots$
{ $\mu(k) = 1 - \sqrt{1 - \frac{\Delta\xi(k) + \Delta\xi(k-1)}{2(\Delta\xi(k-1) + \sigma_n^2)}}$
$aa = \left[1 + \frac{\mu(k)(\mu(k)-2)}{N+1} \right]$
$bb = \frac{N\mu(k)(1-\mu(k))^2(\mu(k)-2)}{(N+1)^2}$
$cc = \frac{(1+N(\mu(k)-2)^2)\mu(k)^2}{(N+1)^2} \sigma_n^2$
$\Delta\xi(k+1) = aa\Delta\xi(k) + bb\Delta\xi(k-1) + cc$
}

Apresentamos a seguir na Figura 3.1 as curvas de $\mu(k)$ para diferentes valores do que poderia neste caso ser chamado de relação sinal (desejado) ruído ou $SNR =$

¹Note que o asterisco (*) foi retirado dos valores ótimos visando exclusivamente uma simplificação.

$10 \log \frac{\sigma_d^2}{\sigma_n^2}$ de 0 a 40 dB. Note que para $\sigma_n^2 = 0$ (caso sem ruído), o SNR se aproxima de infinito enquanto que o tamanho de passo permanece fixo na unidade.

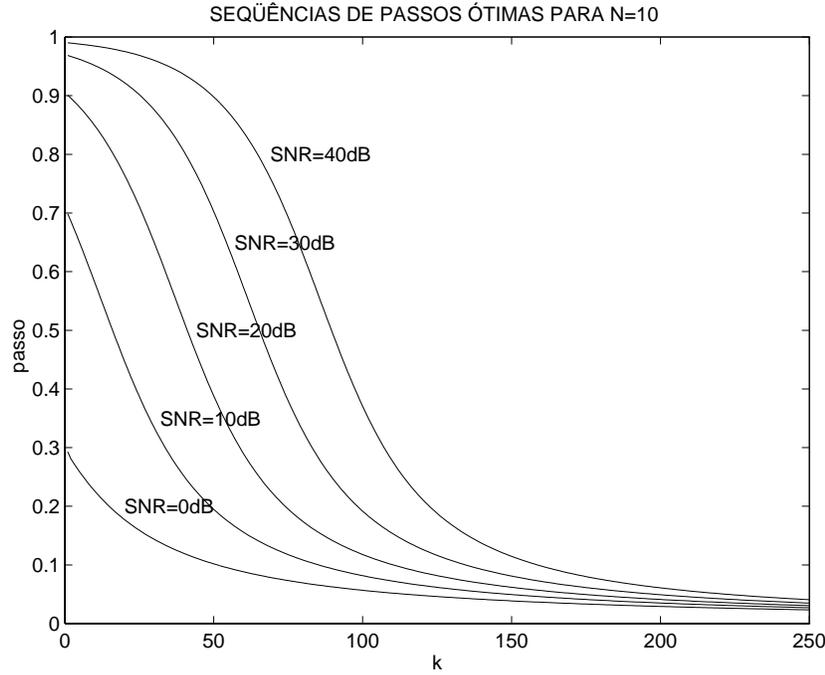


Figura 3.1: Seqüências de $\mu(k)$ ótimas para o algoritmo BNDR-LMS.

Numa implementação prática esta seqüência ótima pode ser computada *a priori* e armazenada na memória ou computada “on the fly”. Para esta última opção, desde que uma fórmula recursiva e compacta não está disponível, uma aproximação para esta curva é de grande interesse. Usaremos aqui duas classes de seqüências também propostas em [4]. Elas foram escolhidas devido à sua simplicidade e, como veremos posteriormente, por conduzirem a bons resultados. A primeira classe é a seqüência ótima para o algoritmo NLMS a qual é dada por

$$\mu(k) = \mu(k-1) \frac{1 - \frac{\mu(k-1)}{N+1}}{1 - \frac{\mu^2(k-1)}{N+1}} \quad (3.17)$$

Para o algoritmo NLMS, a inicialização correta para esta seqüência é dada por $\mu(0) = 1 - \frac{\sigma_d^2}{\sigma_d^2}$. Entretanto, no nosso caso podemos escolher um valor inicial para o passo tal que as duas seqüências (a ótima para o algoritmo NLMS e a ótima para o algoritmo BNDR-LMS) fiquem próximas, como será visto posteriormente.

A segunda classe de seqüências (referidas daqui para frente por aproximação $1/k$) é bastante simples e também foi usada em [4]. Esta seqüência é dada por

$$\mu(k) = \begin{cases} 1 & \text{se } 0 \leq k \leq c(N + 1) \\ \max\{\mu_{min}, \frac{1}{1-c+\frac{k}{N+1}}\} & \text{se } k > c(N + 1) \end{cases} \quad (3.18)$$

O parâmetro c está relacionado ao SNR da seqüência ótima. Um valor de passo mínimo foi introduzido aqui (podendo também ser usado em todas seqüências) visando prover o algoritmo de uma capacidade de acompanhamento em caso de pequena não-estacionariedade do ambiente.

Os resultados de alguns experimentos demonstrarão o desempenho superior do algoritmo com o esquema de passo variável proposto. Para a primeira simulação, usamos como sinal de entrada um ruído branco numa configuração de identificação de sistema com $N = 10$, $\sigma_n^2 = 10^{-2}$ e $SNR = 20dB$. A Figura 3.2 mostra a seqüência de passos ótima obtida com o algoritmo descrito na Tabela 3.1 e outras curvas relativas às duas classes de aproximações usadas.

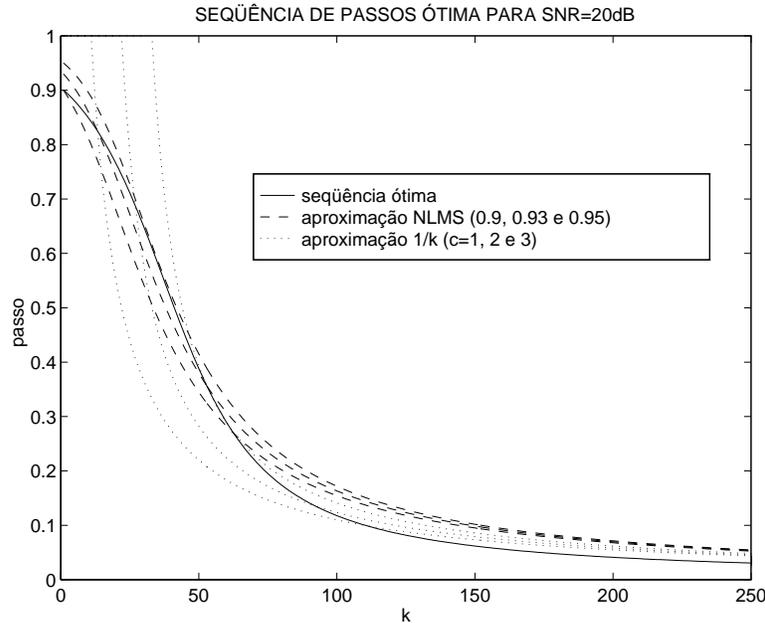


Figura 3.2: Seqüência de passos ótima e duas classes de seqüências aproximadas.

Da Figura 3.2, podemos conjecturar qual curva devemos usar. Se usássemos a mínima norma da diferença entre as seqüências ótima e aproximada como um

critério para decidirmos qual curva implementar, os parâmetros da equação (3.18) escolhidos para este exemplo seriam $\mu(0) = 0.93$ e $c = 3$.

Com estes parâmetros rodamos uma simulação com um valor de passo fixo, a seqüência ótima e duas aproximações. As curvas de aprendizado (média de 1000 rodadas) estão mostradas na Figura 3.3 onde podemos ver que o mesmo valor pequeno de MSE em estado estacionário é compartilhado pelas três seqüências de passos variáveis usadas. O passo fixo foi ajustado para um ϵ , como esperado, resulta no desajuste mais alto.

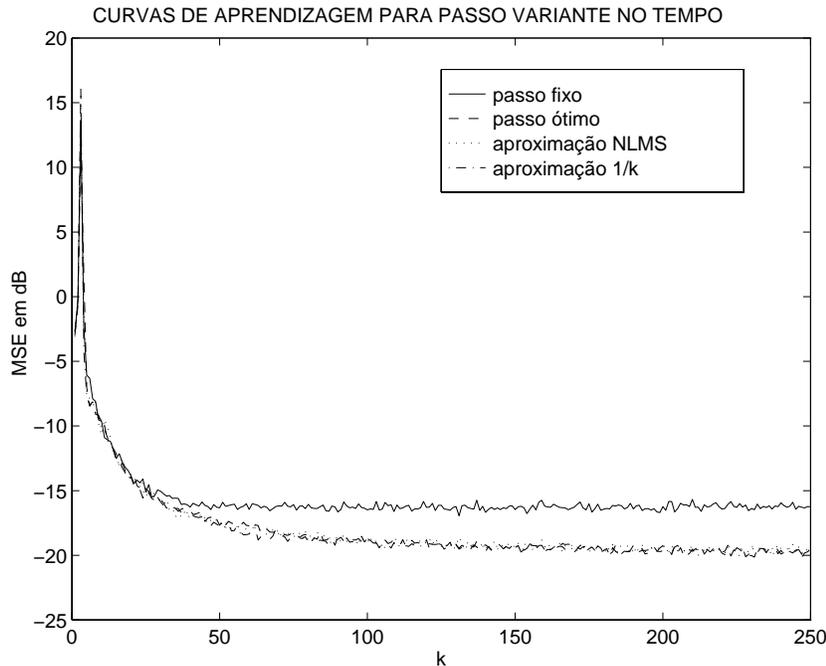


Figura 3.3: Curvas de aprendizado para passo fixo, passo variável ótimo e suas duas aproximações.

Um segundo experimento foi realizado de modo a avaliar o desempenho desta seqüência ótima no caso onde o sinal de entrada é correlacionado. A mesma configuração foi usada com um sinal de entrada tendo um número de condicionamento (razão entre o maior e o menor autovalor da matriz de autocorrelação do sinal de entrada) em torno de 180. A Figura 3.4 mostra-nos que, mesmo para um sinal de entrada correlacionado, a seqüência de passos variáveis proposta tem um bom desempenho. Observa-se na mesma figura que, no exemplo dado, o algoritmo BNDR-LMS

com seqüência ótima tem um desempenho melhor que o algoritmo NLMS também com seqüência ótima.

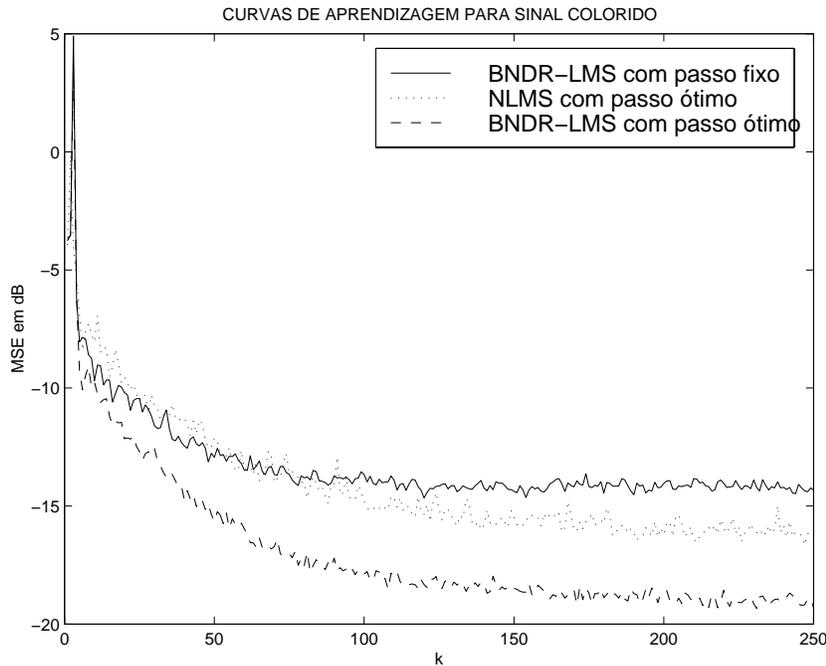


Figura 3.4: Comparando as curvas de aprendizagem para o caso de sinal de entrada colorido.

Uma observação final é a possibilidade de usarmos um estimador para $\xi(k)$ ao invés de calcularmos $\Delta\xi(k)$ usando (3.15) como descrito no algoritmo da Tabela 3.1. Um outro experimento foi feito usando o seguinte estimador:

$$\xi(k + 1) = \lambda\xi(k) + (1 - \lambda)e^2(k) \quad (3.19)$$

Este experimento mostrou-nos que um valor razoável para λ é algo em torno de 0.96. A vantagem desta abordagem alternativa é a possibilidade de um acompanhamento rápido de mudanças súbitas e fortes no ambiente. Neste caso, o erro instantâneo torna-se alto e a estimativa $\xi(k + 1)$ é elevada tal que o valor de μ se aproxima da unidade novamente e uma rápida re-adaptação se inicia.

Quando usando esta abordagem, vale a pena lembrarmos que, dado que a equação (3.16) é do tipo $1 - \sqrt{1 - x}$, o passo $\mu(k)$ pode ser escrito como $\frac{x}{1 + \sqrt{1 - x}}$ o qual é uma expressão numericamente menos sensível. A equação (3.20) mostra-nos esta

expressão.

$$\mu(k) = \frac{\frac{\xi(k) + \xi(k-1) - 2\sigma_n^2}{2\xi(k-1)}}{1 + \sqrt{1 - \frac{\xi(k) + \xi(k-1) - 2\sigma_n^2}{2\xi(k-1)}}} \quad (3.20)$$

3.5 Resultados das Simulações

Nesta seção, aplicamos o algoritmo BNDR-LMS com restrições ao caso de eliminação de interferência numa transmissão digital usando DS-CDMA da estação base para a estação móvel (“downlink”) de um sistema de comunicações móveis.

O sinal recebido para o sistema com K usuários simultâneos pode ser escrito como

$$\mathbf{x}(k) = \sum_{i=1}^K A_i b_i(k) \mathbf{s}_i + \mathbf{n}(k) \quad (3.21)$$

onde A_i é a amplitude do sinal do usuário i , \mathbf{s}_i é a seqüência de assinatura (código) do i -ésimo usuário e $b_i(k) \in \{\pm 1\}$ é o bit transmitido do i -ésimo usuário. No receptor móvel estaremos somente interessados em detectar um usuário (aqui assumido ser usuário $i = 1$). Uma maneira de construir os coeficientes do receptor é minimizar sua energia de saída tendo como restrição a parcela deste sinal correspondente ao código do usuário desejado passar com resposta unitária. O problema é, então, encontrar um vetor de coeficientes $\mathbf{w}(k)$ tal que resolva

$$\min_{\mathbf{w}(k)} \|\mathbf{x}^T(k) \mathbf{w}(k)\|^2 \quad \text{sujeito a} \quad \mathbf{s}_1^T \mathbf{w}(k) = 1 \quad (3.22)$$

onde, usando a notação da seção anterior, vemos que o sinal referência $d(k) = 0$, $\mathbf{C} = \mathbf{s}_1$ e $\mathbf{f} = 1$. O sistema usado em nosso experimento consiste de $K = 5$ usuários cujas seqüências de espalhamento foram códigos de Gold de tamanho 7 [35]. A relação sinal-ruído (SNR) para o usuário desejado foi ajustada para 8dB e a potência de interferência dos usuários foi ajustada para 20 vezes mais forte que a potência do usuário desejado ($A_i = \sqrt{20}$ para $i \neq 1$). Na simulação, usamos a seqüência de passos ótima [33] descrita na seção anterior.

A Figura 3.5 mostra as curvas de aprendizagem para os algoritmos LMS, NLMS e BNDR-LMS (média de 500 rodadas). O valor do passo para o algoritmo LMS

foi escolhido para ser $\mu = 5 \times 10^{-4}$ tal que seu desajuste seja comparável ao dos algoritmos normalizados usando seqüências ótimas. Como pode ser visto nesta figura, os desempenhos dos algoritmos NLMS e BNDR-LMS são superiores ao do algoritmo LMS em termos de taxa de convergência. Devido provavelmente ao sinal de entrada o qual não é suficientemente correlacionado neste exemplo, o algoritmo BNDR-LMS não apresentou o melhor desempenho e o algoritmo NLMS é sugerido nestes casos. Esta afirmação é baseada no fato de que mesmo o algoritmo RLS não mostrou um desempenho muito melhor que o do algoritmo NLMS neste particular exemplo. Vale a pena mencionarmos que a estrutura GSC usando os algoritmos NLMS e BNDR-LMS também foi simulada e apresentou curvas de aprendizagem idênticas.

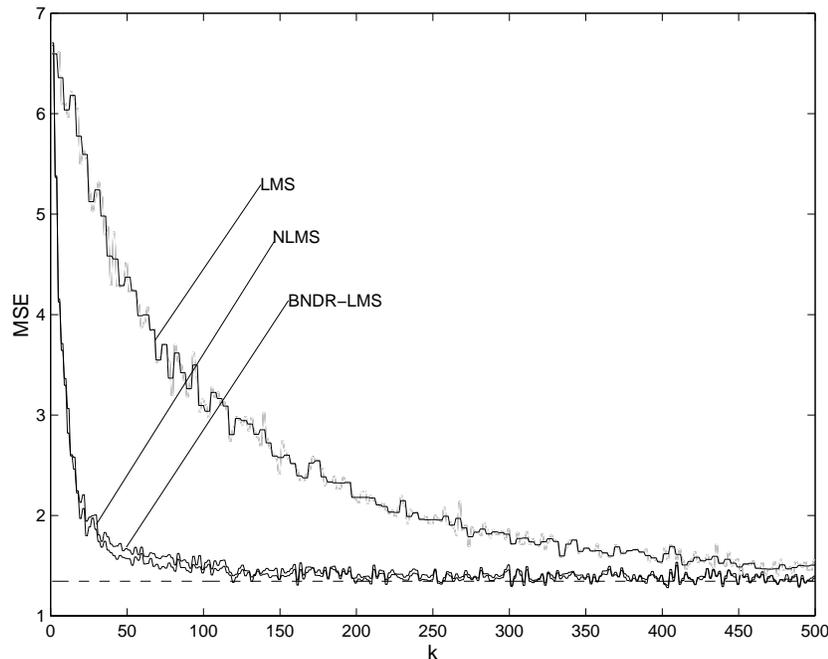


Figura 3.5: Curvas de aprendizagem dos algoritmos LMS, NLMS e BNDR-LMS com restrições.

3.6 Conclusões

Este capítulo introduziu a versão com restrições do algoritmo BNDR-LMS usando a estrutura desenvolvida por Frost [30]. Um método simples de obter os algoritmos normalizados (NLMS e BNDR-LMS) foi apresentado e foi mostrado que este método é também válido para as versões com restrições destes algoritmos. O algoritmo BNDR-LMS com restrições resultante usando a estrutura Frost apresentou resultados idênticos em relação ao seu equivalente sem restrições usando estrutura GSC. Uma seqüência de passos variáveis ótima para o algoritmo BNDR-LMS foi também obtida. O algoritmo foi aplicado à recepção móvel CDMA e resultados de simulação mostraram taxa de convergência mais rápida assim como um pequeno desajuste quando o passo variável ótimo é usado.

Capítulo 4

Algoritmos QR Rápidos: uma Abordagem Unificada

4.1 Introdução

Esta seção trata dos conceitos básicos usados nos algoritmos RLS que empregam decomposição QR convencional e inversa. Os métodos de triangularização da matriz de entrada de dados e o significado das variáveis internas destes algoritmos são enfatizados de modo a instituir a notação usada neste trabalho bem como introduzir as relações mais importantes usadas daqui por diante.

4.1.1 O Algoritmo QR Convencional

Assim como no algoritmo RLS convencional, nós estamos interessados em minimizar a seguinte função custo

$$\xi(k) = \sum_{i=0}^k \lambda^{k-i} e^2(i) = \mathbf{e}^T(k) \mathbf{e}(k) = \|\mathbf{e}(k)\|^2 \quad (4.1)$$

onde cada componente do vetor $\mathbf{e}(k)$ é o erro *a posteriori* no instante i ponderado por $\lambda^{(k-i)/2}$ (λ é o fator de esquecimento e i varia de 0 a k). O vetor $\mathbf{e}(k)$ é dado por

$$\mathbf{e}(k) = \mathbf{d}(k) - \mathbf{X}(k)\mathbf{w}(k) \quad (4.2)$$

Na equação acima, o vetor de sinal desejado ponderado $\mathbf{d}(k)$, o vetor de coeficientes $\mathbf{w}(k)$ e a matriz de dados de entrada $\mathbf{X}(k)$ são definidos por

$$\mathbf{d}(k) = \begin{bmatrix} d(k) \\ \lambda^{1/2}d(k-1) \\ \vdots \\ \lambda^{k/2}d(0) \end{bmatrix} \quad (4.3)$$

$$\mathbf{w}(k) = \begin{bmatrix} w_0(k) \\ w_1(k) \\ \vdots \\ w_N(k) \end{bmatrix} \quad (4.4)$$

$$\mathbf{X}(k) = \begin{bmatrix} \mathbf{x}^T(k) \\ \lambda^{1/2}\mathbf{x}^T(k-1) \\ \vdots \\ \lambda^{k/2}\mathbf{x}^T(0) \end{bmatrix} \quad (4.5)$$

onde N é a ordem do filtro (número de coeficientes menos um), $\mathbf{x}(k)$ é o vetor sinal de entrada $[x(k) \ x(k-1) \ \cdots \ x(k-N)]^T$ e as amostras antes do instante $k=0$ são consideradas nulas.

A solução ótima para o problema dos mínimos quadrados num dado instante k pode ser encontrada ao derivarmos a função custo em relação a $\mathbf{w}(k)$ e igualarmos a zero, resultando em

$$\mathbf{w}(k) = \mathbf{R}_D^{-1}(k)\mathbf{p}_D(k) \quad (4.6)$$

onde $\mathbf{R}_D(k) = \mathbf{X}^T(k)\mathbf{X}(k)$ é a matriz de correlação de dados determinística e $\mathbf{p}_D(k) = \mathbf{X}^T(k)\mathbf{d}(k)$ é o vetor de correlação cruzada determinístico entre a entrada e o sinal desejado.

A expressão (4.6) é usada na abordagem RLS convencional. A inversa de $\mathbf{R}_D(k)$ pode tornar-se mal condicionada, por exemplo, devido a perda de persistência de excitação do sinal de entrada ou devido a efeitos de quantização [2]. Visando evitar

possíveis soluções imprecisas, a abordagem de decomposição QR usada nesta tese triangularizará a matriz de dados de entrada através do uso de matrizes de rotação de Givens.

A premultiplicação de (4.2) por $\mathbf{Q}(k)$ (matriz a qual representa o processo global de triangularização via matrizes de rotação de Givens elementares) triangulariza $\mathbf{X}(k)$ e uma vez que $\mathbf{Q}(k)$ é ortogonal (na verdade é ortonormal), ela não afetará a função custo.

$$\mathbf{Q}(k)\mathbf{e}(k) = \begin{bmatrix} \mathbf{e}_{q_1}(k) \\ \mathbf{e}_{q_2}(k) \end{bmatrix} = \begin{bmatrix} \mathbf{d}_{q_1}(k) \\ \mathbf{d}_{q_2}(k) \end{bmatrix} - \begin{bmatrix} \mathbf{O} \\ \mathbf{U}(k) \end{bmatrix} \mathbf{w}(k) \quad (4.7)$$

onde $\mathbf{U}(k)$ é o fator Cholesky de $\mathbf{X}^T(k)\mathbf{X}(k)$ (i.e. $\mathbf{U}^T(k)\mathbf{U}(k) = \mathbf{X}^T(k)\mathbf{X}(k)$) e os subscritos 1 e 2 indicam as primeiras $k - N$ e últimas $N + 1$ componentes do vetor, respectivamente.

O erro quadrático ponderado (ou função custo) pode ser minimizado escolhendo-se $\mathbf{w}(k)$ tal que o termo $\mathbf{d}_{q_2}(k) - \mathbf{U}(k)\mathbf{w}(k)$ seja zero. Os coeficientes são então calculados usando o procedimento conhecido como “backsubstitution” (veja [2, 36] para mais detalhes).

Usando uma vez mais o fato que $\mathbf{Q}(k)$ é ortonormal e a definição em (4.5), nós podemos escrever o produto $\mathbf{Q}(k)\mathbf{X}(k)$ como

$$\mathbf{Q}(k) \begin{bmatrix} 1 & \mathbf{0}^T \\ \mathbf{0} & \mathbf{Q}^T(k-1) \end{bmatrix} \begin{bmatrix} 1 & \mathbf{0}^T \\ \mathbf{0} & \mathbf{Q}(k-1) \end{bmatrix} \begin{bmatrix} \mathbf{x}^T(k) \\ \lambda^{1/2}\mathbf{X}(k-1) \end{bmatrix} = \begin{bmatrix} \mathbf{O} \\ \mathbf{U}(k) \end{bmatrix} \quad (4.8)$$

Na equação acima, se chamarmos $\tilde{\mathbf{Q}}(k)$ o produto das duas primeiras matrizes à esquerda e executarmos a multiplicação das duas matrizes restantes, temos

$$\tilde{\mathbf{Q}}(k) \begin{bmatrix} \mathbf{x}^T(k) \\ \mathbf{O} \\ \lambda^{1/2}\mathbf{U}(k-1) \end{bmatrix} = \begin{bmatrix} \mathbf{O} \\ \mathbf{U}(k) \end{bmatrix} \quad (4.9)$$

De (4.9) vemos que $\tilde{\mathbf{Q}}(k)$ é um produto de matrizes de rotações de Givens que anula o vetor de entrada corrente. Além disto a natureza recursiva de $\mathbf{Q}(k)$ pode ser expressa por

$$\mathbf{Q}(k) = \tilde{\mathbf{Q}}(k) \begin{bmatrix} 1 & \mathbf{0}^T \\ \mathbf{0} & \mathbf{Q}(k-1) \end{bmatrix} \quad (4.10)$$

Uma vez que $\tilde{\mathbf{Q}}(k)$ é responsável por zerar $\mathbf{x}^T(k)$ como mostrado em (4.9), sua estrutura inclui uma sub-matriz \mathbf{I}_{k-N-1} . Felizmente, podemos evitar a característica de ordem sempre crescente ao deletarmos as linhas e colunas de $\tilde{\mathbf{Q}}(k)$ onde se localiza \mathbf{I}_{k-N-1} e rescrevermos (4.9) como

$$\begin{bmatrix} \mathbf{0}^T(k) \\ \mathbf{U}(k) \end{bmatrix} = \mathbf{Q}_\theta(k) \begin{bmatrix} \mathbf{x}^T(k) \\ \lambda^{1/2} \mathbf{U}(k-1) \end{bmatrix} \quad (4.11)$$

onde $\mathbf{Q}_\theta(k)$ é $\tilde{\mathbf{Q}}(k)$ após remover a seção \mathbf{I}_{k-N-1} junto com as linhas e colunas correspondentes.

Lembrando (4.7), vemos que $\mathbf{e}_{q_1}(k) = \mathbf{d}_{q_1}(k)$ e o produto $\mathbf{Q}(k)\mathbf{d}(k)$ podem ser escritos como

$$\begin{aligned} \begin{bmatrix} \mathbf{e}_{q_1}(k) \\ \mathbf{d}_{q_2}(k) \end{bmatrix} &= \tilde{\mathbf{Q}}(k) \begin{bmatrix} 1 & \mathbf{0}^T \\ \mathbf{0} & \mathbf{Q}(k-1) \end{bmatrix} \begin{bmatrix} \mathbf{d}(k) \\ \lambda^{1/2} \mathbf{d}(k-1) \end{bmatrix} \\ &= \tilde{\mathbf{Q}}(k) \begin{bmatrix} \mathbf{d}(k) \\ \lambda^{1/2} \mathbf{e}_{q_1}(k-1) \\ \lambda^{1/2} \mathbf{d}_{q_2}(k-1) \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{e}_{q_1}(k) \\ \lambda^{1/2} \mathbf{e}_{q_1}(k-1) \\ \mathbf{d}_{q_2}(k) \end{bmatrix} \end{aligned} \quad (4.12)$$

onde a última multiplicação pode ser facilmente entendida se notarmos em (4.9) que $\tilde{\mathbf{Q}}(k)$ alterará somente o primeiro e os últimos $N+1$ componentes de um vetor.

De (4.12), é também possível remover a seção crescente de $\tilde{\mathbf{Q}}(k)$ resultando na seguinte expressão:

$$\begin{bmatrix} e_{q_1}(k) \\ \mathbf{d}_{q_2}(k) \end{bmatrix} = \mathbf{Q}_\theta(k) \begin{bmatrix} d(k) \\ \lambda^{1/2} \mathbf{d}_{q_2}(k-1) \end{bmatrix} \quad (4.13)$$

onde $e_{q_1}(k)$ é o primeiro elemento do sinal de erro *rotacionado* e $\mathbf{d}_{q_2}(k)$ é um vetor com os últimos $N + 1$ elementos do vetor sinal desejado *rotacionado*.

Neste ponto, é importante enfatizar a estrutura de $\mathbf{Q}_\theta(k)$ como um produto de matrizes dado por $\prod_{i=N}^0 \mathbf{Q}_{\theta_i}(k)$. Esta estrutura dependerá do tipo de triangularização de $\mathbf{U}(k)$: matriz triangular superior ou inferior. Esta escolha, como será visto posteriormente, determinará também duas classes de algoritmos. O modo pelo qual a matriz $\mathbf{U}(k)$ é triangularizada está mostrado na Figura 4.1 com a correspondente matriz $\mathbf{Q}_{\theta_i}(k)$ sendo dada por

$$\text{SUPERIOR: } \mathbf{Q}_{\theta_i}(k) = \begin{bmatrix} \cos\theta_i(k) & \mathbf{0}^T & -\sin\theta_i(k) & \mathbf{0}^T \\ \mathbf{0} & \mathbf{I}_i & \mathbf{0} & \mathbf{0} \cdots \mathbf{0} \\ \sin\theta_i(k) & \mathbf{0}^T & \cos\theta_i(k) & \mathbf{0}^T \\ \mathbf{0} & \mathbf{0} \cdots \mathbf{0} & \mathbf{0} & \mathbf{I}_{N-i} \end{bmatrix} \quad (4.14)$$

$$\text{INFERIOR: } \mathbf{Q}_{\theta_i}(k) = \begin{bmatrix} \cos\theta_i(k) & \mathbf{0}^T & -\sin\theta_i(k) & \mathbf{0}^T \\ \mathbf{0} & \mathbf{I}_{N-i} & \mathbf{0} & \mathbf{0} \cdots \mathbf{0} \\ \sin\theta_i(k) & \mathbf{0}^T & \cos\theta_i(k) & \mathbf{0}^T \\ \mathbf{0} & \mathbf{0} \cdots \mathbf{0} & \mathbf{0} & \mathbf{I}_i \end{bmatrix} \quad (4.15)$$

É importante ressaltar que os ângulos $\theta_i(k)$ em (4.14) e (4.15) não são os mesmos embora escritos da mesma maneira por motivo de simplicidade. É também relevante mencionarmos que as duas possibilidades para uma matriz triangular superior (triângulo de zeros do lado inferior direito como na figura ou triângulo de zeros do lado inferior esquerdo) assim como as duas possibilidades para uma matriz triangular inferior (triângulo de zeros do lado superior esquerdo como na figura ou triângulo de zeros do lado superior direito) conduzem a algoritmos idênticos.

Como um exemplo, as estruturas de $\mathbf{Q}_\theta(k)$ para triangularizações superior e inferior de $\mathbf{U}(k)$ com $N = 2$ são dadas por

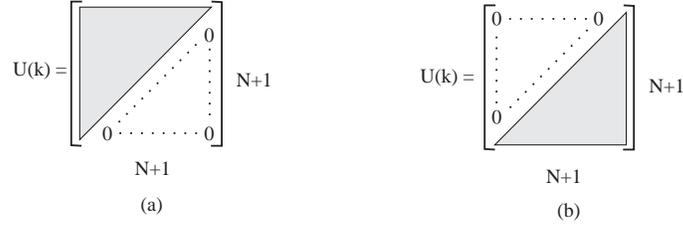


Figura 4.1: As diferentes triangularizações de $\mathbf{U}(k)$: (a) SUPERIOR e (b) INFERIOR.

$$SUPERIOR: \mathbf{Q}_\theta(k) = \begin{bmatrix} c\theta_2 c\theta_1 c\theta_0 & -c\theta_2 c\theta_1 s\theta_0 & -c\theta_2 s\theta_1 & -s\theta_2 \\ s\theta_0 & c\theta_0 & 0 & 0 \\ s\theta_1 c\theta_0 & -s\theta_1 s\theta_0 & c\theta_1 & 0 \\ s\theta_2 c\theta_1 c\theta_0 & -s\theta_2 c\theta_1 s\theta_0 & -s\theta_2 s\theta_1 & c\theta_2 \end{bmatrix} \quad (4.16)$$

$$INFERIOR: \mathbf{Q}_\theta(k) = \begin{bmatrix} c\theta_2 c\theta_1 c\theta_0 & -s\theta_2 & -c\theta_2 s\theta_1 & -c\theta_2 c\theta_1 s\theta_0 \\ s\theta_2 c\theta_1 c\theta_0 & c\theta_2 & -s\theta_2 s\theta_1 & -s\theta_2 c\theta_1 s\theta_0 \\ s\theta_1 c\theta_0 & 0 & c\theta_1 & -s\theta_1 s\theta_0 \\ s\theta_0 & 0 & 0 & c\theta_0 \end{bmatrix} \quad (4.17)$$

onde $c\theta_i = \cos\theta_i(k)$ e $s\theta_i = \sin\theta_i(k)$.

As últimas duas equações sugerem que $\mathbf{Q}_\theta(k)$ em ambos casos seja particionada como

$$\mathbf{Q}_\theta(k) = \begin{bmatrix} \gamma(k) & \mathbf{g}^T(k) \\ \mathbf{f}(k) & \mathbf{E}(k) \end{bmatrix} \quad (4.18)$$

onde $\gamma(k) = \prod_{i=0}^N \cos\theta_i(k)$ e os elementos de $\mathbf{f}(k)$, $\mathbf{g}(k)$ e $\mathbf{E}(k)$ dependem do tipo de triangularização.

De modo a ter todas as equações do algoritmo QR tradicional ($O[N^2]$), pós-multiplicamos o vetor transposto $\mathbf{e}_q^T(k)\mathbf{Q}(k)$ pelo vetor $[1 \ 0 \ \dots \ 0]^T$ conhecido em inglês pelo termo “pinning vector”.

$$\mathbf{e}_q^T(k)\mathbf{Q}(k) \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \mathbf{e}^T(k)\mathbf{Q}^T(k)\mathbf{Q}(k) \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} = e(k) \quad (4.19)$$

Por outro lado, das equações (4.10) e (4.18), e do fato de $\mathbf{Q}_\theta(k)$ ser $\tilde{\mathbf{Q}}(k)$ após remover as $k - N - 1$ colunas e linhas crescentes, podemos concluir que $\mathbf{Q}(k)[1 \ 0 \ \cdots \ 0]^T = [\gamma(k) \ 0 \ \cdots \ 0 \ \mathbf{f}^T(k)]^T$. Uma vez que $\mathbf{e}_{q_2}(k)$ é um vetor nulo (não esquecendo que $\mathbf{w}(k)$ em (4.7) foi escolhido de modo a fazê-lo igual ao vetor zero), é fácil mostrar que

$$e(k) = e_{q_1}(k)\gamma(k) \quad (4.20)$$

Esta última equação é particularmente útil em aplicações onde os coeficientes do filtro adaptativo não são necessários uma vez que podemos obter $e(k)$ sem calcular $\mathbf{w}(k)$.

As equações do algoritmo QR convencional são apresentadas na Tabela 4.1. Vale a pena mencionarmos que neste caso o tipo de triangularização usada não tem influência no desempenho do algoritmo e também que se os coeficientes do filtro são necessários, eles podem ser obtidos usando o procedimento conhecido como “backsubstitution” [2].

4.1.2 Interpretando as Variáveis Internas

Examinando a estrutura de $\mathbf{Q}_\theta(k)$ como expressa em (4.18) e recordando (4.11) e o fato de que esta matriz é ortonormal, podemos escrever as duas próximas equações.

$$\begin{bmatrix} \gamma(k) & \mathbf{g}^T(k) \\ \mathbf{f}(k) & \mathbf{E}(k) \end{bmatrix} \begin{bmatrix} \mathbf{x}^T(k) \\ \lambda^{1/2}\mathbf{U}(k-1) \end{bmatrix} = \begin{bmatrix} \mathbf{0}^T \\ \mathbf{U}(k) \end{bmatrix} \quad (4.21)$$

Tabela 4.1: As equações do algoritmo QR convencional.

QR
<p>for $k = 1, 2, \dots$</p> <p>{ Obtendo $\mathbf{Q}_\theta(k)$ e atualizando $\mathbf{U}(k)$:</p> $\begin{bmatrix} \mathbf{0}^T \\ \mathbf{U}(k) \end{bmatrix} = \mathbf{Q}_\theta(k) \begin{bmatrix} \mathbf{x}^T(k) \\ \lambda^{1/2}\mathbf{U}(k-1) \end{bmatrix}$ <p>Obtendo $\gamma(k)$:</p> $\gamma(k) = \prod_{i=0}^N \cos\theta_i(k)$ <p>Obtendo $e_{q_1}(k)$ e atualizando $\mathbf{d}_{q_2}(k)$:</p> $\begin{bmatrix} e_{q_1}(k) \\ \mathbf{d}_{q_2}(k) \end{bmatrix} = \mathbf{Q}_\theta(k) \begin{bmatrix} d(k) \\ \lambda^{1/2}\mathbf{d}_{q_2}(k-1) \end{bmatrix}$ <p>Obtendo $e(k)$:</p> $e(k) = e_{q_1}(k)\gamma(k)$ <p>}</p>

$$\begin{aligned} \mathbf{I}_{N+2} &= \mathbf{Q}_\theta(k)\mathbf{Q}_\theta^T(k) = \begin{bmatrix} \gamma(k) & \mathbf{g}^T(k) \\ \mathbf{f}(k) & \mathbf{E}(k) \end{bmatrix} \begin{bmatrix} \gamma(k) & \mathbf{f}^T(k) \\ \mathbf{g}(k) & \mathbf{E}^T(k) \end{bmatrix} \\ &= \mathbf{Q}_\theta^T(k)\mathbf{Q}_\theta(k) = \begin{bmatrix} \gamma(k) & \mathbf{f}^T(k) \\ \mathbf{g}(k) & \mathbf{E}^T(k) \end{bmatrix} \begin{bmatrix} \gamma(k) & \mathbf{g}(k)^T \\ \mathbf{f}(k) & \mathbf{E}(k) \end{bmatrix} \end{aligned} \quad (4.22)$$

De (4.21) e (4.22), um número de relações, as quais são comuns a ambos métodos de triangularização, pode ser derivada. Ressaltamos as seguintes relações

$$\mathbf{f}(k)\mathbf{x}^T(k) + \lambda^{1/2}\mathbf{E}(k)\mathbf{U}(k-1) = \mathbf{U}(k) \quad (4.23)$$

$$\gamma(k)\mathbf{f}(k) + \mathbf{E}(k)\mathbf{g}(k) = \mathbf{0} \quad (4.24)$$

Agora, se multiplicarmos a transposta de (4.11) por ela mesma, obtemos

$$\mathbf{U}^T(k)\mathbf{U}(k) = \mathbf{x}(k)\mathbf{x}^T(k) + \lambda\mathbf{U}^T(k-1)\mathbf{U}(k-1) \quad (4.25)$$

Pré-multiplicando (4.23) por $\mathbf{U}^T(k)$ e comparando a (4.25) encontramos que

$$\mathbf{f}(k) = \mathbf{U}^{-T}(k)\mathbf{x}(k) \quad (4.26)$$

$$\mathbf{E}(k) = \lambda^{1/2}\mathbf{U}^{-T}(k)\mathbf{U}^T(k-1) \quad (4.27)$$

Substituindo (4.26) e (4.27) em (4.24), segue que

$$\mathbf{g}(k) = -\gamma(k)\mathbf{U}^{-T}(k-1)\mathbf{x}(k)/\sqrt{\lambda} \quad (4.28)$$

Veremos que (4.26), (4.27) e (4.28) são relações chaves no entendimento dos outros algoritmos da família QR. De modo a compreendermos o significado destas variáveis, as quais dependem do método de triangularização, é necessário introduzir a aplicação da decomposição QR aos problemas de predição progressiva (forward) e regressiva (backward) bem como comparar o método QR discutido até agora com o procedimento de ortogonalização de Gram-Schmidt.

O Problema da Predição Regressiva

No problema da predição regressiva, tentamos obter uma estimativa da amostra passada de uma dada seqüência de entrada usando a informação sobre a seqüência atualmente disponível. O sinal $x(K-N-1)$ é o sinal desejado e a predição é baseada em $\mathbf{x}(k)$. O vetor de erro regressivo ponderado é

$$\mathbf{e}_b(k) = \mathbf{d}_b(k) - \mathbf{X}(k)\mathbf{w}_b(k) = \begin{bmatrix} x(k-N-1) \\ \lambda^{1/2}x(k-N-2) \\ \vdots \\ \lambda^{(k-N-1)/2}x(0) \\ \mathbf{0}_{N+1} \end{bmatrix} - \mathbf{X}(k)\mathbf{w}_b(k) \quad (4.29)$$

onde $\mathbf{w}_b(k)$ é o vetor de coeficientes de predição regressiva.

A última equação pode ser rescrita em termos de $\mathbf{X}^{(N+2)}(k)$, a matriz de dados de entrada de ordem $N+1$.

$$\mathbf{e}_b(k) = [\mathbf{X}(k) \mathbf{d}_b(k)] \begin{bmatrix} -\mathbf{w}_b(k) \\ 1 \end{bmatrix} = \mathbf{X}^{(N+2)}(k) \begin{bmatrix} -\mathbf{w}_b(k) \\ 1 \end{bmatrix} \quad (4.30)$$

O vetor de erro regressivo ponderado rotacionado é definido abaixo e será usado posteriormente na derivação dos algoritmos QR rápidos.

$$\mathbf{e}_{b_q}(k) = \mathbf{Q}(k)\mathbf{e}_b(k) = \begin{bmatrix} \mathbf{O} & \mathbf{e}_{b_{q_1}}(k) \\ \mathbf{U}(k) & \mathbf{d}_{b_{q_2}}(k) \end{bmatrix} \begin{bmatrix} -\mathbf{w}_b(k) \\ 1 \end{bmatrix} \quad (4.31)$$

O Problema de Predição Progressiva

No problema de predição progressiva, o sinal desejado é $x(k)$ e a predição será realizada com $\mathbf{x}(k-1)$. Em termos de vetores ponderados, temos

$$\mathbf{e}_f(k) = \mathbf{d}_f(k) - \begin{bmatrix} \mathbf{X}(k-1) \\ \mathbf{0}^T \end{bmatrix} \mathbf{w}_f(k) = \begin{bmatrix} x(k) \\ \lambda^{1/2}x(k-1) \\ \vdots \\ \lambda^{k/2}x(0) \end{bmatrix} - \begin{bmatrix} \mathbf{X}(k-1) \\ \mathbf{0}^T \end{bmatrix} \mathbf{w}_f(k) \quad (4.32)$$

onde $\mathbf{w}_f(k)$ é o vetor de coeficientes de predição progressiva.

A última equação pode também ser rescrita em termos de $\mathbf{X}^{(N+2)}(k)$, a matriz de dados de entrada de ordem $N+1$.

$$\mathbf{e}_f(k) = \begin{bmatrix} \mathbf{d}_f(k) & \mathbf{X}(k-1) \\ & \mathbf{0}^T \end{bmatrix} \begin{bmatrix} 1 \\ -\mathbf{w}_f(k) \end{bmatrix} = \mathbf{X}^{(N+2)}(k) \begin{bmatrix} 1 \\ -\mathbf{w}_f(k) \end{bmatrix} \quad (4.33)$$

O vetor de erros progressivos ponderados rotacionado é definido por

$$\mathbf{e}_{f_q}(k) = \begin{bmatrix} \mathbf{Q}(k-1) & \mathbf{0} \\ & \mathbf{0}^T & 1 \end{bmatrix} \mathbf{e}_f(k) = \begin{bmatrix} \mathbf{e}_{f_{q_1}}(k) & \mathbf{O} \\ \mathbf{d}_{f_{q_2}}(k) & \mathbf{U}(k-1) \\ \lambda^{k/2}x(0) & \mathbf{0}^T \end{bmatrix} \begin{bmatrix} 1 \\ -\mathbf{w}_f(k) \end{bmatrix} \quad (4.34)$$

É interessante ressaltar que todas as variáveis dos preditores regressivo e progressivo estão relacionadas a preditores de ordem N (o que equivale na notação aqui usada a $N + 1$ coeficientes de predição). Assim, temos $\mathbf{e}_b(k) = \mathbf{e}_b^{(N+1)}(k)$ e $\mathbf{e}_f(k) = \mathbf{e}_f^{(N+1)}(k)$.

Ortogonalização de Gram-Schmidt para $U(k)$ Triangular Inferior

A técnica de Gram-Schmidt busca um conjunto de vetores ortogonais $\{\mathbf{e}_0, \mathbf{e}_1, \dots, \mathbf{e}_N\}$ gerando o mesmo espaço gerado por um outro conjunto de vetores $\{\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_N\}$ que não são mutuamente ortogonais.

Isto é usualmente conseguido fazendo $\mathbf{e}_0 = \mathbf{x}_0$, $\mathbf{e}_1 = \mathbf{x}_1 - \hat{\mathbf{x}}_1$ (onde $\hat{\mathbf{x}}_1$ é a projeção de \mathbf{x}_1 em \mathbf{e}_0) e aí por diante, tal que $\mathbf{e}_i = \mathbf{x}_i - \hat{\mathbf{x}}_i$, $\hat{\mathbf{x}}_i = \sum_{j=0}^{i-1} k_{ji} \mathbf{e}_j$ com $k_{ji} = \mathbf{e}_j^T \mathbf{x}_i / \|\mathbf{e}_j\|^2$, $j < i$. Após obtermos \mathbf{e}_N , a ortonormalidade é forçada por substituirmos \mathbf{e}_i por $\mathbf{e}_i / \|\mathbf{e}_i\|$. Este procedimento triangulariza uma matriz consistindo de vetores \mathbf{x}_i como suas colunas. O resultado é uma matriz triangular superior.

Nós escolheremos um outro conjunto de vetores ortogonais tais que a matriz $\mathbf{X}(k)$ (rescrita abaixo) seja corretamente triangularizada.

$$\begin{aligned} \mathbf{X}(k) &= [\mathbf{x}_0 \ \mathbf{x}_1 \ \cdots \ \mathbf{x}_N] = \begin{bmatrix} \mathbf{x}^T(k) \\ \lambda^{1/2} \mathbf{x}^T(k-1) \\ \vdots \\ \lambda^{k/2} \mathbf{x}^T(0) \end{bmatrix} \\ &= \begin{bmatrix} x(k) & x(k-1) & \cdots & x(k-N) \\ \lambda^{1/2} x(k-1) & \lambda^{1/2} x(k-2) & & \vdots \\ \vdots & \vdots & & \lambda^{(k-N)/2} x(0) \\ \lambda^{(k-1)/2} x(1) & \lambda^{(k-1)/2} x(0) & & 0 \\ \lambda^{k/2} x(0) & 0 & \cdots & 0 \end{bmatrix} \quad (4.35) \end{aligned}$$

Fazendo $\mathbf{e}_N = \mathbf{x}_0$, $\mathbf{e}_{N-1} = \mathbf{x}_1 - \hat{\mathbf{x}}_1$, \dots , $\mathbf{e}_0 = \mathbf{x}_N - \hat{\mathbf{x}}_N$, nós temos $\mathbf{x}_0 = \mathbf{e}_N$, $\mathbf{x}_1 = \mathbf{e}_{N-1} + k_{N1} \mathbf{e}_N$, \dots , $\mathbf{x}_N = \mathbf{e}_0 + k_{1N} \mathbf{e}_1 + \dots + k_{NN} \mathbf{e}_N$, o que significa que

$$\mathbf{X}(k) = [\mathbf{e}_0 \ \mathbf{e}_1 \ \cdots \ \mathbf{e}_N] \begin{bmatrix} 0 & \cdots & 0 & 1 \\ \vdots & & & k_{1N} \\ 0 & & & \vdots \\ 1 & k_{N1} & \cdots & k_{NN} \end{bmatrix} = \mathbf{G}(k) \mathbf{K}(k) \quad (4.36)$$

Definindo $\mathbf{D}^2(k) = \mathbf{G}^T(k) \mathbf{G}(k)$ e sendo este produto uma matriz diagonal cujos elementos são $\|\mathbf{e}_i\|^2$, $\mathbf{D}(k)$ é uma matriz diagonal cujos elementos são $\|\mathbf{e}_i\|$. Podemos, então, escrever

$$\mathbf{U}^T(k) \mathbf{U}(k) = \mathbf{X}^T(k) \mathbf{X}(k) = \mathbf{K}^T(k) \mathbf{G}^T(k) \mathbf{G}(k) \mathbf{K}(k) = [\mathbf{D}(k) \mathbf{K}(k)]^T [\mathbf{D}(k) \mathbf{K}(k)] \quad (4.37)$$

Uma vez que $\mathbf{U}(k) = \mathbf{D}(k) \mathbf{K}(k)$ [37] e usando (4.18), (4.26), (4.27) e (4.28), temos a seguinte expressão para $\mathbf{Q}_\theta(k)$

$$\mathbf{Q}_\theta(k) = \begin{bmatrix} \gamma(k) & -\gamma(k) \lambda^{-1/2} [\mathbf{D}^{-1}(k-1) \mathbf{K}^{-T}(k-1) \mathbf{x}(k)]^T \\ \mathbf{D}^{-1}(k) \mathbf{K}^{-T}(k) \mathbf{x}(k) & \lambda^{1/2} \mathbf{D}^{-1}(k) \mathbf{K}^{-T}(k) \mathbf{K}^T(k-1) \mathbf{D}^T(k-1) \end{bmatrix} \quad (4.38)$$

É possível obter um significado físico para a expressão $\mathbf{K}^{-T}(k) \mathbf{x}(k)$ se (4.36) for inicialmente reescrita como

$$\mathbf{G}^T(k) = \mathbf{K}^{-T}(k) \mathbf{X}^T(k) = \mathbf{K}^{-T}(k) [\mathbf{x}(k) \ \lambda^{1/2} \mathbf{x}(k-1) \ \cdots] = \begin{bmatrix} \mathbf{e}_0^T \\ \mathbf{e}_1^T \\ \vdots \\ \mathbf{e}_N^T \end{bmatrix} \quad (4.39)$$

Do problema de predição regressivo (4.29), sabemos que

$$\mathbf{e}_b^{(i)}(k) = \mathbf{d}_b^{(i)}(k) - \mathbf{X}^{(i)}(k)\mathbf{w}_b^{(i)}(k) \quad (4.40)$$

onde $\mathbf{d}_b^{(i)}(k)$ é o vetor sinal desejado regressivo ponderado de ordem $i - 1$ ou $[x(k-i) \ \lambda^{1/2}x(k-i-1) \ \dots \ \lambda^{(k-i)/2}x(0) \ \mathbf{0}_i^T]^T$.

Diferenciando-se $\mathbf{e}_b^{(i)T}(k)\mathbf{e}_b^{(i)}(k)$ com relação a $\mathbf{w}_b^{(i)}(k)$, é simples mostrar que o vetor de coeficientes de predição regressiva ótimo é dado por

$$\mathbf{w}_b^{(i)}(k) = \left[\mathbf{X}^{(i)T}(k)\mathbf{X}^{(i)}(k) \right]^{-1} \mathbf{X}^{(i)T}(k)\mathbf{d}_b^{(i)}(k) \quad (4.41)$$

Se reconhecermos $\mathbf{d}_b^{(i)}(k)$ de (4.29) como \mathbf{x}_i em (4.35) e substituirmos primeiro (4.41) em (4.40) e então $\mathbf{X}^{(i)}(k)$ por $\mathbf{G}^{(i)}(k)\mathbf{K}^{(i)}(k)$ (de ordem $i - 1$), nós obtemos, após algumas manipulações algébricas, a expressão

$$\mathbf{e}_b^{(i)}(k) = \mathbf{x}_i - \sum_{j=0}^{i-1} \frac{\mathbf{e}_j \mathbf{e}_j^T \mathbf{x}_i}{\|\mathbf{e}_j\|^2} \quad (4.42)$$

a qual corresponde a $\mathbf{e}_{N-i} = \mathbf{x}_i - \hat{\mathbf{x}}_i$, ou seja, um dos vetores da nova base mostrada em (4.36).

Uma vez sabendo que \mathbf{e}_i é igual a $\mathbf{e}_b^{(N-i)}(k)$, estes valores podem ser usados em (4.39) e segue que

$$\mathbf{K}^{-T}(k)\mathbf{x}(k) = \begin{bmatrix} e_b^{(N)}(k) \\ e_b^{(N-1)}(k) \\ \vdots \\ e_b^{(0)}(k) \end{bmatrix} \quad (4.43)$$

onde o produto acima é o vetor de erros de predição regressiva *a posteriori* (com diferente número de coeficientes) no instante k .

Podemos ressaltar que (4.43) traz uma interpretação para os elementos não-nulos da linha de $\mathbf{K}^{-T}(k)$ como sendo os coeficientes de filtros de predição regressiva de

diferentes ordens. Recordando que os elementos da matriz diagonal $\mathbf{D}(k)$ são dados por $\| \mathbf{e}_i(k) \| = \| \mathbf{e}_b^{(N-i)}(k) \|$, o vetor $\mathbf{f}(k)$ de (4.18) como visto em (4.38) é dado por

$$\mathbf{f}(k) = \begin{bmatrix} e_b^{(N)}(k) / \| \mathbf{e}_b^{(N)}(k) \| \\ e_b^{(N-1)}(k) / \| \mathbf{e}_b^{(N-1)}(k) \| \\ \vdots \\ e_b^{(0)}(k) / \| \mathbf{e}_b^{(0)}(k) \| \end{bmatrix} \quad (4.44)$$

e será referido como o vetor de erros de predição regressiva *a posteriori* normalizado no instante k .

Além disso, usando a mesma interpretação de $\mathbf{D}(k-1)$ e $\mathbf{K}^{-T}(k-1)$, o vetor $\mathbf{g}(k)$ de (4.18) pode ser mostrado corresponder a

$$\mathbf{g}(k) = -\gamma(k)\mathbf{a}(k) = -\gamma(k)\lambda^{-1/2} \begin{bmatrix} e_b'^{(N)}(k) / \| \mathbf{e}_b^{(N)}(k-1) \| \\ e_b'^{(N-1)}(k) / \| \mathbf{e}_b^{(N-1)}(k-1) \| \\ \vdots \\ e_b'^{(0)}(k) / \| \mathbf{e}_b^{(0)}(k-1) \| \end{bmatrix} \quad (4.45)$$

onde $\mathbf{a}(k)$ é o vetor de erros de predição regressiva *a priori* no instante k normalizado pelas energias dos erros regressivos *a posteriori* no instante $k-1$ e ponderados por $\lambda^{-1/2}$.

A expressão para $\mathbf{E}(k)$ dada em (4.38), entretanto, não conduz a uma interpretação física relevante e (4.27) continua a melhor representação para $\mathbf{E}(k)$. Uma outra expressão alternativa para $\mathbf{E}(k)$ (ainda sem significado físico) é $\mathbf{A}^{-T}(k)$ onde $\mathbf{A}(k)$ é o fator Cholesky de $\mathbf{I} + \mathbf{a}(k)\mathbf{a}^T(k)$ [38].

Ortogonalização de Gram-Schmidt para $\mathbf{U}(k)$ Triangular Superior

As informações dadas em (4.44) e (4.45) são exclusivamente válidas para o caso de triangularização inferior de $\mathbf{U}(k)$. Para a triangularização superior da matriz $\mathbf{U}(k)$ devemos escolher um conjunto de vetores ortogonais diferente $\{\mathbf{e}_0, \mathbf{e}_1, \dots, \mathbf{e}_N\}$ tal que $\mathbf{e}_0 = \mathbf{x}_N, \mathbf{e}_1 = \mathbf{x}_{N-1} - \hat{\mathbf{x}}_{N-1}, \dots, \mathbf{e}_N = \mathbf{x}_0 - \hat{\mathbf{x}}_0$. Neste caso, nós temos

$\mathbf{x}_N = \mathbf{e}_0, \mathbf{x}_{N-1} = \mathbf{e}_1 + k_{0\ N-1}\mathbf{e}_0, \dots, \mathbf{x}_0 = \mathbf{e}_N + k_{00}\mathbf{e}_0 + k_{10}\mathbf{e}_1 + \dots + k_{N-1\ 0}\mathbf{e}_{N-1}$, o que implica em

$$\mathbf{X}(k) = [\mathbf{e}_0 \ \mathbf{e}_1 \ \dots \ \mathbf{e}_N] \begin{bmatrix} k_{00} & \dots & k_{0\ N-1} & 1 \\ \vdots & & & 0 \\ k_{N-1\ 0} & & & \vdots \\ 1 & 0 & \dots & \end{bmatrix} = \mathbf{G}(k)\mathbf{K}(k) \quad (4.46)$$

As equações (4.37) a (4.39) permanecem válidas neste caso de triangularização superior. Do problema de predição progressiva (4.32), temos

$$\mathbf{e}_f^{(i)}(k) = \mathbf{d}_f^{(i)}(k) - \begin{bmatrix} \mathbf{X}^{(i)}(k-1) \\ \mathbf{0}^T \end{bmatrix} \mathbf{w}_f^{(i)}(k) \quad (4.47)$$

onde $\mathbf{d}_f^{(i)}(k-i) = \mathbf{d}_f(k-i)$ e $\mathbf{d}_f(k-i) = \mathbf{x}_i$ de (4.35).

Derivando-se $[\mathbf{e}_f^{(i)}(k)]^T \mathbf{e}_f^{(i)}(k)$ em relação a $\mathbf{w}_f^{(i)}(k)$ e igualando o resultado a zero, encontramos o vetor de coeficientes progressivos ótimo de ordem $(i-1)$ e no instante k o qual é dado por

$$\mathbf{w}_f^{(i)}(k) = \left\{ [\mathbf{X}^{(i)}(k-1)]^T \mathbf{X}^{(i)}(k-1) \right\}^{-1} \begin{bmatrix} \mathbf{X}^{(i)}(k-1) \\ \mathbf{0}^T \end{bmatrix}^T \mathbf{d}_f^{(i)}(k) \quad (4.48)$$

Substituindo esta equação em (4.47) e fazendo $\mathbf{X}^{(i)}(k-1) = \mathbf{G}^{(i)}(k-1)\mathbf{K}^{(i)}(k-1)$, nós temos

$$\mathbf{e}_f^{(i)}(k) = \mathbf{d}_f^{(i)}(k) - \begin{bmatrix} \sum_{j=0}^{i-1} \mathbf{e}_j^{(i)}(k-1)[\mathbf{e}_j^{(i)}(k-1)]^T & \mathbf{0} \\ \mathbf{0}^T & 0 \end{bmatrix} \mathbf{d}_f^{(i)}(k) \quad (4.49)$$

da qual é possível obter os vetores de $\mathbf{G}(k)$ em (4.46) os quais são dados por

$$\mathbf{e}_i = \begin{bmatrix} \mathbf{e}_f^{(i)}(k-N+i) \\ \mathbf{0}_{N-i} \end{bmatrix} \quad (4.50)$$

Esta expressão para \mathbf{e}_i pode então ser usada em (4.39) para o caso da triangularização superior e segue-se que

$$\mathbf{K}^{-T}(k)\mathbf{x}(k) = \begin{bmatrix} e_f^{(0)}(k-N) \\ e_f^{(1)}(k-N+1) \\ \vdots \\ e_f^{(N)}(k) \end{bmatrix} \quad (4.51)$$

onde o produto acima é o vetor de erros (com diferentes ordens e em instantes distintos do tempo) de predição progressiva *a posteriori*.

Vale também a pena mencionarmos que (4.51) traz uma interpretação dos elementos não-nulos das linhas de $\mathbf{K}^{-T}(k)$ como os coeficientes de filtros de predição progressiva de diferentes ordens e em instantes distintos do tempo. Lembrando que os elementos da matriz diagonal $\mathbf{D}(k)$ são dados por $\|\mathbf{e}_i(k)\| = \|e_f^{(i)}(k-N+i)\|$, o vetor $\mathbf{f}(k)$ é agora dado por

$$\mathbf{f}(k) = \begin{bmatrix} e_f^{(0)}(k-N)/\|e_f^{(0)}(k-N)\| \\ e_f^{(1)}(k-N+1)/\|e_f^{(1)}(k-N+1)\| \\ \vdots \\ e_f^{(N)}(k)/\|e_f^{(N)}(k)\| \end{bmatrix} \quad (4.52)$$

e será referido como o vetor de erros de predição progressiva *a posteriori* normalizado.

Usando a mesma interpretação de $\mathbf{D}(k-1)$ e $\mathbf{K}^{-T}(k-1)$, o vetor $\mathbf{g}(k)$ corresponde no caso de triangularização superior a

$$\mathbf{g}(k) = -\gamma(k)\mathbf{a}(k) = -\gamma(k)\lambda^{-1/2} \begin{bmatrix} e_f^{\prime(0)}(k-N)/\|e_f^{(0)}(k-N-1)\| \\ e_f^{\prime(1)}(k-N+1)/\|e_f^{(1)}(k-N)\| \\ \vdots \\ e_f^{\prime(N)}(k)/\|e_f^{(N)}(k-1)\| \end{bmatrix} \quad (4.53)$$

onde $\mathbf{a}(k)$ neste caso é o vetor de erros de predição progressiva *a priori* normalizado pelas energias de erros de predição progressivos *a posteriori* e ponderados por $\lambda^{-1/2}$.

Vimos que neste caso de triangularização superior, os erros normalizados presentes em $\mathbf{Q}_\theta(k)$ são de diferentes ordens em instantes distintos de tempo (atualização de ordem e tempo) e este fato parece ser a causa do esforço computacional extra dos algoritmos rápidos derivados do uso deste tipo de triangularização.

4.1.3 O Algoritmo QR Inverso

Uma abordagem alternativa baseada na atualização do inverso do fator Cholesky foi apresentada em [15]. Este algoritmo, conhecido como QR inverso (IQR), permite o cálculo do vetor de pesos (ou coeficientes) sem “backsubstitution” e algumas de suas relações serão usadas posteriormente neste trabalho.

Começando de (4.6) e usando $[\mathbf{x}(k) \ \lambda^{1/2} \mathbf{X}^T(k-1)]^T$ ao invés de $\mathbf{X}(k)$ na definição de $\mathbf{R}_D(k)$ e $\mathbf{p}_D(k)$, nós podemos, após algumas manipulações, mostrar que

$$\mathbf{w}(k) = \mathbf{w}(k-1) + \mathbf{U}^{-1}(k) \mathbf{U}^{-T}(k) \mathbf{x}(k) e'(k) \quad (4.54)$$

onde $e'(k)$ é o erro *a priori* ou $d(k) - \mathbf{x}^T(k) \mathbf{w}(k-1)$ e o termo multiplicando esta variável é conhecido como *Ganho de Kalman*.

O produto $\mathbf{U}^{-1}(k) \mathbf{U}^{-T}(k)$ em (4.54) pode ser obtido de valores prévios se invertermos (4.25) e usarmos o tão chamado *lema de inversão de matriz* [2]. Após algumas operações algébricas, o resultado é

$$\begin{aligned} \mathbf{U}^{-1}(k) \mathbf{U}^{-T}(k) &= \lambda^{-1} \mathbf{U}^{-1}(k-1) \mathbf{U}^{-T}(k-1) - \\ &\quad - \lambda^{-1} \gamma^2(k) \mathbf{U}^{-1}(k-1) \mathbf{a}(k) \mathbf{a}^T(k) \mathbf{U}^{-T}(k-1) \end{aligned} \quad (4.55)$$

Definindo o vetor $\mathbf{u}(k)$ como

$$\mathbf{u}(k) = -\lambda^{-1/2} \gamma(k) \mathbf{U}^{-1}(k-1) \mathbf{a}(k) \quad (4.56)$$

nós podemos rescrever (4.54) e (4.55) como

$$\mathbf{U}^{-1}(k)\mathbf{U}^{-T}(k) + \mathbf{u}(k)\mathbf{u}^T(k) = \lambda^{-1}\mathbf{U}^{-1}(k-1)\mathbf{U}^{-T}(k-1) \quad (4.57)$$

$$\mathbf{w}(k) = \mathbf{w}(k-1) - \gamma(k)\mathbf{u}(k)e'(k) \quad (4.58)$$

onde o vetor de Kalman é agora expresso como $-\gamma(k)\mathbf{u}(k)$.

Foi observado em [15] que (4.57) implica na existência de uma matriz ortogonal $\mathbf{Q}_I(k)$ tal que

$$\mathbf{Q}_I(k) \begin{bmatrix} \mathbf{0}^T \\ \lambda^{-1/2}\mathbf{U}^{-T}(k-1) \end{bmatrix} = \begin{bmatrix} \mathbf{u}^T(k) \\ \mathbf{U}^{-T}(k) \end{bmatrix} \quad (4.59)$$

Isto pode ser verificado tomando-se a transposta de (4.59) e multiplicando-se por ela mesma, resultando na equação (4.57). Felizmente, $\mathbf{Q}_I(k)$ já é conhecida. Admitindo-se uma partição de $\mathbf{Q}_I(k)$ similar à usada em (4.18) e impondo ortonormalidade, nós podemos ver que (4.59) conduz a $\mathbf{Q}_I(k) = \mathbf{Q}_\theta(k)$.

Posto que sabemos que $\mathbf{Q}_\theta(k)$ é ortonormal, se pós-multiplicarmos esta matriz pela sua primeira linha transposta, teremos

$$\mathbf{Q}_\theta(k) \begin{bmatrix} \gamma(k) \\ -\gamma(k)\mathbf{a}(k) \end{bmatrix} = \begin{bmatrix} 1 \\ \mathbf{0} \end{bmatrix} \quad (4.60)$$

Combinando (4.59) (com $\mathbf{Q}_I(k) = \mathbf{Q}_\theta(k)$) e (4.60) (após dividirmos ambos termos por $\gamma(k)$) numa única equação, temos

$$\begin{bmatrix} 1/\gamma(k) & \mathbf{u}^T(k) \\ \mathbf{0} & \mathbf{U}^{-T}(k) \end{bmatrix} = \mathbf{Q}_\theta(k) \begin{bmatrix} 1 & \mathbf{0}^T \\ -\mathbf{a}(k) & \lambda^{-1/2}\mathbf{U}^{-T}(k-1) \end{bmatrix} \quad (4.61)$$

A equação (4.61) é uma relação chave ao algoritmo QR inverso. De modo a termos todas as equações necessárias a este algoritmo, analisemos agora a relação

entre os erros *a posteriori* e os erros *a priori*. Substituindo $\mathbf{Q}_\theta(k)$ por sua partição (dada em (4.18), (4.26), (4.27) e (4.28)) em (4.13), segue que

$$e(k) = \gamma(k)e_{q_1}(k) = \gamma^2(k)e'(k) \quad (4.62)$$

Seguindo procedimentos similares nos problemas de predição regressiva e progressiva, é possível mostrar que

$$e_b(k) = \gamma(k)e_{b_{q_1}}(k) = \gamma^2(k)e'_b(k) \quad (4.63)$$

$$e_f(k) = \gamma(k-1)e_{f_{q_1}}(k) = \gamma^2(k-1)e'_f(k) \quad (4.64)$$

As equações do algoritmo QR inverso são apresentadas na Tabela 4.2 enquanto que a descrição algorítmica detalhada pode ser encontrada em [15].

4.2 Classificação dos Algoritmos QR Rápidos

Do método de decomposição QR convencional ($O[N^2]$) [1, 2] vários algoritmos rápidos ($O[N]$) foram derivados [16]–[19]. Estes algoritmos podem ser classificados em termos do tipo de triangularização aplicado à matriz de dados de entrada (triangular superior ou inferior) e do tipo de vetor de erros (*a posteriori* ou *a priori*) envolvidos no processo de atualização. Ficou claro do procedimento de ortogonalização de Gram-Schmidt que uma triangularização superior (na notação usada nesta tese) envolve a atualização de erros de predição progressiva (forward) enquanto que uma triangularização inferior envolve a atualização de erros de predição regressiva (backward). Esta seção apresenta as equações destes algoritmos ditos rápidos bem como um algoritmo novo, designado por FQR_PRI_F, o qual é um algoritmo QR rápido usando triangularização superior (do fator Cholesky da matriz de correlação de dados) e atualizando erros de predição progressiva *a priori* (vetor $\mathbf{a}(k)$) ou “Fast QR based on *a PRIori* Forward errors”. A Tabela 4.3 apresenta a classificação e o modo pelo qual estes algoritmos serão designados daqui para frente.

Tabela 4.2: As equações do algoritmo QR inverso.

IQR
<p>for $k = 1, 2, \dots$</p> <p>{ Obtendo $\mathbf{a}(k)$:</p> $\mathbf{a}(k) = \lambda^{-1/2} \mathbf{U}^{-T}(k-1) \mathbf{x}(k)$ <p>Obtendo $\mathbf{Q}_\theta(k)$ e $\gamma(k)$:</p> $\begin{bmatrix} 1/\gamma(k) \\ \mathbf{0} \end{bmatrix} = \mathbf{Q}_\theta(k) \begin{bmatrix} 1 \\ -\mathbf{a}(k) \end{bmatrix}$ <p>Obtendo $\mathbf{u}(k)$ e atualizando $\mathbf{U}^{-T}(k)$:</p> $\begin{bmatrix} \mathbf{u}^T(k) \\ \mathbf{U}^{-T}(k) \end{bmatrix} = \mathbf{Q}_\theta(k) \begin{bmatrix} \mathbf{0}^T \\ \lambda^{-1/2} \mathbf{U}^{-T}(k-1) \end{bmatrix}$ <p>Obtendo $e(k)$:</p> $e(k) = [d(k) - \mathbf{x}^T(k) \mathbf{w}(k-1)] \gamma^2(k)$ <p>Atualizando o vetor de coeficientes:</p> $\mathbf{w}(k) = \mathbf{w}(k-1) - \mathbf{u}(k) e(k)$ <p>}</p>

Vale a pena mencionarmos que o algoritmo FQR_PRLB foi desenvolvido independentemente em [18] e [19] usando abordagens diferentes. A abordagem a ser usada aqui foi derivada [19] de conceitos usados no algoritmo QR inverso [15]. O mesmo algoritmo foi também derivado em [39] como uma extensão em treliça do algoritmo QR inverso [40].

Na derivação dos algoritmos QR rápidos, começamos por aplicar a decomposição QR aos problemas de predição regressiva e progressiva cujos erros de predição foram definidos em (4.29) e (4.32). Nosso objetivo é triangularizar $\mathbf{X}^{(N+2)}(k)$ das equações (4.30) e (4.33) de modo a obter $\mathbf{Q}^{(N+2)}(k)$ tal que

Tabela 4.3: Classificação dos algoritmos QR rápidos.

Tipo de Erro	Predição	
	Progressivo (Forward)	Regressivo (Backward)
<i>A Posteriori</i>	FQR_POS_F [16]	FQR_POS_B [17]
<i>A Priori</i>	novo FQR_PRI_F [20]	FQR_PRI_B [19, 18]

$$\mathbf{Q}^{(N+2)}(k)\mathbf{X}^{(N+2)}(k) = \begin{bmatrix} \mathbf{O} \\ \mathbf{U}^{(N+2)}(k) \end{bmatrix} \quad (4.65)$$

4.3 Algoritmos que Usam Triangularização Superior (Atualizando Erros de Predição Progressiva)

Derivaremos aqui os algoritmos FQR_POS_F [16] e o novo algoritmo FQR_PRI_F. Inicialmente, se pré-multiplicarmos o vetor desejado ponderado regressivo $\mathbf{d}_b(k)$ definido em (4.29) por $\mathbf{Q}(k)$ e usarmos (4.10), duas importantes relações seguirão

$$\|\mathbf{e}_b(k)\|^2 = e_{bq_1}^2(k) + \lambda \|\mathbf{e}_b(k-1)\|^2 \quad (4.66)$$

$$\begin{bmatrix} e_{bq_1}(k) \\ \mathbf{d}_{bq_2}(k) \end{bmatrix} = \mathbf{Q}_\theta(k) \begin{bmatrix} d_b(k) \\ \lambda^{1/2} \mathbf{d}_{bq_2}(k-1) \end{bmatrix} \quad (4.67)$$

onde $d_b(k) = x(k - N - 1)$.

No problema de predição regressiva, a triangularização como vista em (4.65) é obtida pelo uso de três matrizes, $\mathbf{Q}^{(N+2)}(k) = \mathbf{Q}'_b(k)\mathbf{Q}_b(k)\mathbf{Q}(k)$, onde $\mathbf{Q}_b(k)$ e $\mathbf{Q}'_b(k)$ são dois conjuntos de rotações de Givens aplicados para gerar, respectivamente, $\|\mathbf{e}_b(k)\|$ e $\|\mathbf{e}_b^{(0)}(k)\|$. Como resultado, temos

$$\begin{aligned}
\mathbf{U}^{(N+2)}(k) &= \mathbf{Q}'_{\theta b}(k) \begin{bmatrix} \mathbf{0}^T & \| \mathbf{e}_b(k) \| \\ \mathbf{U}(k) & \mathbf{d}_{bq_2}(k) \end{bmatrix} \\
&= \begin{bmatrix} \mathbf{z}^T(k) & \| \mathbf{e}_b^{(0)}(k) \| \\ \mathbf{R}(k) & \mathbf{0} \end{bmatrix}
\end{aligned} \tag{4.68}$$

onde $\mathbf{Q}'_{\theta b}(k)$ é uma submatriz de $\mathbf{Q}'_b(k)$, $[\mathbf{z}(k)\mathbf{R}^T(k)]^T$ é a parte esquerda de $\mathbf{U}^{(N+2)}(k)$ e $\| \mathbf{e}_b^{(0)}(k) \|$ é a norma do erro regressivo de um preditor com zero coeficiente.

No problema de predição progressiva, a pré-multiplicação do vetor desejado ponderado progressivo, $\mathbf{d}_f(k)$ como definido em (4.32), por $\begin{bmatrix} \mathbf{Q}(k-1) & \mathbf{0} \\ \mathbf{0}^T & 1 \end{bmatrix}$ e o uso de (4.10) conduzirão a outras duas relações importantes dadas por

$$\| \mathbf{e}_f(k) \|^2 = e_{f_{q_1}}^2(k) + \lambda \| \mathbf{e}_f(k-1) \|^2 \tag{4.69}$$

$$\begin{bmatrix} e_{f_{q_1}}(k) \\ \mathbf{d}_{f_{q_2}}(k) \end{bmatrix} = \mathbf{Q}_\theta(k-1) \begin{bmatrix} d_f(k) \\ \lambda^{1/2} \mathbf{d}_{f_{q_2}}(k-1) \end{bmatrix} \tag{4.70}$$

onde $d_f(k) = x(k)$.

A triangularização superior de $\mathbf{U}^{(N+2)}(k)$ no problema de predição progressiva é implementada ao pré-multiplicarmos $\mathbf{e}_f(k)$ pelo produto $\mathbf{Q}_f(k) \begin{bmatrix} \mathbf{Q}(k-1) & \mathbf{0} \\ \mathbf{0}^T & 1 \end{bmatrix}$, onde $\mathbf{Q}_f(k)$ é um conjunto de rotações de Givens gerando $\| \mathbf{e}_f(k) \|$ pela eliminação dos primeiros $k-N$ elementos do vetor de dados rotacionado do preditor progressivo. O resultado é

$$\mathbf{U}^{(N+2)}(k) = \begin{bmatrix} \mathbf{d}_{f_{q_2}}(k) & \mathbf{U}(k-1) \\ \| \mathbf{e}_f(k) \| & \mathbf{0}^T \end{bmatrix} \tag{4.71}$$

Trabalhando com dimensões não crescentes, é fácil mostrar que [2]

$$\mathbf{Q}_\theta^{(N+2)}(k) = \mathbf{Q}_{\theta f}(k) \begin{bmatrix} \mathbf{Q}_\theta(k-1) & \mathbf{0} \\ \mathbf{0}^T & 1 \end{bmatrix} \tag{4.72}$$

onde $\mathbf{Q}_{\theta_f}(k)$ é uma rotação de Givens simples gerando $\|\mathbf{e}_f(k)\|$ como em (4.69).

Se tomarmos as inversas de (4.68) e (4.71), os resultados são

$$\begin{aligned} [\mathbf{U}^{(N+2)}(k)]^{-1} &= \begin{bmatrix} \mathbf{0} & \mathbf{R}^{-1}(k) \\ \frac{1}{\|\mathbf{e}_b^{(0)}(k)\|} & \frac{-\mathbf{z}^T(k)\mathbf{R}^{-1}(k)}{\|\mathbf{e}_b^{(0)}(k)\|} \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{0}^T & \frac{1}{\|\mathbf{e}_f(k)\|} \\ \mathbf{U}^{-1}(k-1) & \frac{-\mathbf{U}^{-1}(k-1)\mathbf{d}_{fq_2}(k)}{\|\mathbf{e}_f(k)\|} \end{bmatrix} \end{aligned} \quad (4.73)$$

Nós podemos usar as expressões de $[\mathbf{U}^{(N+2)}(k)]^{-1}$ dada em (4.73) para obter os vetores $\mathbf{f}^{(N+2)}(k+1)$ e $\mathbf{a}^{(N+2)}(k+1)$. A escolha de um destes vetores determinará o algoritmo: atualizando $\mathbf{f}(k)$ (erros progressivos *a posteriori*) conduzirá ao algoritmo FQR_POS_F [16] e atualizando $\mathbf{a}(k)$ (erros progressivos *a priori*) conduzirá ao algoritmo FQR_PRI_F [20].

4.3.1 O Algoritmo FQR_POS_F

No algoritmo FQR_POS_F, o vetor $\mathbf{f}^{(N+2)}(k+1) = [\mathbf{U}^{(N+2)}(k+1)]^{-T} \mathbf{x}^{(N+2)}(k+1)$ é expresso em termos das relações obtidas nos problemas de predição progressivo e regressivo. Usaremos, inicialmente, a expressão para $[\mathbf{U}^{(N+2)}(k)]^{-1}$ em (4.73) que vem da predição regressiva avaliada no instante $k+1$ para calcular $\mathbf{f}^{(N+2)}(k+1)$. Neste caso, substituímos $\mathbf{x}^{(N+2)}(k+1)$ por $[\mathbf{x}^T(k+1) \ x(k-N)]^T$ e então pré-multiplicamos o resultado por $\mathbf{Q}'_{\theta_b}(k+1)$. O resultado final após algumas manipulações algébricas (usando a equação (4.68) para ajudar na simplificação da expressão) é

$$\mathbf{f}^{(N+2)}(k+1) = \mathbf{Q}'_{\theta_b}(k+1) \begin{bmatrix} \frac{e_b(k+1)}{\|\mathbf{e}_b(k+1)\|} \\ \mathbf{f}(k+1) \end{bmatrix} \quad (4.74)$$

Usando a outra expressão para $[\mathbf{U}^{(N+2)}(k)]^{-1}$ (da predição progressiva) no instante $k+1$ e substituindo $\mathbf{x}^{(N+2)}(k+1)$ por $[x(k+1) \ \mathbf{x}^T(k)]^T$, temos

$$\mathbf{f}^{(N+2)}(k+1) = \begin{bmatrix} \mathbf{f}(k) \\ \frac{e_f(k+1)}{\|\mathbf{e}_f(k+1)\|} \end{bmatrix} \quad (4.75)$$

Combinando (4.74) e (4.75), temos uma expressão para atualizar $\mathbf{f}(k)$ dada por

$$\begin{bmatrix} \frac{e_b(k+1)}{\|\mathbf{e}_b(k+1)\|} \\ \mathbf{f}(k+1) \end{bmatrix} = \mathbf{Q}'_{\theta_b T}(k+1) \begin{bmatrix} \mathbf{f}(k) \\ \frac{e_f(k+1)}{\|\mathbf{e}_f(k+1)\|} \end{bmatrix} \quad (4.76)$$

Uma vez que temos $\mathbf{f}(k+1)$, podemos encontrar os ângulos de $\mathbf{Q}_\theta(k+1)$ pós-multiplicando esta matriz pelo vetor $[1 \ 0 \ \dots \ 0]^T$. De (4.18), podemos ver que o resultado é

$$\mathbf{Q}_\theta(k+1) \begin{bmatrix} 1 \\ \mathbf{0} \end{bmatrix} = \begin{bmatrix} \gamma(k+1) \\ \mathbf{f}(k+1) \end{bmatrix} \quad (4.77)$$

Entretanto, as quantidades requeridas para computar os ângulos de $\mathbf{Q}'_{\theta_b}(k+1)$ não estão disponíveis no instante k , e uma estratégia especial torna-se necessária. A matriz atualizada $\mathbf{Q}'_{\theta_b}(k+1)$ é obtida [2, 41] com o uso do vetor $\mathbf{c}(k+1)$ definido por

$$\begin{aligned} \mathbf{c}(k+1) &= \hat{\mathbf{Q}}_\theta^{(N+2)}(k+1) \mathbf{Q}'_{\theta_b}(k) \begin{bmatrix} 1 \\ \mathbf{0} \end{bmatrix} \\ &= \mathbf{Q}'_{\theta_b}(k+1) \begin{bmatrix} b \\ \mathbf{0} \end{bmatrix} \end{aligned} \quad (4.78)$$

A submatriz $\hat{\mathbf{Q}}_\theta^{(N+2)}(k+1)$ consistindo dos últimos $(N+2) \times (N+2)$ elementos de $\mathbf{Q}_\theta^{(N+2)}(k+1)$ está disponível de (4.72) (predição progressiva) e b não necessita ser explicitamente calculado para obter-se os ângulos de θ'_b .

Finalmente, a estimação do processo conjunto é calculada com (4.13) e (4.20), e as equações do algoritmo FQR_POS_F estão apresentadas na Tabela 4.4. Uma descrição detalhada deste algoritmo é encontrada no Apêndice C.

4.3.2 O Novo Algoritmo FQR_PRI_F

Expressando $\mathbf{a}^{(N+2)}(k+1) = [\mathbf{U}^{(N+2)}(k)]^{-T} \mathbf{x}^{(N+2)}(k+1)/\sqrt{\lambda}$ em termos das matrizes em (4.73) e pré-multiplicando aquela que é proveniente do problema de predição regressiva por $\mathbf{Q}'_{\theta_b}(k)\mathbf{Q}'_{\theta_b}(k)$ conduz a

$$\begin{bmatrix} \frac{e'_b(k+1)}{\sqrt{\lambda}\|\mathbf{e}_b(k)\|} \\ \mathbf{a}(k+1) \end{bmatrix} = \mathbf{Q}'_{\theta_b}(k)^T \begin{bmatrix} \mathbf{a}(k) \\ \frac{e'_f(k+1)}{\sqrt{\lambda}\|\mathbf{e}_f(k)\|} \end{bmatrix} \quad (4.79)$$

Uma vez que temos $\mathbf{a}(k+1)$, os ângulos de $\mathbf{Q}_\theta(k+1)$ são encontrados por meio da seguinte relação obtida ao pós-multiplicarmos $\mathbf{Q}_\theta^T(k+1)$ pelo vetor $[1 \ 0 \ \dots \ 0]^T$.

$$\begin{bmatrix} 1/\gamma(k+1) \\ \mathbf{0} \end{bmatrix} = \mathbf{Q}_\theta(k+1) \begin{bmatrix} 1 \\ -\mathbf{a}(k+1) \end{bmatrix} \quad (4.80)$$

Uma vez que os ângulos de $\mathbf{Q}'_{\theta_b}(k+1)$ podem ser atualizados pelo mesmo procedimento usado no algoritmo FQR_POS_F, já temos todas as equações necessárias do novo algoritmo QR-RLS rápido apresentado na Tabela 4.5. A descrição detalhada deste algoritmo é encontrada no Apêndice C.

4.4 Algoritmos que Usam Triangularização Inferior (Atualizando Erros de Predição Regressiva)

Seguindo passos similares como na triangularização superior, é possível obter a matriz triangular inferior $\mathbf{U}^{(N+2)}(k)$ dos problemas de predição progressivo e regressivo.

No problema de predição regressiva, $\mathbf{U}^{(N+2)}(k)$ triangular inferior é obtida com o uso de $\mathbf{Q}^{(N+2)}(k) = \mathbf{Q}_b(k)\mathbf{Q}(k)$, onde $\mathbf{Q}_b(k)$ é um conjunto de rotações de Givens aplicado para gerar $\|\mathbf{e}_b(k)\|$. O fator Cholesky resultante é

$$\mathbf{U}^{(N+2)}(k) = \begin{bmatrix} \mathbf{0}^T & \| \mathbf{e}_b(k) \| \\ \mathbf{U}(k) & \mathbf{d}_{bq_2}(k) \end{bmatrix} \quad (4.81)$$

Por outro lado, no problema de regressão progressiva, a matriz triangular inferior de $\mathbf{U}^{(N+2)}(k)$ é implementada pela pré-multiplicação de $\mathbf{e}_f(k)$ pelo produto $\mathbf{Q}'_f(k)\mathbf{Q}_f(k)$ $\begin{bmatrix} \mathbf{Q}(k-1) & \mathbf{0} \\ \mathbf{0}^T & 1 \end{bmatrix}$, onde $\mathbf{Q}_f(k)$ e $\mathbf{Q}'_f(k)$ são dois conjuntos de rotações de Givens gerando $\| \mathbf{e}_f(k) \|$ e $\| \mathbf{e}_f^{(0)}(k) \|$, respectivamente. A expressão resultante é

$$\mathbf{U}^{(N+2)}(k) = \mathbf{Q}'_{\theta_f}(k) \begin{bmatrix} \mathbf{d}_{fq_2}(k) & \mathbf{U}(k-1) \\ \| \mathbf{e}_f(k) \| & \mathbf{0}^T \end{bmatrix} = \begin{bmatrix} \mathbf{0} & \mathbf{R}(k) \\ \| \mathbf{e}_f^{(0)}(k) \| & \mathbf{z}^T(k) \end{bmatrix} \quad (4.82)$$

onde $[\mathbf{R}^T(k) \mathbf{z}(k)]^T$ é a parte direita de $\mathbf{U}^{(N+2)}(k)$. Mantendo em mente que (4.66), (4.67), (4.69) e (4.70) são ainda válidas, $\| \mathbf{e}_f(k) \|$ podem ser computadas recursivamente usando (4.69).

Tomando a inversa de (4.81) e (4.82) nós temos os seguintes resultados

$$\begin{aligned} [\mathbf{U}^{(N+2)}(k)]^{-1} &= \begin{bmatrix} \frac{-\mathbf{U}^{-1}(k)\mathbf{d}_{bq_2}(k)}{\| \mathbf{e}_b(k) \|} & \mathbf{U}^{-1}(k) \\ \frac{1}{\| \mathbf{e}_b(k) \|} & \mathbf{0}^T \end{bmatrix} \\ &= \begin{bmatrix} \frac{-\mathbf{z}^T(k)\mathbf{R}^{-1}(k)}{\| \mathbf{e}_f^{(0)}(k) \|} & \frac{1}{\| \mathbf{e}_f^{(0)}(k) \|} \\ \mathbf{R}^{-1}(k) & \mathbf{0} \end{bmatrix} \end{aligned} \quad (4.83)$$

Com os resultados obtidos em (4.83), podemos uma vez mais expressar os vetores $\mathbf{f}^{(N+2)}(k+1)$ e $\mathbf{a}^{(N+2)}(k+1)$ em termos das partições de $[\mathbf{U}^{(N+2)}(k+1)]^{-1}$. Se atualizarmos $\mathbf{f}(k)$, o algoritmo resultante será o FQR_POS_B enquanto que se atualizarmos $\mathbf{a}(k)$ teremos o algoritmo FQR_PRI_B.

4.4.1 O Algoritmo FQR_POS_B

Expressando $\mathbf{f}^{(N+2)}(k+1) = [\mathbf{U}^{(N+2)}(k+1)]^{-T} \mathbf{x}^{(N+2)}(k+1)$ em termos das matrizes em (4.83) e pré-multiplicando aquela que é proveniente do problema de predição progressiva por $\mathbf{Q}'_{\theta_f}(k+1)\mathbf{Q}'_{\theta_f T}(k+1)$ conduz a

$$\begin{bmatrix} \frac{e_b(k+1)}{\|\mathbf{e}_b(k+1)\|} \\ \mathbf{f}(k+1) \end{bmatrix} = \mathbf{Q}'_{\theta_f}(k+1) \begin{bmatrix} \mathbf{f}(k) \\ \frac{e_f(k+1)}{\|\mathbf{e}_f(k+1)\|} \end{bmatrix} \quad (4.84)$$

Durante a derivação de (4.84), foi observado que o último elemento de $\mathbf{f}(k+1)$ é $\frac{x(k+1)}{\|\mathbf{e}_f^{(0)}(k+1)\|}$. O termo $\frac{e_f(k+1)}{\|\mathbf{e}_f(k+1)\|}$ pode ser calculado como $\gamma(k)\sin\theta_f(k+1)$ onde $\sin\theta_f(k+1) = \frac{e_{fq}(k+1)}{\|\mathbf{e}_f(k+1)\|}$ é o seno do ângulo da matriz de rotação $\mathbf{Q}_f(k+1)$.

Um vez tendo $\mathbf{f}(k+1)$, encontramos $\mathbf{Q}_\theta(k+1)$ com a mesma relação usada nos algoritmos de triangularização superior, ou seja, (4.77). Além disto, a estimação conjunta é realizada da mesma maneira e as equações do algoritmo estão apresentadas na Tabela 4.6. As descrições detalhadas de duas versões ligeiramente diferentes deste mesmo algoritmo são encontradas no Apêndice C.

4.4.2 O Algoritmo FQR_PRI_B

Este último algoritmo desta família é obtido ao expressarmos o vetor $\mathbf{a}^{(N+2)}(k+1) = [\mathbf{U}^{(N+2)}(k)]^{-T} \mathbf{x}^{(N+2)}(k+1)/\sqrt{\lambda}$ em termos das matrizes em (4.83) e pré-multiplicando aquela que é oriunda do problema de predição progressiva por $\mathbf{Q}'_{\theta_f}(k)\mathbf{Q}'_{\theta_f T}(k)$. A equação de atualização é

$$\begin{bmatrix} \frac{e'_b(k+1)}{\sqrt{\lambda}\|\mathbf{e}_b(k)\|} \\ \mathbf{a}(k+1) \end{bmatrix} = \mathbf{Q}'_{\theta_f}(k) \begin{bmatrix} \mathbf{a}(k) \\ \frac{e'_f(k+1)}{\sqrt{\lambda}\|\mathbf{e}_f(k)\|} \end{bmatrix} \quad (4.85)$$

É novamente importante ressaltar que, durante a derivação, foi observado que o último elemento de $\mathbf{a}(k+1)$ em (4.85) já é conhecido como sendo igual a $\frac{x(k+1)}{\sqrt{\lambda}\|\mathbf{e}_f^{(0)}(k)\|}$. Este fato leva a duas versões ligeiramente diferentes do mesmo algoritmo.

Mais uma vez, se tomarmos $\mathbf{a}(k + 1)$, podemos encontrar $\mathbf{Q}_\theta(k + 1)$ usando (4.80) e a estimação conjunta é feita com (4.13) e (4.20). As equações do algoritmo FQR_PRI_B são apresentadas na Tabela 4.7. As descrições detalhadas deste algoritmo são encontradas no Apêndice C.

4.5 Conclusões

Este capítulo derivou um novo algoritmo chamado FQR_PRI_F ou “fast QR decomposition RLS algorithm using *a priori* forward errors” (baseado na triangularização superior da matriz de entrada de dados de acordo com a notação usada aqui) e suas relações com outros membros da família de algoritmos QR rápidos. Uma estrutura didática foi usada para classificar os quatro algoritmos QR rápidos desta família e suas derivações bem como suas descrições simples (somente equações matriciais) e detalhadas foram apresentadas.

Em termos de complexidade computacional, a Tabela 4.8 mostra as comparações entre os quatro algoritmos de acordo com as descrições detalhadas no Apêndice C. Note que $p = N + 1$ é o número de coeficientes do filtro adaptativo.

Finalmente, é importante ressaltar que os algoritmos QR rápidos com triangularização inferior da matriz de entrada de dados ou, equivalentemente, atualizando erros de predições regressivas são de complexidades mínimas e apresentam estabilidade retrógrada sob excitação persistente [17, 40].

Tabela 4.4: As equações do algoritmo FQR_POS_F.

FQR_POS_F
<p>for $k = 1, 2, \dots$</p> <p>{ Obtendo $\ \mathbf{e}_f(k+1) \$:</p> $\begin{bmatrix} e_{fq_1}(k+1) \\ \mathbf{d}_{fq_2}(k+1) \end{bmatrix} = \mathbf{Q}_\theta(k) \begin{bmatrix} x(k+1) \\ \lambda^{1/2} \mathbf{d}_{fq_2}(k) \end{bmatrix}$ $\ \mathbf{e}_f(k+1) \ = \sqrt{e_{fq_1}^2(k+1) + \lambda \ \mathbf{e}_f(k) \ ^2}$ <p>Obtendo $\mathbf{Q}_{\theta_f}(k+1)$:</p> $\cos\theta_f(k+1) = \lambda^{1/2} \ \mathbf{e}_f(k) \ / \ \mathbf{e}_f(k+1) \ $ $\sin\theta_f(k+1) = e_{fq_1}(k+1) / \ \mathbf{e}_f(k+1) \ $ <p>Obtendo $\mathbf{c}(k+1)$:</p> $\mathbf{Q}_\theta^{(N+2)}(k+1) = \mathbf{Q}_{\theta_f}(k+1) \begin{bmatrix} \mathbf{Q}_\theta(k) & \mathbf{0} \\ \mathbf{0}^T & 1 \end{bmatrix}$ $\hat{\mathbf{Q}}_\theta^{(N+2)}(k+1) = \text{últimos } (N+2) \times (N+2) \text{ elementos de } \mathbf{Q}_\theta^{(N+2)}(k+1)$ $\mathbf{c}(k+1) = \hat{\mathbf{Q}}_\theta^{(N+2)}(k+1) \mathbf{Q}'_{\theta b}(k) \begin{bmatrix} 1 \\ \mathbf{0} \end{bmatrix}$ <p>Obtendo $\mathbf{Q}'_{\theta b}(k+1)$:</p> $\begin{bmatrix} b \\ \mathbf{0} \end{bmatrix} = \mathbf{Q}'_{\theta b}{}^T(k+1) \mathbf{c}(k+1)$ <p>Obtendo $\mathbf{f}(k+1)$:</p> $\begin{bmatrix} \frac{e_b(k+1)}{\ \mathbf{e}_b(k+1) \ } \\ \mathbf{f}(k+1) \end{bmatrix} = \mathbf{Q}'_{\theta b}{}^T(k+1) \begin{bmatrix} \mathbf{f}(k) \\ \frac{e_f(k+1)}{\ \mathbf{e}_f(k+1) \ } \end{bmatrix}$ <p>Obtendo $\mathbf{Q}_\theta(k+1)$:</p> $\begin{bmatrix} 1 \\ \mathbf{0} \end{bmatrix} = \mathbf{Q}_\theta^T(k+1) \begin{bmatrix} \gamma(k+1) \\ \mathbf{f}(k+1) \end{bmatrix}$ <p>Estimação Conjunta:</p> $\begin{bmatrix} e_{q_1}(k+1) \\ \mathbf{d}_{q_2}(k+1) \end{bmatrix} = \mathbf{Q}_\theta(k+1) \begin{bmatrix} d(k+1) \\ \lambda^{1/2} \mathbf{d}_{q_2}(k) \end{bmatrix}$ $e(k+1) = e_{q_1}(k+1) \gamma(k+1)$ <p>}</p>

Tabela 4.5: As equações do algoritmo FQR_PRLF.

FQR_PRLF
<p>for $k = 1, 2, \dots$</p> <p>{ Obtendo $e'_f(k+1)$:</p> $\begin{bmatrix} e_{fq_1}(k+1) \\ \mathbf{d}_{fq_2}(k+1) \end{bmatrix} = \mathbf{Q}_\theta(k) \begin{bmatrix} x(k+1) \\ \lambda^{1/2} \mathbf{d}_{fq_2}(k) \end{bmatrix}$ <p>$e'_f(k+1) = e_{fq_1}(k+1)/\gamma(k)$</p> <p>Obtendo $\mathbf{a}(k+1)$:</p> $\begin{bmatrix} \frac{e'_b(k+1)}{\sqrt{\lambda} \ \mathbf{e}_b(k)\ } \\ \mathbf{a}(k+1) \end{bmatrix} = \mathbf{Q}'_{\theta b}(k) \begin{bmatrix} \mathbf{a}(k) \\ \frac{e'_f(k+1)}{\sqrt{\lambda} \ \mathbf{e}_f(k)\ } \end{bmatrix}$ <p>Obtendo $\mathbf{Q}_{\theta f}(k+1)$:</p> $\ \mathbf{e}_f(k+1)\ = \sqrt{e_{fq_1}^2(k+1) + \lambda \ \mathbf{e}_f(k)\ ^2}$ $\cos\theta_f(k+1) = \lambda^{1/2} \ \mathbf{e}_f(k)\ / \ \mathbf{e}_f(k+1)\ $ $\sin\theta_f(k+1) = e_{fq_1}(k+1) / \ \mathbf{e}_f(k+1)\ $ <p>Obtendo $\mathbf{c}(k+1)$:</p> $\mathbf{Q}_\theta^{(N+2)}(k+1) = \mathbf{Q}_{\theta f}(k+1) \begin{bmatrix} \mathbf{Q}_\theta(k) & \mathbf{0} \\ \mathbf{0}^T & 1 \end{bmatrix}$ <p>$\hat{\mathbf{Q}}_\theta^{(N+2)}(k+1)$ = últimos $(N+2) \times (N+2)$ elementos de $\mathbf{Q}_\theta^{(N+2)}(k+1)$</p> $\mathbf{c}(k+1) = \hat{\mathbf{Q}}_\theta^{(N+2)}(k+1) \mathbf{Q}'_{\theta b}(k) \begin{bmatrix} 1 \\ \mathbf{0} \end{bmatrix}$ <p>Obtendo $\mathbf{Q}'_{\theta b}(k+1)$:</p> $\begin{bmatrix} b \\ \mathbf{0} \end{bmatrix} = \mathbf{Q}'_{\theta b}(k+1) \mathbf{c}(k+1)$ <p>Obtendo $\mathbf{Q}_\theta(k+1)$:</p> $\begin{bmatrix} 1/\gamma(k+1) \\ \mathbf{0} \end{bmatrix} = \mathbf{Q}_\theta(k+1) \begin{bmatrix} 1 \\ -\mathbf{a}(k+1) \end{bmatrix}$ <p>Estimação Conjunta:</p> $\begin{bmatrix} e_{q_1}(k+1) \\ \mathbf{d}_{q_2}(k+1) \end{bmatrix} = \mathbf{Q}_\theta(k+1) \begin{bmatrix} d(k+1) \\ \lambda^{1/2} \mathbf{d}_{q_2}(k) \end{bmatrix}$ <p>$e(k+1) = e_{q_1}(k+1)\gamma(k+1)$</p> <p>}</p>

Tabela 4.6: As equações do algoritmo FQR_POS_B.

FQR_POS_B
<p>for $k = 1, 2, \dots$</p> <p>{ Obtendo $\mathbf{d}_{fq_2}(k+1)$:</p> $\begin{bmatrix} e_{fq_1}(k+1) \\ \mathbf{d}_{fq_2}(k+1) \end{bmatrix} = \mathbf{Q}_\theta(k) \begin{bmatrix} x(k+1) \\ \lambda^{1/2} \mathbf{d}_{fq_2}(k) \end{bmatrix}$ <p>Obtendo $\ \mathbf{e}_f(k+1)\$:</p> $\ \mathbf{e}_f(k+1)\ = \sqrt{e_{fq_1}^2(k+1) + \lambda \ \mathbf{e}_f(k)\ ^2}$ <p>Obtendo $\mathbf{Q}'_{\theta_f}(k+1)$:</p> $\begin{bmatrix} \mathbf{0} \\ \ \mathbf{e}_f^{(0)}(k+1)\ \end{bmatrix} = \mathbf{Q}'_{\theta_f}(k+1) \begin{bmatrix} \mathbf{d}_{fq_2}(k+1) \\ \ \mathbf{e}_f(k+1)\ \end{bmatrix}$ <p>Obtendo $\mathbf{f}(k+1)$:</p> $\begin{bmatrix} \frac{e_b(k+1)}{\ \mathbf{e}_b(k+1)\ } \\ \mathbf{f}(k+1) \end{bmatrix} = \mathbf{Q}'_{\theta_f}(k+1) \begin{bmatrix} \mathbf{f}(k) \\ \frac{e_f(k+1)}{\ \mathbf{e}_f(k+1)\ } \end{bmatrix}$ <p>Obtendo $\mathbf{Q}_\theta(k+1)$:</p> $\begin{bmatrix} 1 \\ \mathbf{0} \end{bmatrix} = \mathbf{Q}_\theta^T(k+1) \begin{bmatrix} \gamma(k+1) \\ \mathbf{f}(k+1) \end{bmatrix}$ <p>Estimação Conjunta:</p> $\begin{bmatrix} e_{q_1}(k+1) \\ \mathbf{d}_{q_2}(k+1) \end{bmatrix} = \mathbf{Q}_\theta(k+1) \begin{bmatrix} d(k+1) \\ \lambda^{1/2} \mathbf{d}_{q_2}(k) \end{bmatrix}$ <p>$e(k+1) = e_{q_1}(k+1)\gamma(k+1)$</p> <p>}</p>

Tabela 4.7: As equações do algoritmo FQR_PRLB.

FQR_PRLB
<p>for $k = 1, 2, \dots$</p> <p>{ Obtendo $\mathbf{d}_{fq_2}(k + 1)$:</p> $\begin{bmatrix} e_{fq_1}(k + 1) \\ \mathbf{d}_{fq_2}(k + 1) \end{bmatrix} = \mathbf{Q}_\theta(k) \begin{bmatrix} x(k + 1) \\ \lambda^{1/2} \mathbf{d}_{fq_2}(k) \end{bmatrix}$ <p>Obtendo $\mathbf{a}(k + 1)$:</p> $\begin{bmatrix} \frac{e'_b(k+1)}{\sqrt{\lambda} \ \mathbf{e}_b(k)\ } \\ \mathbf{a}(k + 1) \end{bmatrix} = \mathbf{Q}'_{\theta_f}(k) \begin{bmatrix} \mathbf{a}(k) \\ \frac{e'_f(k+1)}{\sqrt{\lambda} \ \mathbf{e}_f(k)\ } \end{bmatrix}$ <p>Obtendo $\ \mathbf{e}_f(k + 1)\$:</p> $\ \mathbf{e}_f(k + 1)\ = \sqrt{e_{fq_1}^2(k + 1) + \lambda \ \mathbf{e}_f(k)\ ^2}$ <p>Obtendo $\mathbf{Q}'_{\theta_f}(k + 1)$:</p> $\begin{bmatrix} \mathbf{0} \\ \ \mathbf{e}_f^{(0)}(k + 1)\ \end{bmatrix} = \mathbf{Q}'_{\theta_f}(k + 1) \begin{bmatrix} \mathbf{d}_{fq_2}(k + 1) \\ \ \mathbf{e}_f(k + 1)\ \end{bmatrix}$ <p>Obtendo $\mathbf{Q}_\theta(k + 1)$:</p> $\begin{bmatrix} 1/\gamma(k + 1) \\ \mathbf{0} \end{bmatrix} = \mathbf{Q}_\theta(k + 1) \begin{bmatrix} 1 \\ -\mathbf{a}(k + 1) \end{bmatrix}$ <p>Estimação Conjunta:</p> $\begin{bmatrix} e_{q_1}(k + 1) \\ \mathbf{d}_{q_2}(k + 1) \end{bmatrix} = \mathbf{Q}_\theta(k + 1) \begin{bmatrix} d(k + 1) \\ \lambda^{1/2} \mathbf{d}_{q_2}(k) \end{bmatrix}$ <p>$e(k + 1) = e_{q_1}(k + 1)\gamma(k + 1)$</p> <p>}</p>

Tabela 4.8: Comparação de complexidade computacional.

ALGORITMO	SOMAS	MULT.	DIV.	RAÍZES
FQR_POS_F	$10p+3$	$26p+10$	$3p+2$	$2p+1$
FQR_PRI_F	$10p+3$	$26p+11$	$4p+4$	$2p+1$
FQR_POS_B (VERSÃO 1)	$8p+1$	$19p+4$	$4p+1$	$2p+1$
FQR_POS_B (VERSÃO 2)	$8p+1$	$20p+5$	$3p+1$	$2p+1$
FQR_PRI_B (VERSÃO 1)	$8p-1$	$19p+2$	$5p+1$	$2p+1$
FQR_PRI_B (VERSÃO 2)	$8p+1$	$20p+6$	$4p+2$	$2p+1$

Capítulo 5

As Versões em Treliça dos Algoritmos QR Rápidos

5.1 Introdução

Os algoritmos QR rápidos empregando triangularização inferior da matriz de dados de entrada são conhecidos como algoritmos dos mínimos quadrados híbridos QR-treliça ou “hybrid QR-lattice least squares algorithms”. Ficou claro do capítulo anterior que estes algoritmos podem atualizar os erros de predição regressiva *a posteriori* ou *a priori*. Além disto, eles são conhecidos por seus comportamentos numéricos robustos e por sua complexidade mínima mas não possuem a propriedade de implementação conhecida como “pipeline” dos algoritmos em treliça (ou “lattice”).

O ponto principal deste capítulo é a apresentação das versões em treliça dos algoritmos QR rápidos usando atualização de erros de predição regressiva *a posteriori* e *a priori* ou algoritmos FQR_POS_B e FQR_PRI_B de acordo com nossa classificação. As equações destes algoritmos são combinadas numa maneira recursiva na ordem tal que elas podem ser representadas com algoritmos em treliça com um “loop” único de ordem crescente. Estas versões em treliça podem então ser implementadas com uma estrutura modular que utiliza um único tipo de estágio de treliça para cada algoritmo.

Antes da derivação destas versões “lattice”, especifiquemos na Tabela 5.1 o significado de cada variável usada em ambos algoritmos.

Vale a pena mencionarmos neste ponto que uma variável sem sobrescrito implica numa grandeza de N -ésima ordem ou, de modo equivalente, correspondente a uma filtragem com $N + 1$ coeficientes. A título de exemplo, tomemos a norma da energia progressiva (forward): $\| \mathbf{e}_f(k) \| = \| \mathbf{e}_f^{(N+1)}(k) \|$.

Tabela 5.1: Variáveis usadas nos algoritmos FQR_POS_B e FQR_PRI_B.

$\mathbf{d}_{fq}(k)$: vetor desejado progressivo rotacionado
$\mathbf{d}_{fq2}(k)$: últimos $N+1$ elementos de $\mathbf{d}_{fq}(k)$
$\mathbf{e}_f(k)$: vetor de erros progressivos
$\ \mathbf{e}_f(k) \ $: norma de $\mathbf{e}_f(k)$
$\mathbf{e}_{fq}(k)$: $\mathbf{e}_f(k)$ rotacionado
$e_{fq1}(k)$: primeiro elemento de $\mathbf{e}_{fq}(k)$
$\mathbf{Q}_\theta(k)$: matriz de Givens (atualiza o fator Cholesky)
$x(k)$: sinal de entrada
λ	: fator de esquecimento
$\mathbf{Q}'_{\theta f}(k + 1)$: matriz de Givens (elimina $\mathbf{d}_{fq2}(k + 1)$ em (E3))
$\ \mathbf{e}_f^{(0)}(k) \ $: norma de $\mathbf{e}_f(k)$ num caso de 0 coeficiente
$\mathbf{f}(k)$: erros normalizados <i>a posteriori</i>
$\mathbf{a}(k)$: erros normalizados <i>a priori</i>
$e_b(k)$: erro de predição regressiva
$\ \mathbf{e}_b(k) \ $: norma de $e_b(k)$
$e_f(k)$: erro de predição progressiva <i>a posteriori</i>
$e'_f(k)$: erro de predição progressiva <i>a priori</i>
$e_b(k)$: erro de predição regressiva <i>a posteriori</i>
$e'_b(k)$: erro de predição regressiva <i>a priori</i>
$\gamma(k)$: produto de cosenos dos ângulos de $\mathbf{Q}_\theta(k)$
$\mathbf{e}_q(k)$: vetor de erro rotacionado

(Continuação da Tabela 5.1)

$e_{q_1}(k)$: primeiro elemento de $\mathbf{e}_q(k)$
$\mathbf{d}_q(k)$: vetor desejado rotacionado
$\mathbf{d}_{q_2}(k)$: últimos $N+1$ elementos de $\mathbf{d}_q(k)$
$d(k)$: sinal desejado
$e(k)$: erro de saída

5.2 Derivando as Versões em Treliça

As variáveis internas encontradas nos algoritmos QR rápidos são intimamente relacionadas àquelas encontradas nos algoritmos em treliça convencionais. Esta foi de fato a abordagem usada em [17, 18] para desenvolver estes algoritmos e as implicações estão bem explicadas nestas duas referências. Como apontado em [17], dentro desta ótica a solução para o problema de identificação de parâmetros foi inicialmente abordada usando os algoritmos QR rápidos. O trabalho de [37] ressalta o fato de que $\sin\theta'_f(k)$ e $\sin\theta'_f(k-1)$ representam os coeficientes de reflexão de algoritmos RLS em treliça normalizados (*a priori* e *a posteriori*).

Por outro lado, a idéia principal por trás da geração de uma versão em treliça (ou *totalmente* em treliça como no nosso caso) é a fusão de suas equações usando uma atualização de ordem ao invés de variáveis com ordens fixas. Isto pode ser feito quando resultados parciais possuem esta propriedade de atualização em ordem. Este é de fato o caso para os algoritmos do tipo triangularização inferior visto que suas variáveis internas estão sincronizadas no instante k ou $k-1$ (somente atualização de ordem). A mesma facilidade em obter-se versões em treliça não é observada nos algoritmos que utilizam triangularização superior (FQR_POS_F e FQR_PRI_F) uma vez que os erros normalizados presentes na matriz ortogonal $\mathbf{Q}_\theta(k)$ são de diferentes ordens **em diferentes instantes de tempo** (atualização de ordem e tempo).

Em seguida mostramos como combinar as equações do algoritmo FQR_POS_B de modo a obter sua versão em treliça. Começando por (4.70), rescrevemos esta equação

para o instante $k + 1$, com uma forma explícita de $\mathbf{Q}_\theta(k)$ em termos do produtos de $N + 1$ rotações de Givens $\mathbf{Q}_{\theta_i}(k)$ — veja (4.15) — e com $e_{f_{q_1}}^{(0)}(k + 1) = x(k + 1)$.

$$\begin{aligned} \begin{bmatrix} e_{f_{q_1}}(k + 1) \\ d_{fq_{2_1}}(k + 1) \\ \vdots \\ d_{fq_{2_{N+1}}}(k + 1) \end{bmatrix} &= \begin{bmatrix} \cos\theta_N(k) & -\sin\theta_N(k) & \mathbf{0}^T \\ \sin\theta_N(k) & \cos\theta_N(k) & \mathbf{0}^T \\ \mathbf{0} & \mathbf{0} & \mathbf{I}_N \end{bmatrix} \cdots \\ \cdots \begin{bmatrix} \cos\theta_0(k) & \mathbf{0}^T & -\sin\theta_0(k) \\ \mathbf{0} & \mathbf{I}_N & \mathbf{0} \\ \sin\theta_0(k) & \mathbf{0}^T & \cos\theta_0(k) \end{bmatrix} &\begin{bmatrix} e_{f_{q_1}}^{(0)}(k + 1) \\ \lambda^{1/2}d_{fq_{2_1}}(k) \\ \vdots \\ \lambda^{1/2}d_{fq_{2_{N+1}}}(k) \end{bmatrix} \end{aligned} \quad (5.1)$$

O produto dos primeiros dois termos, da direita para a esquerda, resulta em

$$\begin{bmatrix} \cos\theta_0(k)e_{f_{q_1}}^{(0)}(k + 1) - \sin\theta_0\lambda^{1/2}d_{fq_{2_{N+1}}}(k) \\ \lambda^{1/2}d_{fq_{2_1}}(k) \\ \vdots \\ \lambda^{1/2}d_{fq_{2_N}}(k) \\ \sin\theta_0(k)e_{f_{q_1}}^{(0)}(k + 1) + \cos\theta_0(k)\lambda^{1/2}d_{fq_{2_{N+1}}}(k) \end{bmatrix} \quad (5.2)$$

O primeiro e o último termos da equação acima são, respectivamente, $e_{f_{q_1}}^{(1)}(k + 1)$ e $d_{fq_{2_{N+1}}}(k + 1)$. Se os outros produtos forem computados, podemos chegar as seguintes relações:

$$e_{f_{q_1}}^{(i)}(k + 1) = \cos\theta_{i-1}(k)e_{f_{q_1}}^{(i-1)}(k + 1) - \sin\theta_{i-1}(k)\lambda^{1/2}d_{fq_{2_{N+2-i}}}(k) \quad (5.3)$$

$$d_{fq_{2_{N+2-i}}}(k + 1) = \sin\theta_{i-1}(k)e_{f_{q_1}}^{(i-1)}(k + 1) + \cos\theta_{i-1}(k)\lambda^{1/2}d_{fq_{2_{N+2-i}}}(k) \quad (5.4)$$

onde $i = 1, 2, \dots, N + 1$.

Se usarmos um procedimento similar com a equação $\begin{bmatrix} \mathbf{0} \\ \|\mathbf{e}_f^{(0)}(k + 1)\| \end{bmatrix} =$
 $\mathbf{Q}'_{\theta_f}(k + 1) \begin{bmatrix} \mathbf{d}_{fq_2}(k + 1) \\ \|\mathbf{e}_f(k + 1)\| \end{bmatrix}$ (parte de (4.82) usada no algoritmo FQR_POS_B),
encontraremos

$$\cos\theta'_{f_{i-1}}(k+1) = \frac{\| \mathbf{e}_f^{(i)}(k+1) \|}{\| \mathbf{e}_f^{(i-1)}(k+1) \|} \quad (5.5)$$

$$\sin\theta'_{f_{i-1}}(k+1) = \frac{dfq_{2N+2-i}(k+1)}{\| \mathbf{e}_f^{(i-1)}(k+1) \|} \quad (5.6)$$

Na última equação, i varia de 1 a $N + 1$ e a atualização da energia do erro progressivo (forward) é feita pela seguinte generalização de (4.69)

$$\| \mathbf{e}_f^{(i)}(k+1) \| = \sqrt{\lambda \| \mathbf{e}_f^{(i)}(k) \|^2 + [e_{fq_1}^{(i)}(k+1)]^2} \quad (5.7)$$

Todas as demais equações são reunidas num “loop” único onde computamos resultados parciais dos resultados parciais das equações anteriores. O algoritmo resultante é mostrado na Tabela 5.2 e, embora não idêntico, é similar ao apresentado em [42]. Um estágio de sua estrutura em treliça é mostrado na Figura 5.1 onde os processadores de rotação e de ângulos podem ser facilmente entendidos a partir da descrição algorítmica.

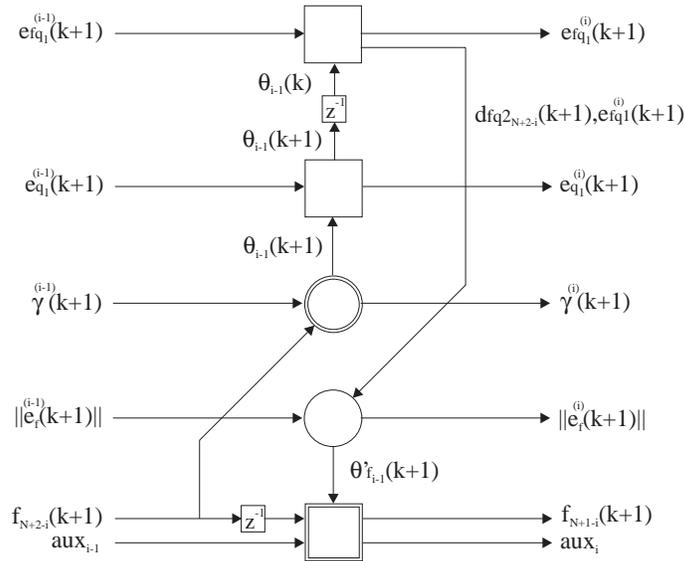


Figura 5.1: Um estágio da estrutura em treliça do algoritmo FQR_POS_B.

Finalmente, a versão em treliça do algoritmo FQR_PRI_B é obtida num modo muito similar ao usado para derivar a versão em treliça do algoritmo FQR_POS_B [19]. O algoritmo é mostrado na Tabela 5.3 e a Figura 5.2 detalha um estágio da estrutura em treliça deste algoritmo.

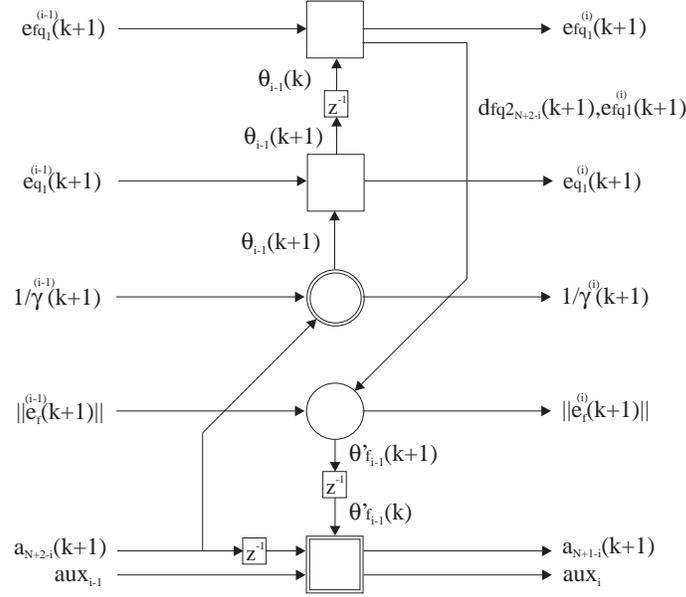


Figura 5.2: Um estágio da estrutura em treliça do algoritmo FQR_PRI_B.

5.3 Resultados das Simulações

Esta seção apresenta alguns resultados de simulações feitas para testar os algoritmos QR rápidos num ambiente de precisão finita. A configuração é um problema de identificação de sistema com uma ordem $N = 10$. O sinal de entrada foi um ruído colorido cuja razão de autovalores é em torno de 187 e a relação sinal desejado para ruído de medição foi feita $SNR = 40dB$. O erro médio quadrático (MSE) em dB foi medido após rodarmos o algoritmo com aritmética de ponto flutuante e com quantização aplicada a mantissa em todas as operações. A mantissa foi arredondada excluindo-se o bit de sinal assumindo que o tamanho de palavra do expoente era suficiente para representar todas faixas dinâmicas. Em todos algoritmos, a restrição de rotações passivas ($\sin^2 \theta + \cos^2 \theta \leq 1$) foi imposta. No primeiro experimento, o

tamanho da mantissa foi variado (8 a 16 bits excluindo o bit de sinal) enquanto mantendo fixo o valor do fator de esquecimento ($\lambda = 0.98$). Em seguida, o λ foi variado (0.90 a 1.0) para um tamanho de palavra de mantissa fixo (10 bits). Os resultados, os quais correspondem a uma média de 10 rodadas independentes, podem ser observados na Figura 5.3 e na Figura 5.4. As figuras mostram que o algoritmo FQR_PRI_B em treliça tem um desempenho em precisão finita próximo aos demais algoritmos QR rápidos especialmente quando λ não está muito próximo de um. É também interessante notar que, embora os algoritmos *a priori* mostrem um desempenho de certa forma inferior, estes algoritmos não requerem a restrição de rotações passivas para terem a consistência retrógrada garantida [43]. É também afirmado em [43] que eles possuem desempenho melhor para pequenos tamanhos de mantissa e λ não tão próximos de 1.

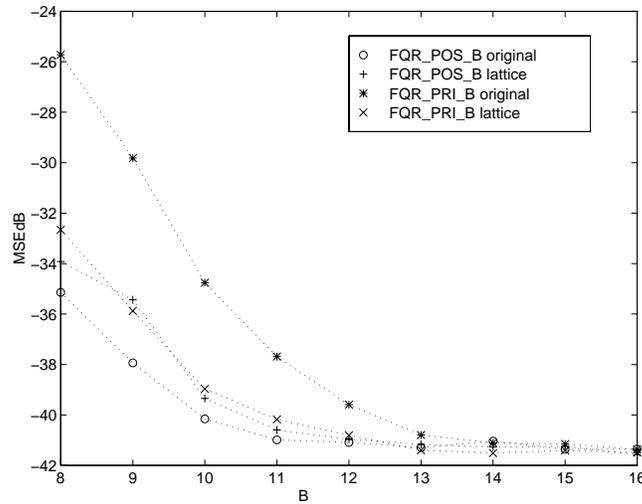


Figura 5.3: Desempenho dos algoritmos num ambiente de precisão finita (variando B, o número de bits da mantissa).

5.4 Conclusões

Este capítulo apresentou as versões totalmente em treliça ou “fully lattice” dos algoritmos QR rápidos que atualizam os erros regressivos *a posteriori* e *a priori*. Os

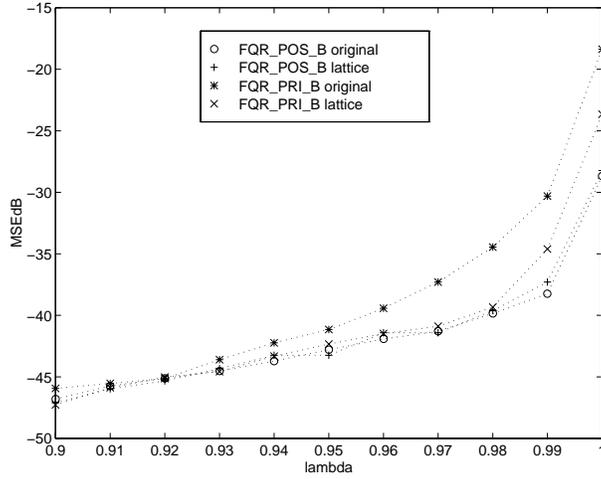


Figura 5.4: MSE em db para diferentes valores de λ .

resultados da ortogonalização de Gram-Schmidt usados no capítulo 4 foram usados para conjecturar o motivo pelo qual somente os algoritmos QR rápidos usando triangularização inferior teriam suas versões em treliça facilmente implementáveis.

As descrições detalhadas das versões em treliça dos algoritmos FQR_POS_B e FQR_PRI_B foram apresentadas.

Resultados de simulações mostraram que o desempenho das versões em treliça num ambiente em precisão finita é comparável com os algoritmos originais. As versões em treliça possuem a mesma complexidade computacional de seus algoritmos originais (FQR_POS_B e FQR_PRI_B) e uma menor complexidade que os algoritmos QR em treliça anteriormente propostos em [22].

Tabela 5.2: A versão em treliça do algoritmo FQR_POS_B.

LATTICE FQR_POS_B
<p>Inicialização:</p> <p>$\epsilon =$ valor positivo e pequeno;</p> <p>for $i = 0 : N + 1$</p> <p style="padding-left: 20px;">{ $\ \mathbf{e}_f^{(i)}(k) \ = \epsilon;$</p> <p style="padding-left: 20px;">}</p> <p>$\mathbf{d}_{fq_2}(k) = \text{zeros}(N + 1, 1);$</p> <p>$\mathbf{d}_{q_2}(k) = \text{zeros}(N + 1, 1);$</p> <p>$\cos\boldsymbol{\theta}(k) = \text{ones}(N + 1, 1);$</p> <p>$\sin\boldsymbol{\theta}(k) = \text{zeros}(N + 1, 1);$</p> <p>$\mathbf{f}(k) = \text{zeros}(N + 1, 1);$</p> <p>for $k = 1, 2, \dots$</p> <p style="padding-left: 20px;">{ $e_{fq_1}^{(0)}(k + 1) = x(k + 1);$</p> <p style="padding-left: 40px;">$\ \mathbf{e}_f^{(0)}(k + 1) \ = \sqrt{[e_{fq_1}^{(0)}(k + 1)]^2 + \lambda \ \mathbf{e}_f^{(0)}(k) \ ^2};$</p> <p style="padding-left: 40px;">$aux_0 = \frac{x(k+1)}{\ \mathbf{e}_f^{(0)}(k+1) \ };$</p> <p style="padding-left: 40px;">$f_{N+1}(k + 1) = aux_0;$</p> <p style="padding-left: 40px;">$\gamma^{(0)}(k + 1) = 1;$</p> <p style="padding-left: 40px;">$e_{q_1}^{(0)}(k + 1) = d(k + 1);$</p> <p style="padding-left: 20px;">for $i = 1 : N + 1$</p> <p style="padding-left: 40px;">{ $d_{fq_{2N+2-i}}(k + 1) = \sin\theta_{i-1}(k)e_{fq_1}^{(i-1)}(k + 1) +$</p> <p style="padding-left: 80px;">$\cos\theta_{i-1}(k)\lambda^{1/2}d_{fq_{2N+2-i}}(k);$</p> <p style="padding-left: 40px;">$e_{fq_1}^{(i)}(k + 1) = \cos\theta_{i-1}(k)e_{fq_1}^{(i-1)}(k + 1) -$</p> <p style="padding-left: 80px;">$\sin\theta_{i-1}(k)\lambda^{1/2}\mathbf{d}_{fq_{2N+2-i}}(k);$</p> <p style="padding-left: 40px;">}</p>

(Continuação da Tabela 5.2)

$$\begin{aligned}
\| \mathbf{e}_f^{(i)}(k+1) \| &= \sqrt{[e_{fq_1}^{(i)}(k+1)]^2 + \lambda \| \mathbf{e}_f^{(i)}(k) \|^2}; \\
\cos\theta'_{f_{i-1}}(k+1) &= \| \mathbf{e}_f^{(i)}(k+1) \| / \| \mathbf{e}_f^{(i-1)}(k+1) \|; \\
\sin\theta'_{f_{i-1}}(k+1) &= d_{fq_{2N+2-i}}(k+1) / \| \mathbf{e}_f^{(i-1)}(k+1) \|; \\
f_{N+1-i}(k+1) &= \frac{f_{N+2-i}(k) - \sin\theta'_{f_{i-1}}(k+1)aux_{i-1}}{\cos\theta'_{f_{i-1}}(k+1)}; \\
aux_i &= -\sin\theta'_{f_{i-1}}(k+1)f_{N+1-i}(k+1) + \cos\theta'_{f_{i-1}}(k+1)aux_{i-1}; \\
\gamma^{(i)}(k+1) &= \sqrt{[\gamma^{(i-1)}(k+1)]^2 - [f_{N+2-i}(k+1)]^2}; \\
\cos\theta_{i-1}(k+1) &= \frac{\gamma^{(i)}(k+1)}{\gamma^{(i-1)}(k+1)}; \\
\sin\theta_{i-1}(k+1) &= \frac{f_{N+2-i}(k+1)}{\gamma^{(i-1)}(k+1)}; \\
dq_{2N+2-i}(k+1) &= \sin\theta_{i-1}(k+1)e_{q_1}^{(i-1)}(k+1) + \\
&\quad \cos\theta_{i-1}(k+1)\lambda^{1/2}d_{q_{2N+2-i}}(k); \\
e_{q_1}^{(i)}(k+1) &= \cos\theta_{i-1}(k+1)e_{q_1}^{(i-1)}(k+1) - \\
&\quad \sin\theta_{i-1}(k+1)\lambda^{1/2}d_{q_{2N+2-i}}(k); \\
\} \\
e(k+1) &= e_{q_1}^{(N+1)}(k+1)\gamma^{(N+1)}(k+1); \\
\}
\end{aligned}$$

Tabela 5.3: A versão em treliça do algoritmo FQR_PRLB.

LATTICE FQR_PRLB
<p>Inicialização:</p> <p>$\epsilon =$ valor positivo e pequeno;</p> <p>for $i = 0 : N + 1$</p> <p>{ $\ \mathbf{e}_f^{(i)}(k) \ = \epsilon;$</p> <p>}</p> <p>$\mathbf{d}_{fq_2}(k) = \text{zeros}(N + 1, 1);$</p> <p>$\mathbf{d}_{q_2}(k) = \text{zeros}(N + 1, 1);$</p> <p>$\cos\boldsymbol{\theta}(k) = \text{ones}(N + 1, 1);$</p> <p>$\cos\boldsymbol{\theta}'_f(k) = \text{ones}(N + 1, 1);$</p> <p>$\sin\boldsymbol{\theta}(k) = \text{zeros}(N + 1, 1);$</p> <p>$\sin\boldsymbol{\theta}'_f(k) = \text{zeros}(N + 1, 1);$</p> <p>$\mathbf{a}(k) = \text{zeros}(N + 1, 1);$</p> <p>for $k = 1, 2, \dots$</p> <p>{ $aux_0 = \frac{x^{(k+1)}}{\sqrt{\lambda}\ \mathbf{e}_f^{(0)}(k)\ };$</p> <p>$a_{N+1}(k + 1) = aux_0;$</p> <p>$e_{fq_1}^{(0)}(k + 1) = x(k + 1);$</p> <p>$\ \mathbf{e}_f^{(0)}(k + 1) \ = \sqrt{[e_{fq_1}^{(0)}(k + 1)]^2 + \lambda \ \mathbf{e}_f^{(0)}(k) \ ^2};$</p> <p>$1/\gamma^{(0)}(k + 1) = 1;$</p> <p>$e_{q_1}^{(0)}(k + 1) = d(k + 1);$</p>

(Continuação da Tabela 5.3)

for $i = 1 : N + 1$

$$\left\{ \begin{array}{l} a_{N+1-i}(k+1) = \frac{a_{N+2-i}(k) - \sin\theta'_{f_{i-1}}(k)aux_{i-1}}{\cos\theta'_{f_{i-1}}(k)}; \end{array} \right.$$

$$aux_i = -\sin\theta'_{f_{i-1}}(k)a_{N+1-i}(k+1) + \cos\theta'_{f_{i-1}}(k)aux_{i-1};$$

$$d_{fq2_{N+2-i}}(k+1) = \sin\theta_{i-1}(k)e_{fq1}^{(i-1)}(k+1) + \cos\theta_{i-1}(k)\lambda^{1/2}d_{fq2_{N+2-i}}(k);$$

$$e_{fq1}^{(i)}(k+1) = \cos\theta_{i-1}(k)e_{fq1}^{(i-1)}(k+1) -$$

$$\sin\theta_{i-1}(k)\lambda^{1/2}d_{fq2_{N+2-i}}(k);$$

$$\|e_f^{(i)}(k+1)\| = \sqrt{[e_{fq1}^{(i)}(k+1)]^2 + \lambda \|e_f^{(i)}(k)\|^2};$$

$$\cos\theta'_{f_{i-1}}(k+1) = \|e_f^{(i)}(k+1)\| / \|e_f^{(i-1)}(k+1)\|;$$

$$\sin\theta'_{f_{i-1}}(k+1) = d_{fq2_{N+2-i}}(k+1) / \|e_f^{(i-1)}(k+1)\|;$$

$$1/\gamma^{(i)}(k+1) = \sqrt{[1/\gamma^{(i-1)}(k+1)]^2 + [a_{N+2-i}(k+1)]^2};$$

$$\cos\theta_{i-1}(k+1) = \frac{1/\gamma^{(i-1)}(k+1)}{1/\gamma^{(i)}(k+1)};$$

$$\sin\theta_{i-1}(k+1) = \frac{a_{N+2-i}(k+1)}{1/\gamma^{(i)}(k+1)};$$

$$d_{q2_{N+2-i}}(k+1) = \sin\theta_{i-1}(k+1)e_{q1}^{(i-1)}(k+1) +$$

$$\cos\theta_{i-1}(k+1)\lambda^{1/2}d_{q2_{N+2-i}}(k);$$

$$e_{q1}^{(i)}(k+1) = \cos\theta_{i-1}(k+1)e_{q1}^{(i-1)}(k+1) -$$

$$\sin\theta_{i-1}(k+1)\lambda^{1/2}d_{q2_{N+2-i}}(k);$$

}

$$e(k+1) = e_{q1}^{(N+1)}(k+1)/[1/\gamma^{(N+1)}(k+1)];$$

}

Capítulo 6

Contribuições à Análise em Precisão Finita dos Algoritmos QR Rápidos

6.1 Introdução

No capítulo 4, foi ressaltado que os algoritmos QR rápidos com triangularização inferior (de acordo com a notação usada neste trabalho) da matriz de entrada de dados são mínimos do ponto de vista de teoria de sistemas e apresentam estabilidade retrógrada [40]. Foi mostrado também que os mesmos tem uma carga computacional menor quando comparados aos algoritmos QR rápidos usando triangularização superior. Além disto, foi observado em [43] que o algoritmo FQR_PRI_B apresenta um desempenho melhor para fatores de esquecimento não tão próximos a um devido ao fato de rotações passivas não serem necessárias.

Estes fatos tornam estes algoritmos candidatos naturais para implementação prática e o efeito da precisão finita é, então, um tópico do maior interesse. Este capítulo trata com a determinação do erro de quantização em estado estacionário de ambos algoritmos FQR_POS_B e FQR_PRI_B. Uma análise em precisão finita completa para ambos algoritmos está além do escopo deste capítulo mas o processo

de obtenção do valor médio quadrático dos erros de quantização acumulados de algumas variáveis é apresentado como uma contribuição à solução deste problema. A validação das expressões obtidas é efetivada por intermédio de simulações em computador.

6.2 Análise em Precisão Infinita

Esta seção faz uma revisão da faixa dinâmica das variáveis internas dos algoritmos FQR_POS_B e FQR_PRI_B. A maioria dos resultados apresentados nesta seção são oriundos de trabalhos já publicados mas foram revistos aqui para unificar a notação e por serem necessários à análise em precisão finita.

6.2.1 Resultados de Precisão Infinita para o Algoritmo FQR_POS_B

Todas as variáveis possuem a mesma notação encontrada no Apêndice C. Note que a versão 2 do algoritmo FQR_POS_B será usada daqui para frente.

Valores Médio-Quadráticos de $\cos \theta_i(k)$ e $\sin \theta_i(k)$

Os resultados seguintes podem ser encontrados em [44].

$$E[\cos^2 \theta_i(k)] \approx \lambda \tag{6.1}$$

$$E[\sin^2 \theta_i(k)] \approx 1 - \lambda \tag{6.2}$$

Valor Médio-Quadrático de $e_{fq_1}^{(i)}(k)$

O resultado seguinte pode ser encontrado em [45].

$$E \left\{ [e_{fq_1}^{(i)}(k)]^2 \right\} \approx \sigma_x^2 \left(\frac{2\lambda}{1 + \lambda} \right)^i \tag{6.3}$$

Valor Médio-Quadrático de $d_{fq2i}(k)$

O resultado seguinte pode ser encontrado em [45].

$$E \{ [d_{fq2i}(k)]^2 \} \approx \frac{\sigma_x^2}{1 + \lambda} \left(\frac{2\lambda}{1 + \lambda} \right)^{N+1-i} \quad (6.4)$$

Valor médio-Quadrático de $\| \mathbf{e}_f^{(i)}(k) \|$

O resultado seguinte pode ser encontrado em [45].

$$E[\| \mathbf{e}_f^{(i)}(k) \|^2] \approx \frac{\sigma_x^2}{1 - \lambda} \left(\frac{2\lambda}{1 + \lambda} \right)^i \quad (6.5)$$

Valores Médio-Quadráticos de $\cos \theta'_{f_i}(k)$ e $\sin \theta'_{f_i}(k)$

Os resultados seguintes podem ser encontrados em [45].

$$E[\cos^2 \theta'_{f_i}(k)] \approx \frac{2\lambda}{1 + \lambda} \quad (6.6)$$

$$E[\sin^2 \theta'_{f_i}(k)] \approx \frac{1 - \lambda}{1 + \lambda} \quad (6.7)$$

Valor Médio-Quadrático de $\gamma^{(i)}(k)$

Se recordarmos do capítulo 4 que $\gamma(k) = \prod_{i=0}^N \cos \theta_i(k)$, usarmos (6.1) e (6.2), e assumirmos independência entre $\cos \theta_i(k)$ e $\cos \theta_j(k)$, $i \neq j$, é fácil encontrar a próxima expressão também obtida em [43] usando uma abordagem diferente.

$$E \{ [\gamma^{(i)}(k)]^2 \} \approx \lambda^i \quad (6.8)$$

Valor Médio-Quadrático de $f_i(k)$

Se tomarmos o quadrado de $\gamma^{(i)}(k + 1)$ da versão 2 do algoritmo FQR_POS_B (Apêndice C) e substituírmos $\cos^2 \theta_{i-1}(k + 1)$ e então $\sin^2 \theta_{i-1}(k + 1)$ pelas expressões também encontradas lá, obtemos

$$[\gamma^{(i)}(k + 1)]^2 = [\gamma^{(i-1)}(k + 1)]^2 - f_{N+2-i}^2(k + 1) \quad (6.9)$$

Tomando agora o valor esperado de (6.9) e a aproximação de (6.8), encontramos a seguinte expressão também disponível em [43].

$$E[f_i^2(k)] \approx \lambda^{N+1-i}(1 - \lambda) \quad (6.10)$$

Tomando do Apêndice C as expressões para $f_{i-1}(k+1)$ e aux_i , e usando-as para calcular $E[f_{i-1}^2(k)] + E[aux_i^2]$, é simples de obter $E[aux_i^2] + E[f_{i-1}^2(k)] = E[aux_{i-1}^2] + E[f_i^2(k)]$. Uma vez que $f_{N+1}(k+1) = aux_{N+1}$, é fácil descobrir que $E[aux_i^2] = E[f_i^2(k)]$ e, portanto, (6.10) pode também ser usada como uma boa aproximação para $E[aux_i^2]$.

Valor Médio-Quadrático de $d_{q2_i}(k)$

O resultado seguinte pode ser encontrado em [44].

$$E[d_{q2_{N+1-i}}^2(k)] \approx \left[\frac{2\lambda}{1+\lambda} \right]^i \left[\frac{\sigma_x^2}{1-\lambda} w_{0,i}^2 + \frac{\sigma_x^2}{1+\lambda} \sum_{j=i+1}^N w_{0,j}^2 \right] \quad (6.11)$$

onde $w_{0,i}^2 = E[w_i^2(k)]$. Observe que embora $w_{0,i}$ não está disponível, uma estimativa grosseira de $\sigma_x^2 w_{0,i}^2$ pode ser obtida baseada na potência do sinal de referência [44].

Valor Médio-Quadrático de $e_{q1}^{(i)}(k)$

Da parte de estimação conjunta do algoritmo FQR_POS_B nós tomamos as expressões de $e_{q1}^{(i)}(k+1)$ e $d_{q2_{N+2-i}}(k+1)$, e usamos para derivar o valor esperado de $[e_{q1}^{(i)}(k+1)]^2 + [d_{q2_{N+2-i}}(k+1)]^2$. Assumindo estacionaridade, encontramos a seguinte relação.

$$E \{ [e_{q1}^{(i)}(k)]^2 \} = E \{ [e_{q1}^{(i-1)}(k)]^2 \} - (1 - \lambda) E [d_{q2_{N+2-i}}^2(k)] \quad (6.12)$$

onde $E \{ [e_{q1}^{(0)}(k)]^2 \} = \sigma_d^2 = \sigma_x^2 \sum_{i=0}^N w_{0,i}^2 + \sigma_n^2$ é a variância do sinal de referência e σ_n^2 é a variância do ruído de medida (é assumido aqui que o algoritmo é aplicado num problema de identificação de sistema com ordem suficiente, i.e., o sistema FIR desconhecido tem a mesma ordem do filtro adaptativo).

Finalmente, da última equação do algoritmo, temos $E[e^2(k)] \approx \lambda^{N+1} E[e_{q1}^2(k)]$. Desde que de (6.12) e (6.11) nós temos que $E[e_{q1}^2(k)] = \sigma_n^2$, a seguinte expressão

resulta:

$$E[e^2(k)] \approx \lambda^{N+1} \sigma_n^2 \quad (6.13)$$

6.2.2 Resultados de Precisão Infinita para o Algoritmo FQR_PRI_B

O algoritmo FQR_PRI_B é similar ao algoritmo FQR_POS_B num sentido de que cinco de suas sete equações são idênticas (três delas são equações matriciais) como pode ser visto das Tabelas 4.6 e 4.7. Isto significa que exceto (6.9) e (6.10) todas outras equações apresentadas neste capítulo permanecem válidas, restando somente a expressão para o valor médio-quadrático de $a_i(k)$ para completar a análise de precisão infinita deste algoritmo. É importante mencionarmos que somente a versão 2 do algoritmo FQR_PRI_B, como descrita no Apêndice C, está sendo investigada aqui.

Valor Médio-Quadrático de $a_i(k)$

Da implementação de (4.80) descrita na versão 2 do algoritmo FQR_PRI_B (passo “Obtendo $\mathbf{Q}_\theta(k+1)$ ”) nós temos (veja Apêndice C)

$$a_{N+2-i}^2(k+1) = [\gamma^{(i)}(k+1)]^{-2} - [\gamma^{(i-1)}(k+1)]^{-2} \quad (6.14)$$

Tomando-se o valor esperado de (6.14), usando a aproximação de (6.8), e empregando a aproximação conhecida por seu termo em inglês “averaging principle” [46, 47], segue a próxima expressão também disponível em [43].

$$E[a_i^2(k)] \approx \lambda^{-(N+2-i)}(1 - \lambda) \quad (6.15)$$

Se pegarmos do Apêndice C as expressões para $a_{i-1}(k+1)$ e aux_i , e usarmos as mesmas para calcular $E[a_{i-1}^2(k)] + E[aux_i^2]$, é simples obtermos $E[aux_i^2] + E[a_{i-1}^2(k)] = E[aux_{i-1}^2] + E[a_i^2(k)]$. Sendo $a_{N+1}(k+1) = aux_{N+1}$, é fácil verificar que $E[aux_i^2] = E[a_i^2(k)]$ e, portanto, (6.15) pode também ser usada como uma boa aproximação para $E[aux_i^2]$.

6.3 Contribuição à Análise em Precisão Finita

Esta seção apresenta inicialmente o modelo para o erro de quantização de ponto fixo a ser usado no resto do capítulo. A primeira equação matricial do algoritmo FQR_PRLB é então analisada como um exemplo do método usado. Os resultados desta análise são muito bons como será visto na próxima seção. Por outro lado, a expressão de análise para a inicialização da segunda equação matricial do mesmo algoritmo apresentou resultados pobres quando o conhecido “averaging principle” foi usado. Uma alternativa para este princípio é então derivada e aplicada para desenvolver uma aproximação mais precisa.

6.3.1 Modelo do Erro de Quantização de Ponto Fixo

Uma quantidade razoável de pressupostos são feitos aqui de modo a termos um modelo simples do erro de quantização. Assumimos que nenhum estouro ou “overflow” devido a adição ou subtração ocorre. Aritmética de complemento a dois é usada para representações numéricas das variáveis internas as quais são assumidas corretamente escaladas para evitar estouro. É assumido que a operação $*$ (multiplicação, divisão ou exponenciação) introduz o seguinte erro de quantização.

$$\eta_*(a, b) \triangleq a * b - Q[a * b] \quad (6.16)$$

onde a e b são escalares e é assumido que as quantizações instantâneas são realizadas pelo arredondamento tal que o erro de quantização é um ruído branco com média zero e variância $\frac{2^{-2B}}{12}$, B sendo o número de bits excluindo o bit de sinal.

O erro de quantização acumulado é definido como a diferença entre a implementação em precisão infinita e a implementação em precisão finita da variável. Se $r(k)$ é uma variável, seu *erro de quantização acumulado* é dado por

$$\Delta r(k) \triangleq r(k) - r_Q(k) \quad (6.17)$$

e nós estamos interessados aqui em descobrir o valor médio quadrático desta quantidade.

6.3.2 O Algoritmo FQR_PRI_B: Valor Médio-Quadrático de

$$\Delta e_{fq_1}^{(i)}(k) \text{ e } \Delta d_{fq_{2i}}(k)$$

Começamos analisando a equação (4.70) em precisão finita. Note que as expressões de análise que serão obtidas desta equação matricial serão válidas para ambos algoritmos FQR_POS_B e FQR_PRI_B (sendo também esperado que sejam bem semelhantes às expressões equivalentes para os algoritmos FQR_POS_F e FQR_PRI_F) uma vez que a expressão seguinte é comum para todos os quatro algoritmos QR rápidos estudados.

$$\begin{bmatrix} e_{fq_1}(k+1) \\ \mathbf{d}_{fq_2}(k+1) \end{bmatrix} = \mathbf{Q}_\theta(k) \begin{bmatrix} x(k+1) \\ \lambda^{1/2} \mathbf{d}_{fq_2}(k) \end{bmatrix} \quad (6.18)$$

onde $x(k+1)$ é assumido já quantizado bem como todos outros sinais externos e constantes. A equação anterior é implementada como

$$\begin{aligned} e_{fq_1}^{(i)}(k+1) &= \cos \theta_{i-1}(k) e_{fq_1}^{(i-1)}(k+1) \\ &\quad - \sin \theta_{i-1}(k) \lambda^{1/2} d_{fq_{2N+2-i}}(k) \end{aligned} \quad (6.19)$$

$$\begin{aligned} d_{fq_{2N+2-i}}(k+1) &= \sin \theta_{i-1}(k) e_{fq_1}^{(i-1)}(k+1) \\ &\quad + \cos \theta_{i-1}(k) \lambda^{1/2} d_{fq_{2N+2-i}}(k) \end{aligned} \quad (6.20)$$

onde $i = 1 : N + 1$ e $e_{fq_1}^{(0)}(k+1) = x(k+1)$. As equações (6.19) e (6.20) acima implementadas em precisão finita são dadas por

$$\begin{aligned} e_{fq_{1;Q}}^{(i)}(k+1) &= \cos \theta_{i-1;Q}(k) e_{fq_{1;Q}}^{(i-1)}(k+1) \\ &\quad - \sin \theta_{i-1;Q}(k) \lambda^{1/2} d_{fq_{2N+2-i;Q}}(k) - \eta_1(k) \end{aligned} \quad (6.21)$$

$$\begin{aligned} d_{fq_{2N+2-i;Q}}(k+1) &= \sin \theta_{i-1;Q}(k) e_{fq_{1;Q}}^{(i-1)}(k+1) \\ &\quad + \cos \theta_{i-1;Q}(k) \lambda^{1/2} d_{fq_{2N+2-i;Q}}(k) - \eta_2(k) \end{aligned} \quad (6.22)$$

onde $\eta_1(k) = \eta_M(\cos \theta_{i-1;Q}(k), e_{fq_{1;Q}}^{(i-1)}(k+1)) - \eta_M(\sin \theta_{i-1;Q}(k), \lambda^{1/2}, d_{fq_{2N+2-i;Q}}(k))$ e $\eta_2(k) = \eta_M(\sin \theta_{i-1;Q}(k), e_{fq_{1;Q}}^{(i-1)}(k+1)) + \eta_M(\cos \theta_{i-1;Q}(k), \lambda^{1/2}, d_{fq_{2N+2-i;Q}}(k))$ são os erros de quantização introduzidos pelas multiplicações.

De (6.19) e (6.21), derivamos a expressão para o erro de quantização acumulado de $e_{f_{q_1}}^{(i)}(k+1)$ como segue.

$$\begin{aligned}
\Delta e_{f_{q_1}}^{(i)}(k+1) &= e_{f_{q_1}}^{(i)}(k+1) - e_{f_{q_1};Q}^{(i)}(k+1) \\
&= \cos \theta_{i-1}(k) e_{f_{q_1}}^{(i-1)}(k+1) \\
&\quad - \sin \theta_{i-1}(k) \lambda^{1/2} d_{fq_{2N+2-i}}(k) \\
&\quad - \cos \theta_{i-1;Q}(k) e_{f_{q_1};Q}^{(i-1)}(k+1) \\
&\quad + \sin \theta_{i-1;Q}(k) \lambda^{1/2} d_{fq_{2N+2-i};Q}(k) + \eta_1(k) \quad (6.23)
\end{aligned}$$

Substituindo em (6.23) os quatro valores quantizados pelos valores em precisão infinita menos o erro de quantização acumulado (tal como $r_Q(k) = r(k) - \Delta r(k)$) e admitindo que os valores absolutos dos produtos cruzados dos erros de quantização acumulados são muito menores do que os valores absolutos dos outros termos, encontramos

$$\begin{aligned}
\Delta e_{f_{q_1}}^{(i)}(k+1) &\approx \cos \theta_{i-1}(k) \Delta e_{f_{q_1}}^{(i-1)}(k+1) + \Delta \cos \theta_{i-1}(k) e_{f_{q_1}}^{(i-1)}(k+1) \\
&\quad - \sin \theta_{i-1}(k) \lambda^{1/2} \Delta d_{fq_{2N+2-i}}(k) \\
&\quad - \Delta \sin \theta_{i-1}(k) \lambda^{1/2} d_{fq_{2N+2-i}}(k) + \eta_1(k) \quad (6.24)
\end{aligned}$$

Se assumimos agora que $\eta_1(k)$ e os erros de quantização tenham média zero e uma pequena correlação cruzada entre si, a expressão seguinte pode ser facilmente obtida.

$$\begin{aligned}
E \left\{ [\Delta e_{f_{q_1}}^{(i)}(k+1)]^2 \right\} &\approx E \left\{ [\cos \theta_{i-1}(k) \Delta e_{f_{q_1}}^{(i-1)}(k+1)]^2 \right\} \\
&\quad + E \left\{ [\Delta \cos \theta_{i-1}(k) e_{f_{q_1}}^{(i-1)}(k+1)]^2 \right\} \\
&\quad + \lambda E \left\{ [\sin \theta_{i-1}(k) \Delta d_{fq_{2N+2-i}}(k)]^2 \right\} \\
&\quad + \lambda \left\{ [\Delta \sin \theta_{i-1}(k) d_{fq_{2N+2-i}}(k)]^2 \right\} + E[\eta_1^2(k)] \quad (6.25)
\end{aligned}$$

A expressão final para o valor médio-quadrático de $\Delta e_{f_{q_1}}^{(i)}(k+1)$ é obtida assumindo estacionariedade, assumindo que as variáveis são descorrelacionadas com os erros de quantização acumulados e usando os resultados da análise de precisão

infinita. Ela é dada por

$$\begin{aligned}
E \left\{ [\Delta e_{fq_1}^{(i)}(k)]^2 \right\} &\approx \lambda E \left\{ [\Delta e_{fq_1}^{(i-1)}(k)]^2 \right\} \\
&+ \sigma_x^2 \left(\frac{2\lambda}{1+\lambda} \right)^{i-1} E \left\{ [\Delta \cos \theta_{i-1}(k)]^2 \right\} \\
&+ \lambda(1-\lambda) E \left\{ [\Delta d_{fq_{2N+2-i}}(k)]^2 \right\} \\
&+ \lambda \frac{\sigma_x^2}{1+\lambda} \left(\frac{2\lambda}{1+\lambda} \right) \left\{ [\Delta \sin \theta_{i-1}(k)]^2 \right\} + E[\eta_1^2(k)] \quad (6.26)
\end{aligned}$$

onde $E[\eta_1^2(k)] = 2 \frac{2^{-2B}}{12}$.

O erro de quantização acumulado de $d_{fq_{2N+2-i}}(k+1)$ será derivado a partir de (6.20) e (6.22).

$$\begin{aligned}
\Delta d_{fq_{2N+2-i}}(k+1) &= d_{fq_{2N+2-i}}(k+1) - d_{fq_{2N+2-i};Q}(k+1) \\
&= \sin \theta_{i-1}(k) e_{fq_1}^{(i-1)}(k+1) \\
&\quad + \cos \theta_{i-1}(k) \lambda^{1/2} d_{fq_{2N+2-i}}(k) \\
&\quad - \sin \theta_{i-1;Q}(k) e_{fq_1;Q}^{(i-1)}(k+1) \\
&\quad - \cos \theta_{i-1;Q}(k) \lambda^{1/2} d_{fq_{2N+2-i};Q}(k) + \eta_2(k) \quad (6.27)
\end{aligned}$$

Usando a mesma abordagem e pressupostos usados para obter $E \left\{ [\Delta e_{fq_1}^{(i)}(k)]^2 \right\}$ nós encontramos o valor médio-quadrático do erro de quantização acumulado de $d_{fq_{2N+2-i}}(k+1)$ dado por

$$\begin{aligned}
E \left\{ [\Delta d_{fq_{2N+2-i}}(k)]^2 \right\} &\approx \frac{\sigma_x^2}{2\lambda(1-\lambda)} \left(\frac{2\lambda}{1+\lambda} \right)^i E \left\{ [\Delta \sin \theta_{i-1}(k)]^2 \right\} \\
&+ \frac{1}{1+\lambda} E \left\{ [\Delta e_{fq_1}^{(i-1)}(k)]^2 \right\} \\
&+ \frac{\sigma_x^2}{2(1-\lambda^2)} \left(\frac{2\lambda}{1+\lambda} \right)^i E \left\{ [\Delta \cos \theta_{i-1}(k)]^2 \right\} \\
&+ \frac{E[\eta_2^2(k)]}{1-\lambda^2} \quad (6.28)
\end{aligned}$$

onde $E[\eta_2^2(k)] = 2 \frac{2^{-2B}}{12}$.

6.3.3 O Algoritmo FQR_PRI_B: Valor Médio-Quadrático de

$$\Delta aux_0$$

Na seção anterior, demos como exemplo de análise em precisão finita as expressões para os valores médio-quadráticos dos erros de quantização acumulados da primeira equação matricial do algoritmo FQR_PRI_B. A próxima equação matricial deste algoritmo tem um parâmetro de inicialização (parâmetro aux_0) dado por

$$aux_0 = \frac{e_{fq_1}(k+1)}{\gamma(k)\lambda^{1/2} \|\mathbf{e}_f(k)\|} \quad (6.29)$$

Num ambiente de precisão finita, a equação anterior é implementada como

$$aux_{0;Q} = \frac{e_{fq_1;Q}(k+1)}{\gamma_Q(k)\lambda^{1/2} \|\mathbf{e}_f(k)\|_Q - \eta_3(k)} - \eta_4(k) \quad (6.30)$$

onde $\eta_3(k)$ e $\eta_4(k)$ são os erros de quantização devidos à multiplicação e à divisão, respectivamente.

De (6.29) e (6.30), nós escrevemos o erro de quantização acumulado de aux_0 como

$$\begin{aligned} \Delta aux_0 &= aux_0 - aux_{0;Q} \\ &= \frac{e_{fq_1}(k+1)}{\gamma(k)\lambda^{1/2} \|\mathbf{e}_f(k)\|} \\ &\quad - \frac{e_{fq_1;Q}(k+1)}{\gamma_Q(k)\lambda^{1/2} \|\mathbf{e}_f(k)\|_Q - \eta_3(k)} + \eta_4(k) \\ &= \frac{e_{fq_1}(k+1)}{r} - \frac{e_{fq_1;Q}(k+1)}{r - \Delta r} + \eta_4(k) \end{aligned} \quad (6.31)$$

onde $r = \gamma(k)\lambda^{1/2} \|\mathbf{e}_f(k)\|$.

Se usarmos a aproximação $\frac{1}{r - \Delta r} \approx \frac{1}{r} \left(1 + \frac{\Delta r}{r}\right)$ em (6.31) e substituirmos $e_{fq_1;Q}(k+1)$ por $e_{fq_1}(k+1) - \Delta e_{fq_1}(k+1)$ nós obtemos

$$\Delta aux_0 \approx \frac{\Delta e_{fq_1}(k+1)}{r} - \frac{[e_{fq_1}(k+1) - \Delta e_{fq_1}(k+1)]\Delta r}{r^2} + \eta_4(k) \quad (6.32)$$

Substituindo as expressões de r e Δr em (6.32) e assumindo que o termo com o produto dos dois erros acumulados é muito menor em valor absoluto que os outros

termos, obtemos

$$\begin{aligned} \Delta aux_0 \approx & \frac{\Delta e_{fq_1}(k+1)}{\lambda^{1/2}\gamma(k) \|\mathbf{e}_f(k)\|} - \frac{e_{fq_1}(k+1)\Delta \|\mathbf{e}_f(k)\|}{\lambda^{1/2}\gamma(k) \|\mathbf{e}_f(k)\|^2} \\ & - \frac{e_{fq_1}(k+1)\Delta\gamma(k)}{\lambda^{1/2}\gamma^2(k) \|\mathbf{e}_f(k)\|} - \frac{e_{fq_1}(k+1)\eta_3(k)}{\lambda\gamma^2(k) \|\mathbf{e}_f(k)\|^2} + \eta_4(k) \end{aligned} \quad (6.33)$$

Se agora assumirmos que os erros instantâneos ($\eta_3(k)$ e $\eta_4(k)$) e os erros acumulados ($\Delta e_{fq_1}(k+1)$, $\Delta \|\mathbf{e}_f(k)\|$, e $\Delta\gamma(k)$) possuem média zero com pequena correlação cruzada entre si, o erro de quantização médio-quadrático de aux_0 resulta.

$$\begin{aligned} E\{[\Delta aux_0]^2\} \approx & \frac{1}{\lambda} E\{[\Delta e_{fq_1}(k)]^2\} E\left[\frac{1}{\gamma^2(k)}\right] E\left[\frac{1}{\|\mathbf{e}_f(k)\|^2}\right] \\ & + \frac{E[e_{fq_1}^2(k)]}{\lambda} E\{[\Delta \|\mathbf{e}_f(k)\|]^2\} E\left[\frac{1}{\gamma^2(k)}\right] E\left[\frac{1}{\|\mathbf{e}_f(k)\|^4}\right] \\ & + \frac{E[e_{fq_1}^2(k)]}{\lambda} E\{[\Delta\gamma(k)]^2\} E\left[\frac{1}{\gamma^4(k)}\right] E\left[\frac{1}{\|\mathbf{e}_f(k)\|^2}\right] \\ & + \frac{E[e_{fq_1}^2(k)]E[\eta_3^2(k)]}{\lambda^2} E\left[\frac{1}{\gamma^4(k)}\right] E\left[\frac{1}{\|\mathbf{e}_f(k)\|^4}\right] \\ & + E[\eta_4^2(k)] \end{aligned} \quad (6.34)$$

Da análise em precisão infinita, temos $E[e_{fq_1}^2(k)]$, $E[\gamma^2(k)]$ e $E[\|\mathbf{e}_f(k)\|^2]$. Se usarmos o “averaging principle” e aproximarmos $E[1/x^2]$ e $E[1/x^4]$ por $1/E[x^2]$ e $1/E[x^4]$, respectivamente, e substituirmos os resultados da análise de precisão infinita, encontramos

$$\begin{aligned} E\{[\Delta aux_0]^2\} \approx & \frac{(1-\lambda)}{\lambda^{N+2}\sigma_x^2} \left(\frac{1+\lambda}{2\lambda}\right)^{N+1} E\{[\Delta e_{fq_1}(k)]^2\} \\ & + \frac{(1-\lambda)^2}{\lambda^{N+2}\sigma_x^2} \left(\frac{1+\lambda}{2\lambda}\right)^{N+1} E\{[\Delta \|\mathbf{e}_f(k)\|]^2\} \\ & + \frac{1-\lambda}{\lambda^{2N+3}} E\{[\Delta\gamma(k)]^2\} \\ & + \frac{(1-\lambda)^2}{\lambda^{2N+3}\sigma_x^2} \left(\frac{1+\lambda}{2\lambda}\right)^{N+1} E[\eta_3^2(k)] \\ & + E[\eta_4^2(k)] \end{aligned} \quad (6.35)$$

onde $E[\eta_3^2(k)] = E[\eta_4^2(k)] = \frac{2^{-2B}}{12}$.

Como será visto das simulações, (6.35) apresenta um valor em dB o qual difere mais de um dB do resultado oriundo da simulação. Entretanto, ainda afirmamos

que (6.34) é uma aproximação razoável para $E\{[\Delta au x_0]^2\}$. Vejamos na subseção seguinte como validar (6.34).

6.3.4 Refinando as aproximações de $E[1/x^2]$ e $E[1/x^4]$

O tão chamado “averaging principle” [46] tem sido amplamente usado em derivações de relações usadas em análises de precisão infinita e finita e seu enunciado é

$$E\left\{\frac{f[x(k)]}{g[y(k)]}\right\} \approx \frac{E\{f[x(k)]\}}{E\{g[y(k)]\}} \quad (6.36)$$

o qual no nosso caso é válido para valor de k elevado e fator de esquecimento próximo à unidade [44].

Uma razão do resultado pobre de (6.35) é devida à aproximação de $E[1/x^2(k)]$ por $1/E[x^2(k)]$ e $E[1/x^4(k)]$ por $1/E^2[x^2(k)]$. O propósito desta subseção é derivarmos aproximações mais confiáveis para estas expressões de modo a validarmos (6.34).

Começamos a buscar uma solução alternativa para este problema tomando o *Teorema de Fatoração do Momento Gaussiano* ou “*Gaussian Moment Factoring Theorem*” [1] para quatro amostras de um processo Gaussiano de média zero ¹ e admitindo que elas são todas iguais a $x(k)$. A expressão resultante é

$$E[x^4(k)] = 3E^3[x^2(k)] \quad (6.37)$$

Se admitirmos agora que $x(k)$ tem um valor médio diferente de zero ($E[x(k)] = \bar{x}$) e modelarmos esta variável por $x(k) = x'(k) + \bar{x}$ com $x'(k)$ sendo Gaussiana real e com média zero, teremos

$$\begin{aligned} E[x^4(k)] &= E[(x'(k) + \bar{x})^4] \\ &= 3E^2[x'^2(k)] + 6E[x'^2(k)]\bar{x}^2 + \bar{x}^4 \\ &= 3E^2[x^2(k)] - 2\bar{x}^4 \end{aligned} \quad (6.38)$$

Usando a mesma abordagem de [4] e a equação (6.38) derivaremos uma expressão

¹ $E[z_1 z_2 z_3 z_4] = E[z_1 z_2]E[z_3 z_4] + E[z_1 z_3]E[z_2 z_4] + E[z_1 z_4]E[z_2 z_3]$

para $E[1/x^2(k)]$.

$$\begin{aligned}
E\left[\frac{1}{x^2(k)}\right] &= \frac{1}{E[x^2(k)]} E\left[\frac{1}{1 - \left(1 - \frac{x^2(k)}{E[x^2(k)]}\right)}\right] \\
&= \frac{1}{E[x^2(k)]} E\left[\sum_{i=0}^{\infty} \left(1 - \frac{x^2(k)}{E[x^2(k)]}\right)^i\right] \\
&= \frac{1}{E[x^2(k)]} \left(1 + 0 + \frac{E[x^4(k)] - E^2[x^2(k)]}{E^2[x^2(k)]} + \dots\right) \\
&= \frac{1}{E[x^2(k)]} \left(\frac{E[x^4(k)]}{E^2[x^2(k)]} + \dots\right) \\
&\approx \frac{3E^2[x^2(k)] - 2\bar{x}^4}{E^3[x^2(k)]} \tag{6.39}
\end{aligned}$$

Finalmente, podemos encontrar uma aproximação para $E[1/x^4(k)]$ se usarmos uma variável auxiliar $y = x^2(k)$ e (6.39). É fácil ver que $E[y] = E[x^2(k)]$ e que $E[y^2] = E[x^4(k)]$ é disponível de (6.38).

$$E\left[\frac{1}{x^4(k)}\right] \approx \frac{3\{3E^2[x^2(k)] - 2\bar{x}^4\}^2 - 2\{E[x^2(k)]\}^4}{\{3E^2[x^2(k)] - 2\bar{x}^4\}^3} \tag{6.40}$$

onde $\bar{x} = E[x(k)]$.

Vale a pena mencionarmos que as aproximações acima são válidas sempre que a expansão em série usada na derivação for válida. O caso crítico ocorre quando $x(k)$ tende a zero (divisão por zero) e os melhores surgem quando \bar{x}/σ_x não é tão baixo. Uma pequena simulação foi realizada para checar os resultados das expressões derivadas aqui. Nesta simulação, $x(k)$ era uma variável aleatória Gaussiana com variância $\sigma_x^2 = 10^{-3}$ e valor médio ajustado para 0.1, 0.3, 0.5 e 1.0. A Tabela 6.1 seguinte mostra os resultados para uma simulação com 100000 amostras, donde ficou claro que estas expressões novas apresentam um desempenho melhor que as aproximações velhas para diferentes valores de \bar{x}/σ_x . Note também que para o caso do menor valor médio ($\bar{x} = 0.1$), a variável cruza o zero muitas vezes sendo esta a razão para os altos valores de $E\left[\frac{1}{x^4(k)}\right]$.

Usando as equações (6.38), (6.39) e (6.40), podemos obter as seguintes expressões a serem usadas em (6.34) cujos resultados, como serão vistos nas simulações, são mais precisos que os resultado obtidos com (6.35). Note ainda que as novas relações

Tabela 6.1: Comparação de desempenho das novas expressões.

EXPRESSÃO	$\bar{x} = 0.1$	$\bar{x} = 0.3$	$\bar{x} = 0.5$	$\bar{x} = 1.0$
$E[x^4(k)]$	$1.6313 \cdot 10^{-4}$	0.0086	0.0640	1.0060
$E^2[x^2(k)]$	$1.2100 \cdot 10^{-4}$	0.0083	0.0630	1.0020
(6.38)	$1.6300 \cdot 10^{-4}$	0.0086	0.0640	1.0060
$E[\frac{1}{x^2(k)}]$	$2.8489 \cdot 10^4$	11.5041	4.0490	1.0030
$\frac{1}{E[x^2(k)]}$	90.9092	10.9890	3.9841	0.9990
(6.39)	122.4642	11.4694	4.0474	1.0030
$E[\frac{1}{x^4(k)}]$	$4.1738 \cdot 10^{13}$	139.0497	16.6687	1.0101
$\frac{1}{E^2[x^2(k)]}$	$8.2645 \cdot 10^3$	120.7584	15.8728	0.9980
(6.40)	$1.1644 \cdot 10^4$	134.6784	16.5959	1.0098

obtidas aqui foram usadas especificamente com o propósito de validarmos (6.34) pois não temos $E[\gamma(k)]$ e $E[\|\mathbf{e}_f(k)\|]$ disponíveis da análise em precisão infinita.

$$\begin{aligned}
 E\left[\frac{1}{\gamma^2(k)}\right] &\approx \frac{3E^2[\gamma^2(k)] - 2E^4[\gamma(k)]}{E^3[\gamma^2(k)]} \\
 E\left[\frac{1}{\gamma^4(k)}\right] &\approx \frac{3\{3E^2[\gamma^2(k)] - 2E^4[\gamma(k)]\}^2 - 2E^4[\gamma^2(k)]}{\{3E^2[\gamma^2(k)] - 2E^4[\gamma(k)]\}^3} \\
 E\left[\frac{1}{\|\mathbf{e}_f(k)\|^2}\right] &\approx \frac{3E^2[\|\mathbf{e}_f(k)\|^2] - 2E^4[\|\mathbf{e}_f(k)\|]}{E^3[\|\mathbf{e}_f(k)\|^2]} \\
 E\left[\frac{1}{\|\mathbf{e}_f(k)\|^4}\right] &\approx \frac{3\{3E^2[\|\mathbf{e}_f(k)\|^2] - 2E^4[\|\mathbf{e}_f(k)\|]\}^2 - 2E^4[\|\mathbf{e}_f(k)\|^2]}{\{3E^2[\|\mathbf{e}_f(k)\|^2] - 2E^4[\|\mathbf{e}_f(k)\|]\}^3}
 \end{aligned} \tag{6.41}$$

6.4 Resultados das Simulações

De modo a verificar a precisão das fórmulas apresentadas neste capítulo, uma simulação em computador do algoritmo FQR_PRI_B rodando num ambiente de aritmética de ponto fixo foi realizada. A experiência consistiu de uma configuração

de identificação de sistema com sinal de entrada e ruído de medição com distribuições normais, médias nulas e variâncias $\sigma_x^2 = 10^{-3}$ e $\sigma_n^2 = 10^{-7}$, respectivamente. O sistema FIR desconhecido tinha ordem $N = 4$ e o fator de esquecimento λ foi escolhido igual a 0.95. Nas 500 rodadas independentes, o algoritmo FQR_PRI_B foi simulado com 1000 iterações e as últimas 800 foram tomadas em média para obter-se os resultados apresentados aqui. Arredondamento em complemento a dois foi usado com $B = 15$ bits (excluindo o bit de sinal) enquanto que a precisão infinita foi implementada com aritmética de ponto flutuante com 64 bits.

Todos resultados teóricos dos erros de quantização acumulados mostrados nas tabelas seguintes foram obtidos das relações obtidas da análise onde as quantidades “desconhecidas” — variáveis não analisadas — foram tomadas das simulações com o propósito de verificar as expressões aqui obtidas. Além disto, a coluna DIFERENÇA foi obtida como o valor absoluto da diferença do valor simulado em dB menos o valor teórico em dB .

A Tabela 6.2 abaixo mostra os resultados de (6.26).

Tabela 6.2: Valor médio-quadrático de $\Delta e_{f_{q_1}}^{(i)}(k)$.

EXPRESSÃO	SIMULADA	TEÓRICA	DIFERENÇA (dB)
$E \left\{ [\Delta e_f^{(1)} q_1(k)]^2 \right\}$	$0.0347 \cdot 10^{-8}$	$0.0330 \cdot 10^{-8}$	0.2260
$E \left\{ [\Delta e_f^{(2)} q_1(k)]^2 \right\}$	$0.0692 \cdot 10^{-8}$	$0.0655 \cdot 10^{-8}$	0.2397
$E \left\{ [\Delta e_f^{(3)} q_1(k)]^2 \right\}$	$0.1041 \cdot 10^{-8}$	$0.0982 \cdot 10^{-8}$	0.2538
$E \left\{ [\Delta e_f^{(4)} q_1(k)]^2 \right\}$	$0.1389 \cdot 10^{-8}$	$0.1307 \cdot 10^{-8}$	0.2643
$E \left\{ [\Delta e_f^{(5)} q_1(k)]^2 \right\}$	$0.1730 \cdot 10^{-8}$	$0.1630 \cdot 10^{-8}$	0.2576

Seguindo, a Tabela 6.3 mostra os resultados de (6.28).

A Tabela 6.4 mostra os resultados de $E \{ [\Delta aux_0]^2 \}$ com o auxílio de diferentes relações. Observe a melhoria dos resultados com o refinamento das expressões. É importante mencionarmos que $E[\gamma(k)]$ e $E[\| \mathbf{e}_f(k) \|^2]$ em (6.41) não foram tomados dos resultados das simulações mas calculados de $E[\gamma^2(k)]$, $E[\| \mathbf{e}_f(k) \|^2]$ e as

Tabela 6.3: Valor médio-quadrático de $\Delta d_{fq2_i}(k)$.

EXPRESSÃO	SIMULADA	TEÓRICA	DIFERENÇA (dB)
$E \{[\Delta d_{fq2_1}(k)]^2\}$	0.3330 10^{-8}	0.3813 10^{-8}	0.5875
$E \{[\Delta d_{fq2_2}(k)]^2\}$	0.3128 10^{-8}	0.3531 10^{-8}	0.5258
$E \{[\Delta d_{fq2_3}(k)]^2\}$	0.2914 10^{-8}	0.3252 10^{-8}	0.4771
$E \{[\Delta d_{fq2_4}(k)]^2\}$	0.2608 10^{-8}	0.2976 10^{-8}	0.5739
$E \{[\Delta d_{fq2_5}(k)]^2\}$	0.2458 10^{-8}	0.2704 10^{-8}	0.4151

variâncias de $\gamma(k)$ e $\| \mathbf{e}_f(k) \|^2$. Estas últimas (variâncias de $\gamma(k)$ e $\| \mathbf{e}_f(k) \|^2$) foram efetivamente obtidas das simulações. Este procedimento foi usado para garantir que $E[x^2] > E^2[x]$ uma vez que $E[x^2] = \sigma_x^2 + E^2[x]$.

Tabela 6.4: Valor médio-quadrático de Δaux_0 .

TIPO DE RESULTADO	$E \{[\Delta aux_0]^2\}$
SIMULADO	3.2229 10^{-7}
EQUAÇÃO (6.35)	2.4923 10^{-7}
DIFERENÇA (dB)	1.1165
EQUAÇÕES (6.34) E (6.41)	2.7751 10^{-7}
DIFERENÇA (dB)	0.6497
EQUAÇÕES (6.34) E (6.41) MELHORADAS ²	2.8572 10^{-7}
DIFERENÇA (dB)	0.5230

6.5 Conclusões

Este capítulo visou uma introdução ao tópico análise em precisão finita dos algoritmos QR rápidos baseados nos erros de predição regressivos (algoritmos FQR_POS_B

²Usando resultados de simulações ao invés de seus correspondentes teóricos.

e FQR_PRI_B). Foram apresentadas descrições panorâmicas da análise em precisão infinita para ambos algoritmos e, embora não mencionado anteriormente, cada expressão foi verificada com simulações em computadores e validadas para um valor médio quadrático dentro de um dB para todas as variáveis. A análise da quantização em ponto fixo foi realizada para uma equação matricial a qual é comum para ambos algoritmos e foi testada para o algoritmo FQR_PRI_B com resultados excelentes. Uma outra expressão foi analisada mostrando um possível problema tipicamente encontrado em procedimento de análise. Uma expressão para esta análise foi desenvolvida usando o princípio da média ou “averaging principle” e uma aproximação alternativa foi derivada para melhorar os resultados se as estatísticas de primeira ordem (média) de algumas variáveis são conhecidas. No nosso caso as aproximações aqui derivadas foram úteis somente para validar parte da análise uma vez que as médias das variáveis não estão disponíveis. Uma simulação foi realizada para avaliar a análise parcial apresentada e os resultados mostram que as fórmulas teóricas encontram-se bem de acordo com aqueles obtidos no experimento. O autor pretende completar a análise partindo desta contribuição dada aqui bem como investigar sua extensão ao caso de ponto flutuante.

Capítulo 7

Conclusões e Sugestões

Este capítulo resume os resultados da tese e ressalta diversos problemas que podem ser investigados em possíveis pesquisas futuras. O trabalho focalizou-se em duas famílias de filtros adaptativos: o algoritmo LMS binormalizado com reutilização de dados (BNDR-LMS) e o algoritmo QR rápido baseado na utilização de erros de predição progressiva *a priori* (FQR_PRI_F). A investigação destes algoritmos conduziu ao estudo de um número de resultados de pesquisa relevantes tais como a análise de erro médio-quadrático do algoritmo BNDR-LMS, a versão com restrições do algoritmo BNDR-LMS e suas aplicações e a versão em treliça bem como a análise em precisão finita dos algoritmos QR rápidos; todos eles abordados ao longo da tese.

7.1 Conclusões

O primeiro capítulo da tese apresentou uma breve revisão da teoria básica de filtragem adaptativa com especial atenção aos algoritmos baseados no LMS e na decomposição QR.

O Capítulo 2 apresentou o algoritmo BNDR-LMS onde foi verificado por meio de simulações em computador que este algoritmo apresenta desempenho favorável em relação a outros algoritmos do tipo LMS normalizados quando o sinal de entrada é correlacionado. Para este algoritmo, a análise de convergência na média quadrática foi apresentada para sinais de entrada brancos bem como sua extensão ao caso de

sinal de entrada colorido. Um modelo simples para o vetor sinal de entrada que incorpora simplicidade e tratabilidade à análise das estatísticas de segunda ordem é empregado. Os resultados de simulações validam a análise e seus pressupostos. Conclui-se que o algoritmo proposto (BNDR-LMS) apresenta uma maior velocidade de convergência para sinais de entrada coloridos que os demais algoritmos do tipo LMS com reutilização de dados de similar complexidade computacional e que esta maior eficiência é melhor verificada em ambientes com baixo nível de ruídos.

No Capítulo 3, uma versão com restrições do algoritmo BNDR-LMS é derivada de modo a aplicar este algoritmo no campo das comunicações móveis. Em particular, um exemplo usando este algoritmo no cenário de um receptor móvel usando “direct-sequence code-division multiple access” (DS-CDMA) foi feito e uma otimização do passo foi proposta para cumprir as exigências de convergência rápida e mínimo erro médio-quadrático (MSE). Conclui-se que, em ambientes onde o nível de ruídos (de observação ou de modelagem) é elevado, faz-se necessário o uso de uma largura de passo (“step-size”) variável.

No Capítulo 4, os algoritmos baseados em decomposição QR usando rotações de Givens foram apresentados numa forma tutorial com o objetivo especial de classificar os algoritmos existentes numa estrutura didática. Este capítulo derivou o novo algoritmo FQR_PRI_F e suas relações com os outros membros da família de algoritmos QR rápidos. Os algoritmos previamente propostos foram re-derivados usando a mesma notação. As equações dos quatro algoritmos “fast” QR classificados neste capítulo foram também fornecidas junto com suas descrições algorítmicas detalhadas em apêndice. Pode-se concluir pelo que foi estudado que o algoritmo FQR_PRI_B apresenta o melhor desempenho (em termos de esforço computacional e comportamento numérico) dentre os algoritmos QR rápidos classificados.

O Capítulo 5 apresentou as versões totalmente em treliça dos algoritmos FQR_POS_B e FQR_PRI_B. Os resultados de simulações mostraram que o desempenho das versões em treliça (ou “lattice”) em implementação em precisão finita é comparável aos algoritmos originais.

Finalmente, o Capítulo 6 aborda a análise em precisão finita dos algoritmos QR

rápidos usando atualização de erros de predição regressiva (*a priori* e *a posteriori*). Os resultados de análise em precisão infinita já disponíveis em outros trabalhos são resumidos usando a mesma notação uniforme do resto da tese bem como é dada uma contribuição à análise de erro em ponto fixo com o estudo do comportamento de algumas expressões neste tipo de ambiente de precisão finita.

7.2 Sugestões para Pesquisa Futura

Esta seção dá algumas sugestões para posteriores pesquisas.

Em relação ao algoritmo BNDR-LMS, parece ser interessante a investigação da relação entre este algoritmo e suas possíveis extensões (i-normalizado DR-LMS) até o algoritmo das projeções ortogonais em adição às suas relações com o algoritmo RLS.

Um outro campo promissor e de interesse é o desempenho e análise em precisão finita deste algoritmo.

Do Capítulo 2, ficou claro que o modelo e as análises lá usadas podem ser prontamente estendidas a outros algoritmos que reutilizam dados ainda não considerados no passado devido a excesso de complexidade nas expressões de análise. Esta é outra contribuição deste trabalho que pode ser objeto de pesquisa futura.

Uma outra sugestão é a possível melhoria de desempenho do passo adaptativo se um sinal de entrada colorido for considerado na derivação da seqüência ótima.

Por último, algumas sugestões para futuras investigações relativas aos algoritmos QR rápidos são apontadas. A finalização da análise em precisão finita é um tópico do maior interesse.

A mesma notação e abordagem usadas no Capítulo 4 poderiam também ser estendidas para derivar os algoritmos QR em treliça [21, 22, 48] e verificar suas relações com os algoritmos RLS em treliça.

Algumas variantes dos algoritmos FQR (alguns deles já desenvolvidos e disponíveis na literatura técnica) também requerem alguma atenção tais como versões multicanal e com restrições e suas aplicações.

Referências Bibliográficas

- [1] HAYKIN, S., *Adaptive Filter Theory*. 2 ed. Englewood Cliffs, NJ, USA, Prentice Hall, 1991.
- [2] DINIZ, P. S. R. *Adaptive Filtering: Algorithms and Practical Implementation*. Norwell, MA, USA, Kluwer Academic Press, 1997.
- [3] ROY, S., SHYNK, J. J., “Analysis of the data-reusing LMS algorithm”. In: *Proceedings of the 32 Midwest Symposium on Circuits and Systems*, pp. 1127–1130, Urbana-Champaign, IL, USA, 1989.
- [4] SLOCK, D. T., “On the convergence behavior of the LMS and the normalized LMS algorithms”, *IEEE Transactions on Signal Processing*, v. 41, pp. 2811–2825, September 1993.
- [5] JENKINS, W. K, HULL, A. W., STRAIT, J. C. et al., *Advanced Concepts in Adaptive Signal Processing*. Norwell, MA, USA, Kluwer Academic Publishers, 1996.
- [6] DINIZ, P. S. R., DE CAMPOS, M. L. R., ANTONIOU, A., “Analysis of LMS-Newton adaptive filtering algorithms with variable convergence factor”, *IEEE Transactions on Signal Processing*, v. 43, pp. 617–627, March 1995.
- [7] YASSA, F. F., “Optimality in the choice of the convergence factor for gradient-based adaptive algorithms”, *IEEE Transactions on Acoustic, Speech, and Signal Processing*, v. ASSP-35, pp. 48–59, January 1987.
- [8] LEE, D. L., MORF, M., FRIEDLANDER, B., “Recursive least squares ladder estimation algorithms”, *IEEE Transactions on Acoustic, Speech, and Signal Processing*, v. ASSP-29, pp. 627–641, June 1981.
- [9] CIOFFI, J. M., KAILATH, T., “Fast, recursive-least-squares transversal filters for adaptive filtering”, *IEEE Transactions on Acoustic, Speech, and Signal Processing*, v. ASSP-32, pp. 302–337, April 1984.
- [10] SCHNAUFER, B. A., *Practical Techniques for Rapid and Reliable Real-Time Adaptive Filtering*. PhD thesis, University of Illinois at Urbana-Champaign, Urbana-Champaign, IL, USA, 1995.

- [11] WIDROW, B., STEARNS, S. D., *Adaptive Signal Processing*. Englewood-Cliffs, NJ, USA, Prentice-Hall, 1985.
- [12] GOODWIN, G. C., SIN, S. K., *Adaptive Filtering Prediction and Control*. Englewood-Cliffs, NJ, USA, Prentice-Hall, 1984.
- [13] CIOFFI, J. M., “The fast Householder filters RLS adaptive filter”. In: *Proceedings of the IEEE International Conference on Acoustic, Speech, and Signal Processing*, pp. 1619–1622, Albuquerque, NM, USA, 1990.
- [14] LIU, K. J. R., HSIEH, S.-F., YAO, K., “Systolic block Householder transformation for RLS algorithm with two-level pipelined implementation”, *IEEE Transactions on Signal Processing*, v. 40, pp. 946–957, April 1992.
- [15] ALEXANDER, S. T., GHIRNIKAR, A. L., “A method for recursive least squares adaptive filtering based upon an inverse QR decomposition”, *IEEE Transactions on Signal Processing*, v. SP-41, pp. 20–30, January 1993.
- [16] CIOFFI, J. M., “The fast adaptive ROTOR’s RLS algorithm”, *IEEE Transactions on Acoustic, Speech, and Signal Processing*, v. ASSP-38, pp. 631–653, April 1990.
- [17] REGALIA, P. A., BELLANGER, M. G., “On the duality between fast QR methods and lattice methods in least squares adaptive filtering”, *IEEE Transactions on Signal Processing*, v. SP-39, pp. 879–891, April 1991.
- [18] MIRANDA, M. D., GERKEN, M., “A hybrid QR-lattice least squares algorithm using a priori errors”. In: *Proceedings of the 38 Midwest Symposium on Circuits and Systems*, pp. 983–986, Rio de Janeiro, RJ, Brazil, August 1995.
- [19] RONTOGIANNIS, A. A., THEODORIDIS, S., “New fast inverse QR least squares adaptive algorithms”. In: *Proceedings of the IEEE International Conference on Acoustic, Speech, and Signal Processing*, pp. 1412–1415, Detroit, MI, USA, 1995.
- [20] APOLINÁRIO JR., J. A., DINIZ, P. S. R., “A new fast QR algorithm based on a priori errors”, *IEEE Signal Processing Letters*, v. 4, pp. 307–309, November 1997.
- [21] LING, F., “Givens rotation based least squares lattice and related algorithms”, *IEEE Transactions on Signal Processing*, v. SP-39, pp. 1541–1551, July 1991.
- [22] PROUDLER, I. K., MCWHIRTER, J. G., SHEPARD, T. J., “Computationally efficient QR decomposition approach to least squares adaptive filtering”, *IEE Proceedings-F*, v. 138, pp. 341–353, August 1991.
- [23] APOLINÁRIO JR., J. A., DE CAMPOS, M. L. R., DINIZ, P. S. R., “The bi-normalized data-reusing LMS algorithm”, submitted to the *IEEE Transactions on Signal Processing*, 1998.

- [24] APOLINÁRIO JR., J. A., DE CAMPOS, M. L. R., DINIZ, P. S. R., “The binormalized data-reusing LMS algorithm”. In: *Anais do XV Simpósio Brasileiro de Telecomunicações*, pp. 77–80, Recife, PE, Brasil, Setembro 1997.
- [25] APOLINÁRIO JR., J. A., DE CAMPOS, M. L. R., DINIZ, P. S. R., “Convergence analysis of the binormalized data-reusing LMS algorithm”. In: *Proceedings of the European Conference on Circuit Theory and Design*, pp. 972–977, Budapest, Hungary, September 1997.
- [26] MAZO, J. E., “On the independence theory of equalizer convergence”, *The Bell System Technical Journal*, v. 58, pp. 962–993, May–June 1979.
- [27] DE CAMPOS, M. L. R., ANTONIOU, A., “A new quasi-Newton adaptive filtering algorithm”, *IEEE Transactions on Circuits and Systems — Part II*, v. 44, pp. 924–934, November 1997.
- [28] D. T. SLOCK, “On the convergence behavior of the LMS and NLMS algorithms”. In: *V European Signal Processing Conference*, (Barcelona, Spain), pp. 197–200, 1990.
- [29] PAPOULIS, A., *Probability, Random Variables, and Stochastic Processes*. 3 ed., McGraw-Hill, 1991.
- [30] FROST, III, O. L., “An algorithm for linearly constrained adaptive array processing”, *Proceedings of the IEEE*, v. 60, pp. 926–935, August 1972.
- [31] GRIFFITHS, L. J., JIM, C. W., “An alternative approach to linearly constrained adaptive beamforming”, *IEEE Transactions on Antennas and Propagation*, v. AP-30, pp. 27–34, January 1982.
- [32] RESENDE, L. S., ROMANO, J. M. T., BELLANGER, M. G., “A fast least-squares algorithm for linearly constrained adaptive filtering”, *IEEE Transactions on Signal Processing*, v. 44, pp. 1168–1174, May 1996.
- [33] APOLINÁRIO JR., J. A., DINIZ, P. S. R., LAAKSO, T. I. et al., “Step-size optimization of the BNDR-LMS algorithm”. (accepted) *IX European Signal Processing Conference*, Island of Rhodes, Greece, August 1998.
- [34] APOLINÁRIO JR., J. A., WERNER, S., DINIZ, P. S. R. et al., “Constrained normalized adaptive filters for CDMA mobile communications”. (accepted) *IX European Signal Processing Conference*, Island of Rhodes, Greece, August 1998.
- [35] SCHLEGEL, C., ROY, S., ALEXANDER, P. D. et al., “Multiuser projection receivers”, *IEEE Journal on Selected Areas in communications*, v. 14, October 1996.
- [36] GOLUB, G. H., VAN LOAN, C. F., *Matrix Computations*. 2 ed. Baltimore, MD, USA, John Hopkins University Press, 1989.

- [37] MIRANDA, M. D., *Sobre algoritmos dos mínimos quadrados rápidos, recursivos na ordem, que utilizam triangularização ortogonal*. Tese de doutorado, Universidade de São Paulo, São Paulo, SP, Brasil, 1996.
- [38] PAN, C.-T., PLEMMONS, R. J., “Least squares modifications with inverse factorizations: parallel implications”, *Journal of Computational and Applied Mathematics*, v. 27, pp. 109–127, 1989.
- [39] PARK, P. G., KAILATH, T., “A lattice algorithm dual to the extended inverse QR algorithm”, *Signal Processing*, v. 47, pp. 115–133, November 1995.
- [40] MIRANDA, M. D., GERKEN, M., “A hybrid least squares QR-lattice algorithm using a priori errors”, *IEEE Transactions on Signal Processing*, v. 45, pp. 2900–2911, December 1997.
- [41] BELLANGER, M. G., “The FLS-QR algorithm for adaptive filtering”, *Signal Processing*, v. 17, pp. 291–304, August 1989.
- [42] TERRÉ, M., BELLANGER, M. G., “A systolic QRD-based algorithm for adaptive filtering and its implementation”. In: *Proceedings of the IEEE International Conference on Acoustic, Speech, and Signal Processing*, Adelaide, Australia, 1994.
- [43] MIRANDA, M. D., AGUAYO, L., GERKEN, M., “Performance of the a priori and a posteriori QR-LSL algorithms in a limited precision environment”. In: *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, Munich, Germany, April 1997.
- [44] DINIZ, P. S. R., SIQUEIRA, M. G., “Fixed-point error analysis of the QR-recursive least square algorithm”, *IEEE Transactions on Circuits and Systems — Part II*, v. 42, pp. 334–348, May 1995.
- [45] SIQUEIRA, M. G., DINIZ, P. S. R., ALWAN, A., “Infinite precision analysis of the fast QR decomposition RLS algorithm”. In: *Proceedings of the IEEE International Symposium on Circuits and Systems*, London, UK, 1994.
- [46] SAMSON, C., REDDY, V., “Fixed point error analysis of the normalized ladder algorithm”, *IEEE Transactions on Acoustic, Speech, and Signal Processing*, v. 31, pp. 1177–1191, October 1983.
- [47] HUBING, N. E., ALEXANDER, S. T., “Statistical analysis of initialization methods for RLS adaptive filters”, *IEEE Transactions on Signal Processing*, v. 39, pp. 1793–1804, August 1991.
- [48] DESBOUVRIES, F., REGALIA, P. A., “A minimal, Givens rotation based FRLS lattice algorithm”. In: *Proceedings of the VIII European Signal Processing Conference*, 1996.

Apêndice A

1. Equação (2.24):

$$\begin{aligned}
 & E [\mathbf{x}^T(k)\mathbf{x}(k-1)\mathbf{x}^T(k-1)\mathbf{x}(k)] \\
 &= E \left[\sum_{i=0}^N \sum_{j=0}^N x(k-i)x(k-1-i)x(k-1-j)x(k-j) \right] \\
 &= \sum_{i=0}^N E [x^2(k-i)x^2(k-1-i)] \\
 &= (N+1)(\sigma_x^2)^2
 \end{aligned} \tag{A.1}$$

2. Equação (2.25):

$$\begin{aligned}
 & E \left\{ \|\mathbf{x}(k)\|^2 \|\mathbf{x}(k-1)\|^2 - [\mathbf{x}^T(k)\mathbf{x}(k-1)]^2 \right\} \\
 &= E \left[\sum_{i=0}^N \sum_{j=0}^N x^2(k-i)x^2(k-1-j) \right] - E \left\{ [\mathbf{x}^T(k)\mathbf{x}(k-1)]^2 \right\} \\
 &= \sum_{i=0}^N E [x^2(k-i)] \sum_{j=0, j \neq i-1}^N E [x^2(k-1-j)] \\
 &\quad + \sum_{i=1}^N E [x^4(k-i)] - E \left\{ [\mathbf{x}^T(k)\mathbf{x}(k-1)]^2 \right\} \\
 &= (N^2 + N + 1)(\sigma_x^2)^2 + NE[x^4(k)] - (N+1)(\sigma_x^2)^2
 \end{aligned} \tag{A.2}$$

Para sinais estacionários com distribuição Gaussiana, usando o teorema de fatorização do quarto momento, temos $E[x^4(k)] = 3(\sigma_x^2)^2$ [29] e, portanto,

$$E \left\{ \|\mathbf{x}(k)\|^2 \|\mathbf{x}(k-1)\|^2 - [\mathbf{x}^T(k)\mathbf{x}(k-1)]^2 \right\} = N(N+3)(\sigma_x^2)^2 \tag{A.3}$$

3. Equação (2.26):

Seja $[\cdot]_{ij}$ o elemento (i, j) da matriz $[\cdot]$, então

$$\begin{aligned}
 & E \left\{ [\mathbf{x}(k-1)\mathbf{x}^T(k-1)\mathbf{x}(k)\mathbf{x}^T(k)]_{ij} \right\} \\
 &= E \left[x(k-1-i)x(k-j) \sum_{l=0}^N x(k-1-l)x(k-l) \right]
 \end{aligned} \tag{A.4}$$

Para sinais com distribuição Gaussiana podemos usar o teorema da fatorização do quarto momento para obter

$$\begin{aligned}
& E \left\{ [\mathbf{x}(k-1)\mathbf{x}^T(k-1)\mathbf{x}(k)\mathbf{x}^T(k)]_{ij} \right\} \\
&= \sum_{l=0}^N \{ E[x(k-1-i)x(k-j)] E[x(k-1-l)x(k-l)] \\
&\quad + E[x(k-1-i)x(k-1-l)] E[x(k-j)x(k-l)] \\
&\quad + E[x(k-1-i)x(k-l)] E[x(k-j)x(k-1-l)] \} \quad (\text{A.5}) \\
&= \begin{cases} (\sigma_x^2)^2, & i = j \text{ ou } i = j - 2 \\ 0, & \text{caso contrário} \end{cases}
\end{aligned}$$

4. Equação (2.27):

Uma vez mais, usando o fato de que para sinais estacionários com distribuição Gaussiana $E[x^4(k)] = 3(\sigma_x^2)^2$ [29], nós temos

$$\begin{aligned}
E \left\{ [\|\mathbf{x}(k-1)\|^2 \mathbf{x}(k)\mathbf{x}^T(k)]_{ij} \right\} &= E \left[\sum_{l=0}^N x^2(k-1-l)x(k-i)x(k-j) \right] \\
&= \begin{cases} N(\sigma_x^2)^2 + E[x^4(k)], & i = j \\ 0, & \text{caso contrário} \end{cases} \\
&= \begin{cases} (N+3)(\sigma_x^2)^2, & i = j \\ 0, & \text{caso contrário} \end{cases} \quad (\text{A.6})
\end{aligned}$$

Apêndice B

1. Equação (2.39):

Na derivação de (2.39), $\mathbf{x}(k)$ e $\mathbf{x}(k-1)$ foram substituídos por $s_k r_k V_k$ e $s_{k-1} r_{k-1} V_{k-1}$, respectivamente, com $V_k \perp V_{k-1}$. Portanto, $\mathbf{x}^T(k)\mathbf{x}(k-1) = 0$. Além disso, uma aproximação de segunda ordem para $E[1/r_k^2]$ foi usada [4], i.e.,

$$E \left[\frac{1}{\|\mathbf{x}(k)\|^2} \right] = E \left[\frac{1}{\|\mathbf{x}(k-1)\|^2} \right] = E \left[\frac{1}{r_k^2} \right] \approx \frac{1}{(N+2-\nu_x)\sigma_x^2} \quad (\text{B.1})$$

onde ν_x é a *kurtosis* do sinal de entrada.

Para $\mathbf{R} = \sigma_x^2 \mathbf{I}$, usando (2.33) e (2.34) a expressão para $\Delta\xi(k)$ pode ser rescrita como

$$\begin{aligned} \Delta\xi(k+1) &= \sigma_x^2 \text{tr} \{ \text{cov}[\Delta\mathbf{w}(k+1)] \} \\ &= \sigma_x^2 \text{tr} \{ E [\Delta\mathbf{w}(k+1)\Delta\mathbf{w}^T(k+1)] \} \\ &= \sigma_x^2 \text{tr} (E \{ [\mathbf{I} + \mu\mathbf{A}]\Delta\mathbf{w}(k)\Delta\mathbf{w}^T(k)[\mathbf{I} + \mu\mathbf{A}] \}) \\ &\quad + \sigma_x^2 \text{tr} (E \{ \mu[\mathbf{I} + \mu\mathbf{A}]\Delta\mathbf{w}(k)\mathbf{b}^T \}) \\ &\quad + \sigma_x^2 \text{tr} (E \{ \mu\mathbf{b}\Delta\mathbf{w}^T(k)[\mathbf{I} + \mu\mathbf{A}] \}) \\ &\quad + \sigma_x^2 \text{tr} (E [\mu^2\mathbf{b}\mathbf{b}^T]) \\ &= \rho_1 + \rho_2 + \rho_3 + \rho_4 \end{aligned} \quad (\text{B.2})$$

Avaliando cada um destes termos separadamente, obtemos

$$\begin{aligned}
\rho_1 &= \sigma_x^2 \text{tr} \left(E \left\{ [\mathbf{I} + \mu \mathbf{A}] \Delta \mathbf{w}(k) \Delta \mathbf{w}^T(k) [\mathbf{I} + \mu \mathbf{A}] \right\} \right) \\
&= \sigma_x^2 \text{tr} \{ \text{cov}[\Delta \mathbf{w}(k)] \} \\
&\quad - \mu \sigma_x^2 \text{tr} \left\{ E \left[\frac{\mathbf{x}(k-1) \mathbf{x}^T(k-1) \Delta \mathbf{w}(k) \Delta \mathbf{w}^T(k)}{\|\mathbf{x}(k-1)\|^2} \right] \right\} \\
&\quad - \mu \sigma_x^2 \text{tr} \left\{ E \left[\frac{\mathbf{x}(k) \mathbf{x}^T(k) \Delta \mathbf{w}(k) \Delta \mathbf{w}^T(k)}{\|\mathbf{x}(k)\|^2} \right] \right\} \\
&\quad - \mu \sigma_x^2 \text{tr} \left\{ E \left[\frac{\Delta \mathbf{w}(k) \Delta \mathbf{w}^T(k) \mathbf{x}(k-1) \mathbf{x}^T(k-1)}{\|\mathbf{x}(k-1)\|^2} \right] \right\} \\
&\quad - \mu \sigma_x^2 \text{tr} \left\{ E \left[\frac{\Delta \mathbf{w}(k) \Delta \mathbf{w}^T(k) \mathbf{x}(k) \mathbf{x}^T(k)}{\|\mathbf{x}(k)\|^2} \right] \right\} \\
&\quad + \mu^2 \sigma_x^2 \text{tr} \left(E \left\{ \frac{\mathbf{x}(k-1) \mathbf{x}^T(k-1) \Delta \mathbf{w}(k) \Delta \mathbf{w}^T(k) \mathbf{x}(k-1) \mathbf{x}^T(k-1)}{[\|\mathbf{x}(k-1)\|^2]^2} \right\} \right) \\
&\quad + \mu^2 \sigma_x^2 \text{tr} \{ E [\mathbf{x}(k) \mathbf{x}^T(k) \Delta \mathbf{w}(k) \Delta \mathbf{w}^T(k) \mathbf{x}(k-1) \mathbf{x}^T(k-1)] \} \\
&\quad + \mu^2 \sigma_x^2 \text{tr} \{ E [\mathbf{x}(k-1) \mathbf{x}^T(k-1) \Delta \mathbf{w}(k) \Delta \mathbf{w}^T(k) \mathbf{x}(k) \mathbf{x}^T(k)] \} \\
&\quad + \mu^2 \sigma_x^2 \text{tr} \left(E \left\{ \frac{\mathbf{x}(k) \mathbf{x}^T(k) \Delta \mathbf{w}(k) \Delta \mathbf{w}^T(k) \mathbf{x}(k) \mathbf{x}^T(k)}{[\|\mathbf{x}(k)\|^2]^2} \right\} \right) \\
&= \psi_1 + \psi_2 + \dots + \psi_9
\end{aligned} \tag{B.3}$$

onde

$$\begin{aligned}
\psi_1 &= \sigma_x^2 \text{tr} \{ \text{cov}[\Delta \mathbf{w}(k)] \} \\
&= \Delta \xi(k)
\end{aligned} \tag{B.4}$$

$$\begin{aligned}
\psi_2 &= -\mu\sigma_x^2 E \left\{ \text{tr} \left[\frac{\mathbf{x}(k-1)\mathbf{x}^T(k-1)\Delta\mathbf{w}(k)\Delta\mathbf{w}^T(k)}{\|\mathbf{x}(k-1)\|^2} \right] \right\} \\
&= -\mu\sigma_x^2 E \left[\frac{\mathbf{x}^T(k-1)\Delta\mathbf{w}(k)\Delta\mathbf{w}^T(k)\mathbf{x}(k-1)}{\|\mathbf{x}(k-1)\|^2} \right] \\
&= -\mu\sigma_x^2 E \left\{ \frac{[\mathbf{x}^T(k-1)\Delta\mathbf{w}(k)]^2}{\|\mathbf{x}(k-1)\|^2} \right\} \\
&= -\mu\sigma_x^2 E \left\{ \frac{[(1-\mu)\mathbf{x}^T(k-1)\Delta\mathbf{w}(k-1) + \mu n(k-1)]^2}{\|\mathbf{x}(k-1)\|^2} \right\} \\
&= -\mu(1-\mu)^2\sigma_x^2 E [\Delta\mathbf{w}^T(k-1)\mathbf{V}_{k-1}\mathbf{V}_{k-1}^T\Delta\mathbf{w}(k-1)] - \mu^3\sigma_x^2 E \left[\frac{n^2(k-1)}{r_k^2} \right] \\
&= -\frac{\mu(1-\mu)^2\sigma_x^2}{(N+1)} \text{tr} \{ \text{cov} [\Delta\mathbf{w}(k-1)] \} - \mu^3\sigma_n^2\sigma_x^2 E \left[\frac{1}{r_k^2} \right] \\
&= -\frac{\mu(1-\mu)^2}{(N+1)} \Delta\xi(k-1) - \frac{\mu^3\sigma_n^2}{(N+2-\nu_x)}
\end{aligned} \tag{B.5}$$

$$\begin{aligned}
\psi_3 &= -\mu\sigma_x^2 \text{tr} \left\{ E \left[\frac{\mathbf{x}(k)\mathbf{x}^T(k)\Delta\mathbf{w}(k)\Delta\mathbf{w}^T(k)}{\|\mathbf{x}(k)\|^2} \right] \right\} \\
&= -\mu\sigma_x^2 \text{tr} \left\{ E [\mathbf{V}_k\mathbf{V}_k^T\Delta\mathbf{w}(k)\Delta\mathbf{w}^T(k)] \right\} \\
&= -\frac{\mu\sigma_x^2}{N+1} \text{tr} \{ \text{cov} [\Delta\mathbf{w}(k)] \} \\
&= -\frac{\mu}{(N+1)} \Delta\xi(k)
\end{aligned} \tag{B.6}$$

Lembrando que $\text{tr}[\mathbf{A}\mathbf{B}] = \text{tr}[\mathbf{B}\mathbf{A}]$ para quaisquer matrizes quadradas \mathbf{A} e \mathbf{B} , encontramos que

$$\psi_4 = \psi_2 \tag{B.7}$$

$$\psi_5 = \psi_3 \tag{B.8}$$

$$\begin{aligned}
\psi_6 &= \mu^2 \sigma_x^2 \text{tr} \left(E \left\{ \frac{\mathbf{x}(k-1) \mathbf{x}^T(k-1) \Delta \mathbf{w}(k) \Delta \mathbf{w}^T(k) \mathbf{x}(k-1) \mathbf{x}^T(k-1)}{[\|\mathbf{x}(k-1)\|^2]^2} \right\} \right) \\
&= \mu^2 \sigma_x^2 E \left\{ \frac{\Delta \mathbf{w}^T(k) \mathbf{x}(k-1) \mathbf{x}^T(k-1) \mathbf{x}(k-1) \mathbf{x}^T(k-1) \Delta \mathbf{w}(k)}{[\|\mathbf{x}(k-1)\|^2]^2} \right\} \\
&= \mu^2 \sigma_x^2 E \left[\frac{\Delta \mathbf{w}^T(k) \mathbf{x}(k-1) \mathbf{x}^T(k-1) \Delta \mathbf{w}(k)}{\|\mathbf{x}^T(k-1)\|^2} \right] \\
&= -\mu \psi_2
\end{aligned} \tag{B.9}$$

$$\begin{aligned}
\psi_7 &= \mu^2 \sigma_x^2 \text{tr} \{ E[\mathbf{x}(k) \mathbf{x}^T(k) \Delta \mathbf{w}(k) \Delta \mathbf{w}^T(k) \mathbf{x}(k-1) \mathbf{x}^T(k-1)] \} \\
&= \mu^2 \sigma_x^2 \text{tr} \{ E[\mathbf{x}(k-1) \mathbf{x}^T(k-1) \mathbf{x}(k) \mathbf{x}^T(k) \Delta \mathbf{w}(k) \Delta \mathbf{w}^T(k)] \} \\
&= 0
\end{aligned} \tag{B.10}$$

$$\begin{aligned}
\psi_8 &= \mu^2 \sigma_x^2 \text{tr} \{ E[\mathbf{x}(k-1) \mathbf{x}^T(k-1) \Delta \mathbf{w}(k) \Delta \mathbf{w}^T(k) \mathbf{x}(k) \mathbf{x}^T(k)] \} \\
&= \mu^2 \sigma_x^2 \text{tr} \{ E[\mathbf{x}(k) \mathbf{x}^T(k) \mathbf{x}(k-1) \mathbf{x}^T(k-1) \Delta \mathbf{w}(k) \Delta \mathbf{w}^T(k)] \} \\
&= 0
\end{aligned} \tag{B.11}$$

$$\begin{aligned}
\psi_9 &= \mu^2 \sigma_x^2 \text{tr} \left(E \left\{ \frac{\mathbf{x}(k) \mathbf{x}^T(k) \Delta \mathbf{w}(k) \Delta \mathbf{w}^T(k) \mathbf{x}(k) \mathbf{x}^T(k)}{[\|\mathbf{x}(k)\|^2]^2} \right\} \right) \\
&= \mu^2 \sigma_x^2 E \left\{ \frac{\Delta \mathbf{w}^T(k) \mathbf{x}(k) \mathbf{x}^T(k) \mathbf{x}(k) \mathbf{x}^T(k) \Delta \mathbf{w}(k)}{[\|\mathbf{x}(k)\|^2]^2} \right\} \\
&= \mu^2 \sigma_x^2 E \left[\frac{\Delta \mathbf{w}^T(k) \mathbf{x}(k) \mathbf{x}^T(k) \Delta \mathbf{w}(k)}{\|\mathbf{x}(k)\|^2} \right] \\
&= \mu^2 \sigma_x^2 E [\Delta \mathbf{w}^T(k) \mathbf{V}_k \mathbf{V}_k^T \Delta \mathbf{w}(k)] \\
&= \mu^2 \sigma_x^2 \frac{\text{tr} \{ \text{cov} [\Delta \mathbf{w}(k)] \}}{N+1} \\
&= \frac{\mu^2}{N+1} \Delta \xi(k)
\end{aligned} \tag{B.12}$$

Portanto,

$$\begin{aligned}
\rho_1 &= \sigma_x^2 \text{tr} (E \{ [\mathbf{I} + \mu \mathbf{A}] \Delta \mathbf{w}(k) \Delta \mathbf{w}^T(k) [\mathbf{I} + \mu \mathbf{A}] \}) \\
&= \left(1 + \frac{\mu(\mu-2)}{N+1} \right) \Delta \xi(k) + \frac{(1-\mu)^2 \mu(\mu-2)}{N+1} \Delta \xi(k-1) + \frac{\mu^3(\mu-2)}{N+2-\nu_x} \sigma_n^2
\end{aligned} \tag{B.13}$$

De modo similar,

$$\begin{aligned}
\rho_2 &= \mu\sigma_x^2 E \{ \mathbf{b}^T [\mathbf{I} + \mu\mathbf{A}] \Delta \mathbf{w}(k) \} \\
&= \mu\sigma_x^2 E \left[\frac{n(k) \mathbf{x}^T(k) \Delta \mathbf{w}(k)}{\|\mathbf{x}(k)\|^2} \right] \\
&\quad + \mu\sigma_x^2 E \left[\frac{n(k-1) \mathbf{x}^T(k-1) \Delta \mathbf{w}(k)}{\|\mathbf{x}(k-1)\|^2} \right] \\
&\quad - \mu^2 \sigma_x^2 E \left[\frac{n(k) \mathbf{x}^T(k) \Delta \mathbf{w}(k)}{\|\mathbf{x}(k)\|^2} \right] \\
&\quad - \mu^2 \sigma_x^2 E \left[\frac{n(k-1) \mathbf{x}^T(k-1) \Delta \mathbf{w}(k)}{\|\mathbf{x}(k-1)\|^2} \right] \\
&= \mu^2 (1 - \mu) \sigma_x^2 E \left\{ \frac{n(k-1) [(1 - \mu) \mathbf{x}^T(k-1) \Delta \mathbf{w}(k-1) + \mu n(k-1)]}{\|\mathbf{x}(k-1)\|^2} \right\} \\
&= \mu^2 (1 - \mu) \sigma_x^2 E [n^2(k-1)] E \left[\frac{1}{r_{k-1}^2} \right] \\
&= \frac{\mu^2 (1 - \mu)}{(N + 2 - \nu_x)} \sigma_n^2 \\
&= \rho_3
\end{aligned} \tag{B.14}$$

$$\begin{aligned}
\rho_4 &= \mu^2 \sigma_x^2 E \left[\frac{n^2(k)}{\|\mathbf{x}(k)\|^2} + \frac{n^2(k-1)}{\|\mathbf{x}(k-1)\|^2} \right] \\
&= \frac{2\mu^2 \sigma_n^2}{(N + 2 - \nu_x)}
\end{aligned} \tag{B.15}$$

De (B.13)–(B.15) a equação de diferenças para $\Delta\xi(k)$ é finalmente obtida com em (2.39)

$$\begin{aligned}
\Delta\xi(k+1) &= \left[1 + \frac{\mu(\mu-2)}{N+1} \right] \Delta\xi(k) + \frac{(1-\mu)^2 \mu(\mu-2)}{N+1} \Delta\xi(k-1) \\
&\quad + \frac{\mu^2(\mu-2)^2}{N+2-\nu_x} \sigma_n^2
\end{aligned} \tag{B.16}$$

Apêndice C

1. O Algoritmo FQR_POS_F:

FQR_POS_F
<p>Inicialização:</p> <p>$\ \mathbf{e}_f(k) \ = \epsilon = \text{valor positivo e pequeno};$</p> <p>$\mathbf{d}_{fq_2}(k) = \mathbf{d}_{q_2}(k) = \text{zeros}(N + 1, 1);$</p> <p>$\cos\boldsymbol{\theta}(k) = \cos\boldsymbol{\theta}'_b(k) = \text{ones}(N + 1, 1);$</p> <p>$\sin\boldsymbol{\theta}(k) = \sin\boldsymbol{\theta}'_b(k) = \text{zeros}(N + 1, 1);$</p> <p>$\gamma(k) = 1;$</p> <p>$\mathbf{f}(k) = \text{zeros}(N + 1, 1);$</p> <p>for $k = 1, 2, \dots$</p> <p>{ $e_{fq_1}^{(0)}(k + 1) = x(k + 1);$</p> <p style="padding-left: 20px;">for $i = 1 : N + 1$</p> <p style="padding-left: 40px;">{ $e_{fq_1}^{(i)}(k + 1) = \cos\theta_{i-1}(k)e_{fq_1}^{(i-1)}(k) - \sin\theta_{i-1}(k)\lambda^{1/2}d_{fq_2}(k);$</p> <p style="padding-left: 80px;">$d_{fq_2}(k + 1) = \sin\theta_{i-1}(k)e_{fq_1}^{(i-1)}(k) + \cos\theta_{i-1}(k)\lambda^{1/2}d_{fq_2}(k);$</p> <p style="padding-left: 40px;">}</p> <p style="padding-left: 20px;">}</p> <p>$e_{fq_1}(k + 1) = e_{fq_1}^{(N+1)}(k + 1);$</p> <p>$\ \mathbf{e}_f(k + 1) \ = \sqrt{e_{fq_1}^2(k + 1) + \lambda \ \mathbf{e}_f(k) \ ^2};$</p> <p>$\cos\theta_f(k + 1) = \lambda^{1/2} \ \mathbf{e}_f(k) \ / \ \mathbf{e}_f(k + 1) \ ;$</p> <p>$\sin\theta_f(k + 1) = e_{fq_1}(k + 1) / \ \mathbf{e}_f(k + 1) \ ;$</p> <p>$\mathbf{c}(k + 1) = [1; \text{zeros}(N + 1, 1)];$</p> <p>for $i = 1 : N + 1$</p> <p style="padding-left: 20px;">{ $c_{N+3-i} = -\sin\theta'_{b_{N+1-i}}(k)c_1;$</p> <p style="padding-left: 40px;">$c_1 = \cos\theta'_{b_{N+1-i}}(k)c_1;$</p> <p style="padding-left: 20px;">}</p> <p>$\mathbf{c}_{aux} = [0; \mathbf{c}(k + 1)];$</p> <p>for $i = 1 : N + 1$</p> <p style="padding-left: 20px;">{ $oldvalue = c_{aux_{i+1}};$</p> <p style="padding-left: 40px;">$c_{aux_{i+1}} = \sin\theta_{i-1}(k)c_{aux_1} + \cos\theta_{i-1}(k)c_{aux_{i+1}};$</p> <p style="padding-left: 40px;">$c_{aux_1} = \cos\theta_{i-1}(k)c_{aux_1} - \sin\theta_{i-1}(k)oldvalue;$</p> <p style="padding-left: 20px;">}</p> <p>$oldvalue = c_{aux_1};$</p>

(Continuação do Algoritmo FQR_POS_F)

$$\begin{aligned}
& c_{aux_1} = \cos\theta_f(k+1)c_{aux_1} - \sin\theta_f(k+1)c_{aux_{N+3}}; \\
& c_{aux_{N+3}} = \sin\theta_f(k+1)oldvalue + \cos\theta_f(k+1)c_{aux_{N+3}}; \\
& \mathbf{c}(k+1) = \mathbf{c}_{aux}(2 : N+3); \\
& \text{for } i = 1 : N+1 \\
& \quad \{ \\
& \quad \quad oldvalue = c_1; \\
& \quad \quad c_1 = \sqrt{c_1^2 + c_{i+1}^2}; \\
& \quad \quad \cos\theta'_{b_{i-1}}(k+1) = oldvalue/c_1; \\
& \quad \quad \sin\theta'_{b_{i-1}}(k+1) = -c_{i+1}/c_1; \\
& \quad \quad \} \\
& \quad \mathbf{f}^{(N+2)}(k+1) = [\mathbf{f}(k); \sin\theta_f(k+1)\gamma(k)]; \\
& \quad aux_0 = f_1^{(N+2)}(k+1); \\
& \quad \text{for } i = 1 : N+1 \\
& \quad \quad \{ \\
& \quad \quad \quad aux_i = \cos\theta'_{b_{i-1}}(k+1)aux_{i-1} - \sin\theta'_{b_{i-1}}(k+1)f_{i+1}^{(N+2)}(k+1); \\
& \quad \quad \quad f_i(k+1) = \sin\theta'_{b_{i-1}}(k+1)aux_{i-1} + \cos\theta'_{b_{i-1}}(k+1)f_{i+1}^{(N+2)}(k+1); \\
& \quad \quad \quad \} \\
& \quad \quad \gamma^{(0)}(k+1) = 1; \\
& \quad \quad \text{for } i = 1 : N+1 \\
& \quad \quad \quad \{ \\
& \quad \quad \quad \quad \sin\theta_{i-1}(k+1) = f_i(k+1)/\gamma^{(i-1)}(k+1); \\
& \quad \quad \quad \quad \cos\theta_{i-1}(k+1) = \sqrt{1 - \sin^2\theta_{i-1}(k+1)}; \\
& \quad \quad \quad \quad \gamma^{(i)}(k+1) = \cos\theta_{i-1}(k+1)\gamma^{(i-1)}(k+1); \\
& \quad \quad \quad \quad \} \\
& \quad \quad \quad \gamma(k+1) = \gamma^{(N+1)}(k+1); \\
& \quad \quad \quad e_{q_1}^{(0)}(k+1) = d(k+1); \\
& \quad \quad \quad \text{for } i = 1 : N+1 \\
& \quad \quad \quad \quad \{ \\
& \quad \quad \quad \quad \quad e_{q_1}^{(i)}(k+1) = \cos\theta_{i-1}(k+1)e_{q_1}^{(i-1)}(k+1) - \sin\theta_{i-1}(k+1)\lambda^{1/2}d_{q_2_i}(k); \\
& \quad \quad \quad \quad \quad d_{q_2_i}(k+1) = \sin\theta_{i-1}(k+1)e_{q_1}^{(i-1)}(k+1) + \cos\theta_{i-1}(k+1)\lambda^{1/2}d_{q_2_i}(k); \\
& \quad \quad \quad \quad \quad \} \\
& \quad \quad \quad \quad e_{q_1}(k+1) = e_{q_1}^{(N+1)}(k+1); \\
& \quad \quad \quad \quad e(k+1) = e_{q_1}(k+1)\gamma(k+1); \\
& \quad \quad \quad \} \\
& \quad \}
\end{aligned}$$

2. O Algoritmo FQR_PRI_F:

FQR_PRI_F
<p>Inicialização:</p> <p>$\ \mathbf{e}_f(k) \ = \epsilon = \text{valor positivo e pequeno};$</p> <p>$\mathbf{d}_{fq_2}(k) = \mathbf{d}_{q_2}(k) = \text{zeros}(N + 1, 1);$</p> <p>$\cos\boldsymbol{\theta}(k) = \cos\boldsymbol{\theta}'_b(k) = \text{ones}(N + 1, 1);$</p> <p>$\sin\boldsymbol{\theta}(k) = \sin\boldsymbol{\theta}'_b(k) = \text{zeros}(N + 1, 1);$</p> <p>$1/\gamma(k) = 1;$</p> <p>$\mathbf{a}(k) = \text{zeros}(N + 1, 1);$</p> <p>for $k = 1, 2, \dots$</p> <p>{ $e_{fq_1}^{(0)}(k + 1) = x(k + 1);$</p> <p style="padding-left: 20px;">for $i = 1 : N + 1$</p> <p style="padding-left: 40px;">{ $e_{fq_1}^{(i)}(k + 1) = \cos\theta_{i-1}(k)e_{fq_1}^{(i-1)}(k) - \sin\theta_{i-1}(k)\lambda^{1/2}d_{fq_2i}(k);$</p> <p style="padding-left: 40px;">$d_{fq_2i}(k + 1) = \sin\theta_{i-1}(k)e_{fq_1}^{(i-1)}(k) + \cos\theta_{i-1}(k)\lambda^{1/2}d_{fq_2i}(k);$</p> <p style="padding-left: 40px;">}</p> <p style="padding-left: 20px;">}</p> <p style="padding-left: 20px;">$e_{fq_1}(k + 1) = e_{fq_1}^{(N+1)}(k + 1);$</p> <p style="padding-left: 20px;">$e'_f(k + 1) = e_{fq_1}(k + 1)[1/\gamma(k)];$</p> <p style="padding-left: 20px;">$\mathbf{a}^{(N+2)}(k + 1) = [\mathbf{a}(k); \frac{e'_f(k+1)}{\lambda^{1/2}\ \mathbf{e}_f(k)\ }];$</p> <p style="padding-left: 20px;">$aux_0 = a_1^{(N+2)}(k + 1);$</p> <p style="padding-left: 20px;">for $i = 1 : N + 1$</p> <p style="padding-left: 40px;">{ $aux_i = \cos\theta'_{b_{i-1}}(k)aux_{i-1} - \sin\theta'_{b_{i-1}}(k)a_{i+1}^{(N+2)}(k + 1);$</p> <p style="padding-left: 40px;">$a_i(k + 1) = \sin\theta'_{b_{i-1}}(k)aux_{i-1} + \cos\theta'_{b_{i-1}}(k)a_{i+1}^{(N+2)}(k + 1);$</p> <p style="padding-left: 40px;">}</p> <p style="padding-left: 20px;">$\ \mathbf{e}_f(k + 1) \ = \sqrt{e_{fq_1}^2(k + 1) + \lambda \ \mathbf{e}_f(k) \ ^2};$</p> <p style="padding-left: 20px;">$\cos\theta_f(k + 1) = \lambda^{1/2} \ \mathbf{e}_f(k) \ / \ \mathbf{e}_f(k + 1) \ ;$</p> <p style="padding-left: 20px;">$\sin\theta_f(k + 1) = e_{fq_1}(k + 1) / \ \mathbf{e}_f(k + 1) \$</p> <p style="padding-left: 20px;">$\mathbf{c}(k + 1) = [1; \text{zeros}(N + 1, 1)];$</p> <p style="padding-left: 20px;">for $i = 1 : N + 1$</p> <p style="padding-left: 40px;">{ $c_{N+3-i} = -\sin\theta'_{b_{N+1-i}}(k)c_1;$</p> <p style="padding-left: 40px;">$c_1 = \cos\theta'_{b_{N+1-i}}(k)c_1;$</p> <p style="padding-left: 40px;">}</p>

```

caux = [0; c(k + 1)];
for i = 1 : N + 1
{
  oldvalue = cauxi+1;
  cauxi+1 = sinθi-1(k)caux1 + cosθi-1(k)cauxi+1;
  caux1 = cosθi-1(k)caux1 - sinθi-1(k)oldvalue;
}
oldvalue = caux1;
caux1 = cosθf(k + 1)caux1 - sinθf(k + 1)cauxN+3;
cauxN+3 = sinθf(k + 1)oldvalue + cosθf(k + 1)cauxN+3;
c(k + 1) = caux(2 : N + 3);
for i = 1 : N + 1
{
  oldvalue = c1;
  c1 = √(c12 + ci+12);
  cosθ'bi-1(k + 1) = oldvalue/c1;
  sinθ'bi-1(k + 1) = -ci+1/c1;
}
1/γ(0)(k + 1) = 1;
for i = 1 : N + 1
{
  1/γ(i)(k + 1) = √([1/γ(i-1)(k + 1)]2 + ai2(k + 1));
  cosθi-1(k + 1) =  $\frac{1/\gamma^{(i-1)}(k+1)}{1/\gamma^{(i)}(k+1)}$ ;
  sinθi-1(k + 1) =  $\frac{a_i(k+1)}{1/\gamma^{(i)}(k+1)}$ ;
}
γ(k + 1) = 1/[1/γ(N+1)(k + 1)];
eq1(0)(k + 1) = d(k + 1);
for i = 1 : N + 1
{
  eq1(i)(k + 1) = cosθi-1(k + 1)eq1(i-1)(k + 1) - sinθi-1(k + 1)λ1/2dq2i(k);
  dq2i(k + 1) = sinθi-1(k + 1)eq1(i-1)(k + 1) + cosθi-1(k + 1)λ1/2dq2i(k);
}
eq1(k + 1) = eq1(N+1)(k + 1);
e(k + 1) = eq1(k + 1)γ(k + 1);
}

```

3. O Algoritmo FQR_POS_B:

A primeira versão deste algoritmo é baseada no fato de que o último elemento de $\mathbf{f}(k+1)$ (ou $f_{N+1}(k+1) = \frac{x^{(k+1)}}{\|\mathbf{e}_f^{(0)}(k+1)\|}$) é conhecido antes de seu cálculo.

FQR_POS_B - Versão 1
<p>Inicialização:</p> <p>$\epsilon =$ valor positivo e pequeno;</p> <p>$\ \mathbf{e}_f(k)\ = \epsilon$;</p> <p>$\mathbf{d}_{fq2}(k) = \text{zeros}(N+1, 1)$;</p> <p>$\mathbf{d}_{q2}(k) = \text{zeros}(N+1, 1)$;</p> <p>$\cos\boldsymbol{\theta}(k) = \text{ones}(N+1, 1)$;</p> <p>$\sin\boldsymbol{\theta}(k) = \text{zeros}(N+1, 1)$;</p> <p>$\mathbf{f}(k) = \text{zeros}(N+1, 1)$;</p> <p>for $k = 1, 2, \dots$</p> <p>{ $e_{fq1}^{(0)}(k+1) = x(k+1)$;</p> <p> for $i = 1 : N+1$</p> <p> { $e_{fq1}^{(i)}(k+1) = \cos\theta_{i-1}(k)e_{fq1}^{(i-1)}(k+1) - \sin\theta_{i-1}(k)\lambda^{1/2}d_{fq2_{N+2-i}}(k)$;</p> <p> $d_{fq2_{N+2-i}}(k+1) = \sin\theta_{i-1}(k)e_{fq1}^{(i-1)}(k+1) + \cos\theta_{i-1}(k)\lambda^{1/2}d_{fq2_{N+2-i}}(k)$;</p> <p> }</p> <p> $e_{fq1}(k+1) = e_{fq1}^{(N+1)}(k+1)$;</p> <p> $\ \mathbf{e}_f(k+1)\ = \sqrt{e_{fq1}^2(k+1) + \lambda \ \mathbf{e}_f(k)\ ^2}$;</p> <p> $\ \mathbf{e}_f^{(N+1)}(k+1)\ = \ \mathbf{e}_f(k+1)\$;</p> <p> for $i = 1 : N+1$</p> <p> { $\ \mathbf{e}_f^{(N+1-i)}(k+1)\ = \sqrt{\ \mathbf{e}_f^{(N+2-i)}(k+1)\ ^2 + d_{fq2_i}^2(k+1)}$;</p> <p> $\cos\theta_{f_{N+1-i}}(k+1) = \ \mathbf{e}_f^{(N+2-i)}(k+1)\ / \ \mathbf{e}_f^{(N+1-i)}(k+1)\$;</p> <p> $\sin\theta_{f_{N+1-i}}(k+1) = d_{fq2_i}(k+1) / \ \mathbf{e}_f^{(N+1-i)}(k+1)\$;</p> <p> }</p>

(Continuação do Algoritmo FQR_POS_B - Versão 1)

$$\begin{aligned}
 &aux_0 = x(k+1) / \| e_f^{(0)}(k+1) \|; \\
 &f_{N+1}(k+1) = aux_0; \\
 &\text{for } i = 1 : N \\
 &\quad \left\{ \begin{aligned} f_{N+1-i}(k+1) &= \frac{f_{N+2-i}(k) - \sin\theta'_{f_{i-1}}(k+1)aux_{i-1}}{\cos\theta'_{f_{i-1}}(k+1)}; \\ aux_i &= -\sin\theta'_{f_{i-1}}(k+1)f_{N+1-i}(k) + \cos\theta'_{f_{i-1}}(k+1)aux_{i-1}; \end{aligned} \right. \\
 &\quad \left. \right\} \\
 &\quad \gamma^{(0)}(k+1) = 1; \\
 &\quad \text{for } i = 1 : N+1 \\
 &\quad \left\{ \begin{aligned} \sin\theta_{i-1}(k+1) &= f_{N+2-i}(k+1) / \gamma^{(i-1)}(k+1); \\ \cos\theta_{i-1}(k+1) &= \sqrt{1 - \sin^2\theta_{i-1}(k+1)}; \\ \gamma^{(i)}(k+1) &= \cos\theta_{i-1}(k+1)\gamma^{(i-1)}(k+1); \end{aligned} \right. \\
 &\quad \left. \right\} \\
 &\quad \gamma(k+1) = \gamma^{(N+1)}(k+1); \\
 &\quad e_{q_1}^{(0)}(k+1) = d(k+1); \\
 &\quad \text{for } i = 1 : N+1 \\
 &\quad \left\{ \begin{aligned} e_{q_1}^{(i)}(k+1) &= \cos\theta_{i-1}(k+1)e_{q_1}^{(i-1)}(k+1) - \sin\theta_{i-1}(k+1)\lambda^{1/2}d_{q_{2N+2-i}}(k); \\ d_{q_{2N+2-i}}(k+1) &= \sin\theta_{i-1}(k+1)e_{q_1}^{(i-1)}(k+1) + \cos\theta_{i-1}(k+1)\lambda^{1/2}d_{q_{2N+2-i}}(k); \end{aligned} \right. \\
 &\quad \left. \right\} \\
 &\quad e_{q_1}(k+1) = e_{q_1}^{(N+1)}(k+1); \\
 &\quad e(k+1) = e_{q_1}(k+1)\gamma(k+1); \\
 &\quad \left. \right\}
 \end{aligned}$$

A segunda versão do algoritmo FQR_POS_B é baseada na computação direta de $\mathbf{f}(k+1)$ de acordo com (4.84) e requer o cálculo de $\frac{e_f(k+1)}{\|\mathbf{e}_f(k+1)\|}$.

FQR_POS_B - Versão 2
<p>Inicialização:</p> <p>$\epsilon =$ valor positivo e pequeno;</p> <p>$\ \mathbf{e}_f(k)\ = \epsilon$;</p> <p>$\mathbf{d}_{fq2}(k) = \text{zeros}(N+1, 1)$;</p> <p>$\mathbf{d}_{q2}(k) = \text{zeros}(N+1, 1)$;</p> <p>$\cos\boldsymbol{\theta}(k) = \text{ones}(N+1, 1)$;</p> <p>$\sin\boldsymbol{\theta}(k) = \text{zeros}(N+1, 1)$;</p> <p>$\mathbf{f}(k) = \text{zeros}(N+1, 1)$;</p> <p>for $k = 1, 2, \dots$</p> <p>{ $e_{fq1}^{(0)}(k+1) = x(k+1)$;</p> <p style="padding-left: 20px;">for $i = 1 : N+1$</p> <p style="padding-left: 40px;">{ $e_{fq1}^{(i)}(k+1) = \cos\theta_{i-1}(k)e_{fq1}^{(i-1)}(k+1) - \sin\theta_{i-1}(k)\lambda^{1/2}d_{fq2_{N+2-i}}(k)$;</p> <p style="padding-left: 40px; padding-left: 40px;">$d_{fq2_{N+2-i}}(k+1) = \sin\theta_{i-1}(k)e_{fq1}^{(i-1)}(k+1) + \cos\theta_{i-1}(k)\lambda^{1/2}d_{fq2_{N+2-i}}(k)$;</p> <p style="padding-left: 40px;">}</p> <p style="padding-left: 20px;">$e_{fq1}(k+1) = e_{fq1}^{(N+1)}(k+1)$;</p> <p style="padding-left: 20px;">$\ \mathbf{e}_f(k+1)\ = \sqrt{e_{fq1}^2(k+1) + \lambda \ \mathbf{e}_f(k)\ ^2}$;</p> <p style="padding-left: 20px;">$\ \mathbf{e}_f^{(N+1)}(k+1)\ = \ \mathbf{e}_f(k+1)\$;</p> <p style="padding-left: 20px;">for $i = 1 : N+1$</p> <p style="padding-left: 40px;">{ $\ \mathbf{e}_f^{(N+1-i)}(k+1)\ = \sqrt{\ \mathbf{e}_f^{(N+2-i)}(k+1)\ ^2 + d_{fq2_i}^2(k+1)}$;</p> <p style="padding-left: 40px; padding-left: 40px;">$\cos\theta'_{f_{N+1-i}}(k+1) = \ \mathbf{e}_f^{(N+2-i)}(k+1)\ / \ \mathbf{e}_f^{(N+1-i)}(k+1)\$;</p> <p style="padding-left: 40px; padding-left: 40px;">$\sin\theta'_{f_{N+1-i}}(k+1) = d_{fq2_i}(k+1) / \ \mathbf{e}_f^{(N+1-i)}(k+1)\$;</p> <p style="padding-left: 40px;">}</p> <p>}</p>

(Continuação do Algoritmo FQR_POS_B - Versão 2)

$$\begin{aligned}
& aux_0 = \frac{\gamma(k)e_{f_{q_1}}(k+1)}{\|\mathbf{e}_f(k+1)\|}; \\
& \text{for } i = 1 : N + 1 \\
& \quad \{ \begin{aligned} f_{i-1}(k+1) &= \cos\theta'_{f_{N+1-i}}(k+1)f_i(k) - \sin\theta'_{f_{N+1-i}}(k+1)aux_{i-1}; \\ aux_i &= \sin\theta'_{f_{N+1-i}}(k+1)f_i(k) + \cos\theta'_{f_{N+1-i}}(k+1)aux_{i-1}; \end{aligned} \\
& \quad \} \\
& \quad \frac{e_b(k+1)}{\|\mathbf{e}_b(k+1)\|} = f_0(k+1); \\
& \quad f_{N+1}(k+1) = aux_{N+1}; \\
& \quad \gamma^{(0)}(k+1) = 1; \\
& \quad \text{for } i = 1 : N + 1 \\
& \quad \quad \{ \begin{aligned} \sin\theta_{i-1}(k+1) &= f_{N+2-i}(k+1)/\gamma^{(i-1)}(k+1); \\ \cos\theta_{i-1}(k+1) &= \sqrt{1 - \sin^2\theta_{i-1}(k+1)}; \\ \gamma^{(i)}(k+1) &= \cos\theta_{i-1}(k+1)\gamma^{(i-1)}(k+1); \end{aligned} \\
& \quad \quad \} \\
& \quad \gamma(k+1) = \gamma^{(N+1)}(k+1); \\
& \quad e_{q_1}^{(0)}(k+1) = d(k+1); \\
& \quad \text{for } i = 1 : N + 1 \\
& \quad \quad \{ \begin{aligned} e_{q_1}^{(i)}(k+1) &= \cos\theta_{i-1}(k+1)e_{q_1}^{(i-1)}(k+1) - \sin\theta_{i-1}(k+1)\lambda^{1/2}d_{q_{2N+2-i}}(k); \\ d_{q_{2N+2-i}}(k+1) &= \sin\theta_{i-1}(k+1)e_{q_1}^{(i-1)}(k+1) + \cos\theta_{i-1}(k+1)\lambda^{1/2}d_{q_{2N+2-i}}(k); \end{aligned} \\
& \quad \quad \} \\
& \quad \quad e_{q_1}(k+1) = e_{q_1}^{(N+1)}(k+1); \\
& \quad \quad e(k+1) = e_{q_1}(k+1)\gamma(k+1); \\
& \quad \}
\end{aligned}$$

4. O Algoritmo FQR_PRI_B:

A primeira versão deste algoritmo é baseada no fato de que o último elemento de $\mathbf{a}(k+1)$ (ou $a_{N+1}(k+1) = \frac{x(k+1)}{\sqrt{\lambda}\|\mathbf{e}_f^{(0)}(k)\|}$) é conhecido antes do seu cálculo. Esta versão deste algoritmo foi apresentada em [19].

FQR_PRI_B - Versão 1
<p>Inicialização:</p> <p>$\epsilon =$ valor positivo e pequeno;</p> <p>$\ \mathbf{e}_f^{(0)}(k)\ = \epsilon;$</p> <p>$\ \mathbf{e}_f(k)\ = \epsilon;$</p> <p>$\mathbf{d}_{fq2}(k) = \text{zeros}(N+1, 1);$</p> <p>$\mathbf{d}_{q2}(k) = \text{zeros}(N+1, 1);$</p> <p>$\cos\boldsymbol{\theta}(k) = \text{ones}(N+1, 1);$</p> <p>$\cos\boldsymbol{\theta}'_f(k) = \text{ones}(N+1, 1);$</p> <p>$\sin\boldsymbol{\theta}(k) = \text{zeros}(N+1, 1);$</p> <p>$\sin\boldsymbol{\theta}'_f(k) = \text{zeros}(N+1, 1);$</p> <p>$\mathbf{a}(k) = \text{zeros}(N+1, 1);$</p> <p>for $k = 1, 2, \dots$</p> <p>{ $e_{fq1}^{(0)}(k+1) = x(k+1);$</p> <p style="padding-left: 20px;">for $i = 1 : N+1$</p> <p style="padding-left: 40px;">{ $e_{fq1}^{(i)}(k+1) = \cos\theta_{i-1}(k)e_{fq1}^{(i-1)}(k+1) - \sin\theta_{i-1}(k)\lambda^{1/2}d_{fq2_{N+2-i}}(k);$</p> <p style="padding-left: 80px;">$d_{fq2_{N+2-i}}(k+1) = \sin\theta_{i-1}(k)e_{fq1}^{(i-1)}(k+1) + \cos\theta_{i-1}(k)\lambda^{1/2}d_{fq2_{N+2-i}}(k);$</p> <p style="padding-left: 40px;">}</p> <p style="padding-left: 20px;">$e_{fq1}(k+1) = e_{fq1}^{(N+1)}(k+1);$</p> <p style="padding-left: 20px;">$aux_0 = \frac{x(k+1)}{\lambda^{1/2}\ \mathbf{e}_f^{(0)}(k)\ };$</p> <p style="padding-left: 20px;">$a_{N+1}(k+1) = aux_0;$</p> <p style="padding-left: 20px;">for $i = 1 : N$</p> <p style="padding-left: 40px;">{ $a_{N+1-i}(k+1) = \frac{a_{N+2-i}(k) - \sin\theta'_{f_{i-1}}(k)aux_{i-1}}{\cos\theta'_{f_{i-1}}(k)};$</p> <p style="padding-left: 80px;">$aux_i = -\sin\theta'_{f_{i-1}}(k)a_{N+1-i}(k+1) + \cos\theta'_{f_{i-1}}(k)aux_{i-1};$</p> <p style="padding-left: 40px;">}</p> <p style="padding-left: 20px;">$\ \mathbf{e}_f(k+1)\ = \sqrt{e_{fq1}^2(k+1) + \lambda\ \mathbf{e}_f(k)\ ^2};$</p>

$$\begin{aligned}
 & \| \mathbf{e}_f^{(N+1)}(k+1) \| = \| \mathbf{e}_f(k+1) \|; \\
 & \text{for } i = 1 : N + 1 \\
 & \{ \quad \| \mathbf{e}_f^{(N+1-i)}(k+1) \| = \sqrt{\| \mathbf{e}_f^{(N+2-i)}(k+1) \|^2 + d_{fq2_i}^2(k+1)}; \\
 & \quad \cos \theta'_{f_{N+1-i}}(k+1) = \| \mathbf{e}_f^{(N+2-i)}(k+1) \| / \| \mathbf{e}_f^{(N+1-i)}(k+1) \|; \\
 & \quad \sin \theta'_{f_{N+1-i}}(k+1) = d_{fq2_i}(k+1) / \| \mathbf{e}_f^{(N+1-i)}(k+1) \|; \\
 & \} \\
 & 1/\gamma^{(0)}(k+1) = 1; \\
 & \text{for } i = 1 : N + 1 \\
 & \{ \quad 1/\gamma^{(i)}(k+1) = \sqrt{[1/\gamma^{(i-1)}(k+1)]^2 + a_{N+2-i}^2(k+1)}; \\
 & \quad \cos \theta_{i-1}(k+1) = \frac{1/\gamma^{(i-1)}(k+1)}{1/\gamma^{(i)}(k+1)}; \\
 & \quad \sin \theta_{i-1}(k+1) = \frac{a_{N+2-i}(k+1)}{1/\gamma^{(i)}(k+1)}; \\
 & \} \\
 & \gamma(k+1) = 1/[1/\gamma^{(N+1)}(k+1)]; \\
 & e_{q_1}^{(0)}(k+1) = d(k+1); \\
 & \text{for } i = 1 : N + 1 \\
 & \{ \quad e_{q_1}^{(i)}(k+1) = \cos \theta_{i-1}(k+1) e_{q_1}^{(i-1)}(k+1) - \sin \theta_{i-1}(k+1) \lambda^{1/2} d_{q2_{N+2-i}}(k); \\
 & \quad d_{q2_{N+2-i}}(k+1) = \sin \theta_{i-1}(k+1) e_{q_1}^{(i-1)}(k+1) + \cos \theta_{i-1}(k+1) \lambda^{1/2} d_{q2_{N+2-i}}(k); \\
 & \} \\
 & e_{q_1}(k+1) = e_{q_1}^{(N+1)}(k+1); \\
 & e(k+1) = e_{q_1}(k+1) \gamma(k+1); \\
 & \}
 \end{aligned}$$

A segunda versão do algoritmo FQR_PRI_B é baseada na computação direta de $\mathbf{a}(k+1)$ de acordo com (4.85) e requer o cálculo de $\frac{e'_f(k+1)}{\sqrt{\lambda}\|\mathbf{e}_f(k)\|}$. Esta versão deste algoritmo foi apresentada em [18].

FQR_PRI_B - Versão 2
<p>Inicialização:</p> <p>$\epsilon =$ valor positivo e pequeno;</p> <p>$\ \mathbf{e}_f^{(0)}(k)\ = \epsilon;$</p> <p>$\ \mathbf{e}_f(k)\ = \epsilon;$</p> <p>$\mathbf{d}_{fq2}(k) = \text{zeros}(N+1, 1);$</p> <p>$\mathbf{d}_{q2}(k) = \text{zeros}(N+1, 1);$</p> <p>$\cos\boldsymbol{\theta}(k) = \text{ones}(N+1, 1);$</p> <p>$\cos\boldsymbol{\theta}'_f(k) = \text{ones}(N+1, 1);$</p> <p>$\sin\boldsymbol{\theta}(k) = \text{zeros}(N+1, 1);$</p> <p>$\sin\boldsymbol{\theta}'_f(k) = \text{zeros}(N+1, 1);$</p> <p>$\mathbf{a}(k) = \text{zeros}(N+1, 1);$</p> <p>for $k = 1, 2, \dots$</p> <p>{ $e_{fq1}^{(0)}(k+1) = x(k+1);$</p> <p style="padding-left: 20px;">for $i = 1 : N+1$</p> <p style="padding-left: 40px;">{ $e_{fq1}^{(i)}(k+1) = \cos\theta_{i-1}(k)e_{fq1}^{(i-1)}(k+1) - \sin\theta_{i-1}(k)\lambda^{1/2}d_{fq2_{N+2-i}}(k);$</p> <p style="padding-left: 40px;">$d_{fq2_{N+2-i}}(k+1) = \sin\theta_{i-1}(k)e_{fq1}^{(i-1)}(k+1) + \cos\theta_{i-1}(k)\lambda^{1/2}d_{fq2_{N+2-i}}(k);$</p> <p style="padding-left: 40px;">}</p> <p style="padding-left: 20px;">}</p> <p>$e_{fq1}(k+1) = e_{fq1}^{(N+1)}(k+1);$</p> <p>$aux_0 = \frac{e_{fq1}(k+1)}{\gamma(k)\lambda^{1/2}\ \mathbf{e}_f(k)\ };$</p> <p>for $i = 1 : N+1$</p> <p style="padding-left: 20px;">{ $a_{i-1}(k+1) = \cos\theta'_{f_{N+1-i}}(k)a_i(k) - \sin\theta'_{f_{N+1-i}}(k)aux_{i-1};$</p> <p style="padding-left: 20px;">$aux_i = \sin\theta'_{f_{N+1-i}}(k)a_i(k) + \cos\theta'_{f_{N+1-i}}(k)aux_{i-1};$</p> <p style="padding-left: 20px;">}</p> <p>$\frac{e'_b(k+1)}{\lambda^{1/2}\ \mathbf{e}_b(k)\ } = a_0(k+1);$</p> <p>$a_{N+1}(k+1) = aux_{N+1};$</p> <p>$\ \mathbf{e}_f(k+1)\ = \sqrt{e_{fq1}^2(k+1) + \lambda\ \mathbf{e}_f(k)\ ^2};$</p>

$$\begin{aligned}
 & \| \mathbf{e}_f^{(N+1)}(k+1) \| = \| \mathbf{e}_f(k+1) \|; \\
 & \text{for } i = 1 : N + 1 \\
 & \{ \quad \| \mathbf{e}_f^{(N+1-i)}(k+1) \| = \sqrt{\| \mathbf{e}_f^{(N+2-i)}(k+1) \|^2 + d_{fq2_i}^2(k+1)}; \\
 & \quad \cos \theta'_{f_{N+1-i}}(k+1) = \| \mathbf{e}_f^{(N+2-i)}(k+1) \| / \| \mathbf{e}_f^{(N+1-i)}(k+1) \|; \\
 & \quad \sin \theta'_{f_{N+1-i}}(k+1) = d_{fq2_i}(k+1) / \| \mathbf{e}_f^{(N+1-i)}(k+1) \|; \\
 & \} \\
 & 1/\gamma^{(0)}(k+1) = 1; \\
 & \text{for } i = 1 : N + 1 \\
 & \{ \quad 1/\gamma^{(i)}(k+1) = \sqrt{[1/\gamma^{(i-1)}(k+1)]^2 + a_{N+2-i}^2(k+1)}; \\
 & \quad \cos \theta_{i-1}(k+1) = \frac{1/\gamma^{(i-1)}(k+1)}{1/\gamma^{(i)}(k+1)}; \\
 & \quad \sin \theta_{i-1}(k+1) = \frac{a_{N+2-i}(k+1)}{1/\gamma^{(i)}(k+1)}; \\
 & \} \\
 & \gamma(k+1) = 1/[1/\gamma^{(N+1)}(k+1)]; \\
 & e_{q_1}^{(0)}(k+1) = d(k+1); \\
 & \text{for } i = 1 : N + 1 \\
 & \{ \quad e_{q_1}^{(i)}(k+1) = \cos \theta_{i-1}(k+1) e_{q_1}^{(i-1)}(k+1) - \sin \theta_{i-1}(k+1) \lambda^{1/2} d_{q2_{N+2-i}}(k); \\
 & \quad d_{q2_{N+2-i}}(k+1) = \sin \theta_{i-1}(k+1) e_{q_1}^{(i-1)}(k+1) + \cos \theta_{i-1}(k+1) \lambda^{1/2} d_{q2_{N+2-i}}(k); \\
 & \} \\
 & e_{q_1}(k+1) = e_{q_1}^{(N+1)}(k+1); \\
 & e(k+1) = e_{q_1}(k+1) \gamma(k+1); \\
 & \}
 \end{aligned}$$