

RAFAEL DUTRA DEMETRI

**IMPLEMENTAÇÃO EFICIENTE DO MÉTODO DA
CORRELAÇÃO CRUZADA NORMALIZADA PARA
DETECÇÃO DE *DOUBLE-TALK***

FLORIANÓPOLIS

2009

**UNIVERSIDADE FEDERAL DE SANTA CATARINA
PROGRAMA DE PÓS-GRADUAÇÃO
EM ENGENHARIA ELÉTRICA**

**IMPLEMENTAÇÃO EFICIENTE DO MÉTODO DA
CORRELAÇÃO CRUZADA NORMALIZADA PARA
DETECÇÃO DE *DOUBLE-TALK***

Dissertação submetida à
Universidade Federal de Santa Catarina
como parte dos requisitos para a
obtenção do grau de Mestre em Engenharia Elétrica

RAFAEL DUTRA DEMETRI

Florianópolis, Setembro de 2009

IMPLEMENTAÇÃO EFICIENTE DO MÉTODO DA CORRELAÇÃO CRUZADA NORMALIZADA PARA DETECÇÃO DE *DOUBLE-TALK*

Rafael Dutra Demetri

‘Esta Dissertação foi julgada adequada para obtenção do Título de Mestre em Engenharia Elétrica, Área de Concentração em *Comunicações e Processamento de Sinais*, e aprovada em sua forma final pelo Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Santa Catarina.’

Prof. Rui Seara, Dr.
Orientador

Prof. Roberto de Souza Salgado, Ph.D.
Coordenador do Programa de Pós-Graduação em Engenharia Elétrica

Banca Examinadora:

Prof. Rui Seara, Dr.
Presidente

Prof. Hans Helmut Zürn, Ph.D.

Prof. Walter Pereira Carpes Junior, Dr.

Agradecimentos

Aos meus pais, Carlos e Isabel, por me apoiarem de maneira incondicional durante todas as fases da minha vida.

À minha esposa, Maiara, pelo carinho, incentivo e compreensão ao longo do desenvolvimento deste trabalho.

Ao Prof. Rui Seara, pela orientação durante o mestrado e pela enorme ajuda na redação desta dissertação.

Aos membros da banca, Prof. Hans Helmut Zürn e Prof. Walter Pereira Carpes Junior, pelos valiosos comentários, correções e sugestões.

Aos amigos do LINSE, pela ajuda e pelo companheirismo.

À CAPES, pelo apoio financeiro.

Resumo da Dissertação apresentada à UFSC como parte dos requisitos necessários para a obtenção do grau de Mestre em Engenharia Elétrica

IMPLEMENTAÇÃO EFICIENTE DO MÉTODO DA CORRELAÇÃO CRUZADA NORMALIZADA PARA DETECÇÃO DE *DOUBLE-TALK*

Rafael Dutra Demetri

Setembro/2009

Orientador: Rui Seara, Dr.

Área de Concentração: Comunicações e Processamento de Sinais.

Palavras-chave: Cancelamento de eco, *double-talk*, método da correlação cruzada normalizada.

Número de Páginas: 59.

RESUMO: Este trabalho discute uma implementação eficiente do método da correlação cruzada normalizada (NCC) para detecção de *double-talk* em sistemas de cancelamento de eco. A principal vantagem desse método é obter um limiar de detecção ótimo independente das características do sistema e dos sinais envolvidos. No entanto, sua implementação direta envolve a inversão de uma matriz $N \times N$. Na prática, utilizam-se versões simplificadas desse método, que têm como desvantagem a dependência dos coeficientes do filtro adaptativo utilizado no cancelamento de eco. A abordagem aqui proposta utiliza uma versão de baixa complexidade computacional do algoritmo *recursive least-squares* (RLS) para efetuar uma parte importante dos cálculos, eliminando assim a necessidade de algumas aproximações usualmente consideradas. Assim, a implementação proposta depende apenas do conhecimento dos sinais do *far-end* e do *near-end*, não levando em conta outras variáveis do sistema, tais como os coeficientes do filtro. Uma redução adicional de complexidade é obtida se a detecção de *double-talk* for realizada em uma frequência de amostragem menor do que a do sistema. Resultados de simulação mostram muito bom desempenho da abordagem proposta quando comparada com outras técnicas da literatura.

Abstract of Dissertation presented to UFSC as a partial fulfillment of the requirements for the degree of Master in Electrical Engineering.

EFFICIENT IMPLEMENTATION OF THE NORMALIZED CROSS-CORRELATION METHOD FOR DOUBLE-TALK DETECTION

Rafael Dutra Demetri

September/2009

Advisor: Rui Seara, Dr.

Area of Concentration: Communications and Signal Processing.

Keywords: Echo cancellation, double-talk, normalized cross-correlation method.

Number of Pages: 59.

ABSTRACT: This dissertation presents an efficient implementation of the normalized cross-correlation method (NCC) for double-talk detection in echo cancellation systems. The main advantage of the NCC is to obtain an optimum detection threshold, which is independent of the characteristics of both system and signals involved in the process. However, its direct implementation requires an $N \times N$ matrix inversion. In practice, simplified versions of this method are used, which have as disadvantage the dependence on the adaptive filter coefficients used in the echo cancellation. The approach proposed here uses a low complexity version of the recursive least-squares (RLS) algorithm to perform an important part of the required computations, thereby avoiding some commonly used approximations. Therefore, the proposed implementation relies only on the far-end and near-end signals, not taking into account other variables such as the adaptive filter coefficients. Further computational complexity savings can be obtained by performing double-talk detection in a reduced sampling rate condition. Simulation results point out very good performance for the proposed approach compared with other techniques.

Sumário

Lista de Figuras	viii
Lista de Tabelas	ix
1 Introdução	1
1.1 Eco em Telefonia	1
1.2 Abordagens Atuais de Cancelamento de Eco	2
1.3 Proposta Deste Trabalho	4
1.4 Organização da Dissertação	4
2 Cancelamento de Eco	5
2.1 Tipos de Eco	5
2.1.1 Eco de Linha	5
2.1.2 Eco Acústico	7
2.2 Sistemas de Cancelamento de Eco	8
2.2.1 Detecção de <i>Double-Talk</i>	10
2.3 Considerações	10
3 Algoritmos de Filtragem Adaptativa	11
3.1 Abordagem do Erro Quadrático Médio	13
3.1.1 Filtro de Wiener	13
3.1.2 Cálculo do Filtro de Wiener Através do Método <i>Steepest Descent</i>	15
3.1.3 Algoritmo <i>Least-Mean-Square</i> (LMS)	16
3.2 Abordagem dos Mínimos Quadrados	17
3.2.1 Algoritmo <i>Recursive Least-Squares</i> (RLS)	19
3.2.2 Algoritmo <i>Fast Transversal Filter</i> (FTF)	23
3.2.2.1 Filtro Preditor Linear <i>Forward</i>	24
3.2.2.2 Filtro Preditor Linear <i>Backward</i>	27
3.2.2.3 Adaptação dos Coeficientes	30

3.2.3	Algoritmo <i>Stabilized Fast Transversal Filter</i> (SFTF)	30
3.2.4	Algoritmo <i>Fast Newton Transversal Filter</i> (FNTF)	32
3.3	Considerações	35
4	Detecção de Double-Talk	37
4.1	Detectores de <i>Double-Talk</i>	37
4.2	Algoritmo Geigel	39
4.3	Método da Correlação Cruzada	40
4.4	Método da Coerência Espectral	41
4.5	Método da Correlação Cruzada Normalizada	43
4.6	Resultados de Simulação	45
4.7	Considerações	47
5	Implementação Eficiente do Método da Correlação Cruzada Normalizada	49
5.1	Abordagem Considerada	49
5.2	Implementação do NCC Utilizando o Algoritmo FNTF	51
5.3	Considerações	53
6	Considerações Finais	55
6.1	Sumário e Discussão dos Resultados	55
6.2	Propostas para Trabalhos Futuros	57
	Referências	58

Lista de Figuras

2.1	Conexão dos assintantes às centrais locais.	5
2.2	Conversão de quatro para dois fios.	6
2.3	Sinal de eco em uma ligação telefônica.	6
2.4	Eco acústico em telefone viva-voz.	8
2.5	Diagrama de blocos da geração de eco em uma ligação telefônica.	9
2.6	Diagrama de blocos simplificado de um sistema de cancelamento de eco.	9
3.1	Diagrama em blocos do filtro adaptativo.	12
3.2	Ilustração do procedimento de atualização do ganho de Kalman no FTF.	24
4.1	Estrutura básica de um cancelador de eco.	37
4.2	Influência do fator de esquecimento λ no método da correlação cruzada normalizada para NFR = -5 dB e probabilidade de falso alarme $P_f = 0, 1$	46
4.3	Comparação dos valores de P_m obtidos por diferentes técnicas, considerando uma probabilidade de falso alarme $P_f = 0, 1$	47
5.1	Comparação dos valores de P_m obtidos pela implementação proposta do NCC, com $\lambda_{RLS} = 0,9999$ e $\lambda = 0,995$, e o algoritmo convencional, considerando uma probabilidade de falso alarme $P_f = 0, 1$	51
5.2	Influência da ordem dos preditores N_p , considerando NFR = -5 dB e $P_f = 0, 1$	52
5.3	Comparação dos valores de P_m obtidos pelo NCC considerando o algoritmo FNTF, com $N_p = 0$, em relação ao NCC convencional.	53
5.4	Probabilidade de perda P_m para diferentes frequências de amostragem.	54

Lista de Tabelas

3.1	Algoritmo LMS	17
3.2	Algoritmo RLS	21
3.3	Algoritmo FTF	31
3.4	Algoritmo SFTF	33
3.5	Algoritmo FNTF	35
4.1	Valor do Limiar T Obtido para Diferentes Técnicas	47
5.1	Complexidade Computacional da Implementação Proposta	52

1 Introdução

A invenção do telefone, cerca de 150 anos atrás, foi responsável por estender o alcance da comunicação verbal entre as pessoas muito além do que antes era possível. Desde então, a telefonia se tornou uma parte essencial do cotidiano, permitindo de forma eficiente a comunicação entre indivíduos em diversos lugares do planeta.

A evolução da tecnologia fez com que os sistemas de telecomunicação se expandissem além da telefonia convencional. Os sistemas de telefonia móvel, por exemplo, permitem que os usuários se comuniquem sem a necessidade de conexão física através de fios. Em particular, o sistema de telefonia celular é hoje amplamente disponível para a população em geral.

Um outro tipo de sistema de comunicação que vem sendo cada vez mais utilizado é o de voz sobre IP (VoIP). Nesses sistemas, a estrutura concebida para transmissão de dados é também utilizada para transmitir voz. Assim, os sistemas VoIP são muito utilizados no meio corporativo, pois permitem redução de custos através da utilização de uma única rede física para voz e dados. Devido à crescente disponibilidade da Internet em banda larga, os sistemas VoIP vêm também sendo muito utilizados por usuários em geral, já que permitem a realização de ligações telefônicas através da Internet com custo reduzido.

Independentemente da sua natureza, os sistemas de telefonia apresentam diversos desafios técnicos aos projetistas de tais sistemas. Um desses problemas é a presença de eco nas ligações telefônicas.

1.1 Eco em Telefonia

Em telefonia, eco é o retorno da voz de um dos interlocutores em uma conversação. Em muitos casos, a presença de eco pode impactar fortemente a qualidade de voz percebida pelos usuários. Dessa forma, a procura de soluções que reduzam o desconforto causado pelo eco é de grande interesse das partes envolvidas na prestação de serviços de telefonia e de telecomunicações em geral [1].

Na maioria dos sistemas de comunicação atuais (incluindo telefonia fixa, telefonia

celular e sistemas VoIP), a transmissão dos sinais de voz por longas distâncias é feita na forma digital. Esse tipo de transmissão não causa eco, pois o fluxo de bits em um dos caminhos do circuito digital não “vaza” para o outro sentido. No entanto, nas pontas do circuito, os sinais de voz são convertidos em sinais analógicos. Nesses trechos analógicos, algumas situações podem causar o retorno da voz dos usuários. As fontes mais comuns de eco são:

- Descasamento de impedância na híbrida, que é o equipamento que realiza a conversão entre os circuitos a quatro fios e a dois fios nos sistemas de telefonia analógica.
- Acoplamento acústico entre alto-falantes e microfones, principalmente em sistemas viva-voz.

Em qualquer conversação realizada pessoalmente ou através de telefones, as pessoas escutam a própria voz enquanto falam. Em geral, esse efeito não é incômodo e é uma parte natural da fala. Os usuários percebem a própria voz como eco apenas quando o tempo de retorno da voz é elevado. Na maioria das ligações telefônicas locais esse tempo é pequeno e, portanto, a presença de eco não causa impacto na qualidade da voz percebida pelos usuários. No entanto, existem algumas situações nas quais o atraso entre a voz e o seu respectivo eco é grande. Um exemplo são as ligações internacionais via satélite em que o atraso típico é de 25 ms a 150 ms. Já nos sistemas VoIP, utilizam-se geralmente *buffers* de áudio que visam preservar a qualidade da voz em condições adversas de transmissão de dados na rede. Esses *buffers*, usualmente, causam um atraso da ordem de 150 ms. Nos sistemas de telefonia celular, o atraso total típico é de 160 ms a 200 ms, devido às diversas etapas de codificação de voz envolvidas nesses sistemas. Assim, em todos esses casos, devem ser utilizados canceladores de eco visando melhorar a qualidade da voz e o conforto dos usuários.

1.2 Abordagens Atuais de Cancelamento de Eco

Em geral, a estrutura básica dos sistemas de cancelamento de eco é composta por um filtro adaptativo, um detector de *double-talk* e um mecanismo de controle de adaptação [2].

Em um cancelador de eco, os coeficientes do filtro adaptativo são continuamente modificados de modo a modelar a resposta ao impulso do caminho do eco. Dessa forma, é possível gerar uma réplica do sinal de eco e o cancelamento é obtido através de subtração. Um dos algoritmos de filtragem adaptativa mais utilizados é o LMS (*least-mean-square*), devido principalmente à sua simplicidade, robustez e baixa complexidade computacional

[3]. Em aplicações de cancelamento de eco, utilizam-se geralmente variações do LMS, tais como o *normalized LMS* (NLMS) [3] e o *proportional NLMS* (PNLMS) [4]. Outro algoritmo de filtragem adaptativa usualmente considerado em aplicações práticas é o RLS (*recursive least-squares*), que apresenta velocidade de convergência tipicamente uma ordem de grandeza maior do que a do LMS [3]. Uma desvantagem do RLS em relação ao LMS é a sua elevada complexidade computacional. Existem, no entanto, algumas estratégias de implementação do RLS que apresentam complexidade reduzida, como o FTF (*fast transversal filter*) [5] e o FNFTF (*fast newton transversal filter*) [6].

Independentemente do algoritmo adaptativo utilizado, o desempenho de um cancelador de eco é afetado se a adaptação dos coeficientes é realizada nos instantes em que há atividade de fala nos dois lados da linha (*double-talk*). Dessa forma, deve existir no cancelador de eco um mecanismo de controle de adaptação associado a um detector de *double-talk*. Quando for detectado *double-talk*, o mecanismo de controle deve interromper o processo de adaptação, evitando dessa forma uma degradação na estimativa do caminho do eco.

Devido à sua importância para sistemas de cancelamento de eco, diversos métodos para a detecção de *double-talk* vêm sendo propostos na literatura [7]-[11]. Nesses métodos, em geral, uma variável de detecção é calculada a partir dos sinais de entrada e das variáveis internas do cancelador de eco. Essa variável é então comparada com um limiar de detecção e *double-talk* é assumido quando a variável de detecção é menor do que esse limiar. Assim, uma dificuldade geralmente associada aos algoritmos de detecção de *double-talk* é determinar um limiar de detecção que permita um desempenho adequado considerando diferentes situações e características dos sinais envolvidos.

O método da correlação cruzada normalizada (NCC) para detecção de *double-talk* tem como principal vantagem apresentar um limiar de detecção ótimo não dependente das características do sistema e dos sinais envolvidos [12]. Em sua formulação original, os cálculos envolvem apenas os sinais do *far-end* e *near-end*. A implementação direta desse método, no entanto, é geralmente evitada, pois envolve a inversão de uma matriz (o que pode acarretar elevada complexidade computacional). Logo, na prática, utilizam-se em geral versões simplificadas desse método [12], [13]. Uma desvantagem dessas versões simplificadas é que elas dependem do valor dos coeficientes do filtro adaptativo. Na prática, essa dependência traz uma série de desvantagens. Por exemplo, uma falha na detecção de *double-talk* pode fazer o algoritmo adaptativo divergir, o que por consequência tem impacto na detecção de *double-talk* nos instantes seguintes. Nesses casos, o desempenho do cancelador de eco como um todo pode ser fortemente impactado.

1.3 Proposta Deste Trabalho

Neste trabalho, é apresentado um método eficiente para a implementação do NCC. A chave para esse método advém da constatação de que uma parte importante dos cálculos pode ser efetuada através do algoritmo RLS. Dessa forma, podem-se utilizar versões de baixa complexidade desse algoritmo no cálculo da variável de detecção.

A implementação proposta depende apenas do conhecimento dos sinais do *far-end* e *near-end*. Assim, evitam-se as desvantagens associadas à dependência do filtro adaptativo utilizado no cancelamento de eco.

1.4 Organização da Dissertação

Este trabalho é organizado como segue. O Capítulo 2 apresenta os tipos de eco mais comuns em telefonia e as características gerais dos sistemas de cancelamento de eco. O Capítulo 3 traz uma introdução à filtragem adaptativa, mostrando alguns dos algoritmos mais utilizados em aplicações práticas. O Capítulo 4 discute algumas abordagens utilizadas para a detecção de *double-talk* em sistemas de cancelamento de eco, mostrando as suas características, vantagens e desvantagens. São mostrados também resultados de simulação, comparando o desempenho das técnicas apresentadas. O Capítulo 5 descreve a implementação proposta do método da correlação cruzada normalizada, baseada no algoritmo RLS. São apresentados resultados de simulação que mostram muito bom desempenho da técnica proposta. Finalmente, o Capítulo 6 apresenta as considerações finais do trabalho.

2 Cancelamento de Eco

Neste capítulo, é apresentada uma introdução aos sistemas de cancelamento de eco. São discutidos os tipos mais comuns de eco em telefonia e a estrutura básica de um cancelador de eco.

2.1 Tipos de Eco

Em uma ligação telefônica, os tipos mais frequentes de eco são o *eco de linha* e o *eco acústico* [1], que são brevemente descritos a seguir.

2.1.1 Eco de Linha

O *eco de linha* ocorre em ligações no sistema de telefonia convencional (analógica). Neste sistema, os usuários conectam-se a central local mais próxima através de um circuito com um par de fios. Assim, conforme observado na Figura 2.1, os sinais de voz durante uma ligação telefônica trafegam simultaneamente em ambas direções nesse circuito.

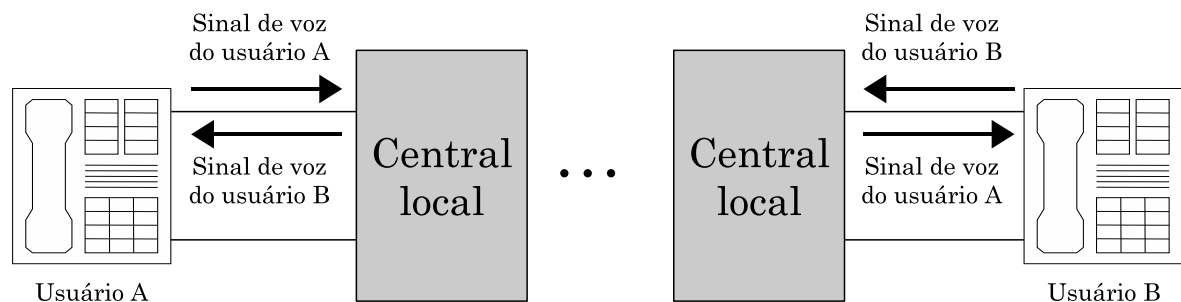


Figura 2.1: Conexão dos assinantes às centrais locais.

A conexão entre as centrais locais não é efetuada através de circuitos a dois fios. Nesses casos, são necessários dois caminhos separados e unidirecionais para os sinais de voz dos usuários devido ao uso de equipamentos unidirecionais como amplificadores e conversores analógico/digital. Assim, as centrais locais devem converter o circuito a dois fios em um conjunto de dois caminhos unidirecionais, chamado circuito a quatro fios. O dispositivo que realiza essa conversão e a separação dos sinais nos dois caminhos é

chamado híbrida. Na Figura 2.2, é mostrada uma ilustração dessa conversão, tomando como exemplo a central local conectada ao usuário B.

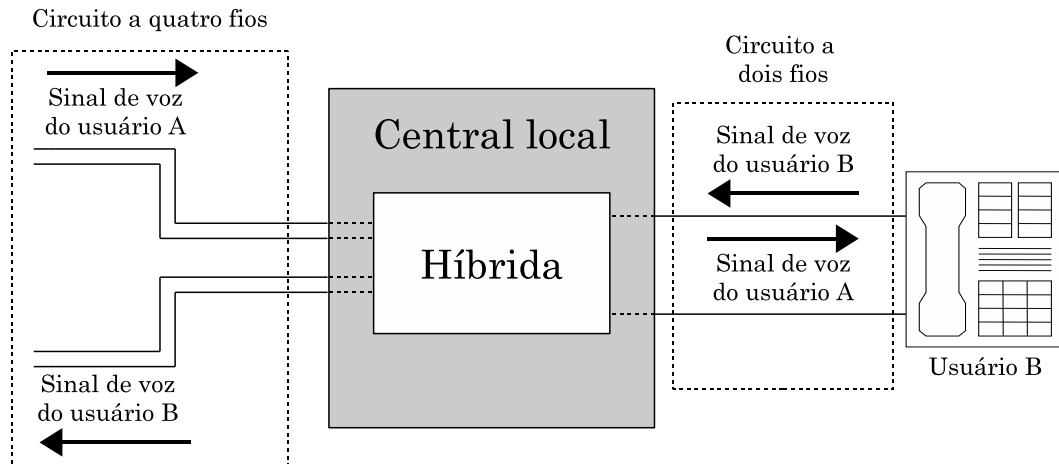


Figura 2.2: Conversão de quatro para dois fios.

Na prática, não se obtém uma perfeita separação dos sinais, principalmente, devido a descasamentos de impedância entre a híbrida e os circuitos envolvidos. Essas imperfeições causam o retorno de parte do sinal de voz dos usuários, causando eco. A Figura 2.3 apresenta como se origina o eco existente no caso particular do exemplo mostrado na Figura 2.2. Nessa situação, uma parcela do sinal de voz do usuário A retorna (eco), somada a voz do usuário B.

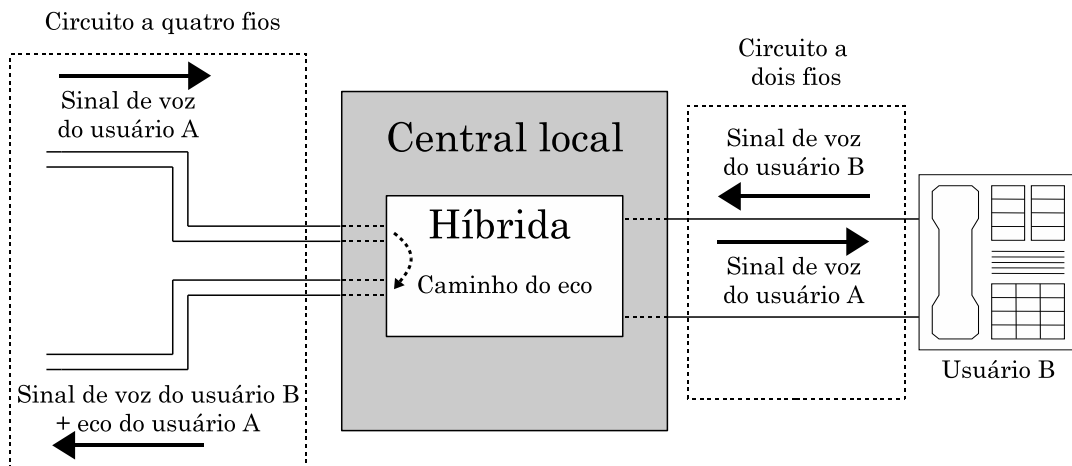


Figura 2.3: Sinal de eco em uma ligação telefônica.

Nas ligações locais (e mesmo em ligações interurbanas), a existência de eco não chega a causar desconforto ao usuário. Isso acontece porque o tempo que a voz leva para retornar é muito pequeno (geralmente, menor do que 20 ms) e, dessa maneira, o usuário percebe o eco apenas como uma sutil reverberação.

A grande motivação para o desenvolvimento de canceladores de eco se deve, entretanto, às situações nas quais existe um grande atraso no retorno da voz, como acontece, por exemplo, em ligações de longa distância via satélite. Nesses casos, a voz que retorna com atraso é percebida como um eco desconfortável que dificulta a conversação. Assim, em situações como essas, é desejável a utilização de um *cancelador de eco de linha* (*LEC, line echo canceller*), visando melhorar a qualidade da conversação e o conforto dos usuários [1].

Recentemente, devido à popularização das redes de voz sobre *IP* (VoIP), renovou-se o interesse na pesquisa e no desenvolvimento de novas estratégias de cancelamento de eco. Nessas redes, o tempo de retorno da voz é usualmente muito elevado (podendo chegar a mais de 300 ms) e a qualidade de conversação sem a utilização de um *LEC* torna-se muito pobre [14].

2.1.2 Eco Acústico

O *eco acústico* ocorre em sistemas em que existe acoplamento acústico entre alto-falante e microfone como, por exemplo, em telefones (analógicos ou digitais) equipados com função viva-voz e em equipamentos de teleconferência.

Ligações telefônicas através de viva-voz têm se tornado cada vez mais populares. Um exemplo de aplicação são os sistemas embarcados em automóveis que permitem ao usuário realizar ligações telefônicas enquanto mantém as mãos ao volante. Sistemas viva-voz e teleconferências são também muito utilizados no meio corporativo, pois permitem a realização de reuniões sem o deslocamento físico dos seus participantes.

Uma ilustração de caso em que ocorre o eco acústico é mostrada na Figura 2.4, que representa uma ligação telefônica entre o usuário A em um telefone convencional e o usuário B em um telefone viva-voz. Nesse exemplo, o sinal de voz do usuário A é reproduzido pelo alto-falante e captado pelo microfone, gerando eco. Além do caminho direto, ilustrado na figura, o som pode sofrer múltiplas reflexões antes de ser captado pelo microfone (principalmente, em ambientes fechados), gerando um efeito de reverberação. Assim, devido à baixa velocidade de propagação do som no ar (aproximadamente 344 m/s), o tempo de atraso no caminho do eco somado ao tempo de reverberação pode ser relativamente elevado. Por exemplo, em uma sala de escritório típica, esse tempo pode chegar a 300 ms [3].

Dessa forma, deve-se utilizar em sistemas viva-voz um *cancelador de eco acústico* (*AEC, acoustic echo canceller*) para reduzir o desconforto causado pelo eco aos usuários de tais sistemas.

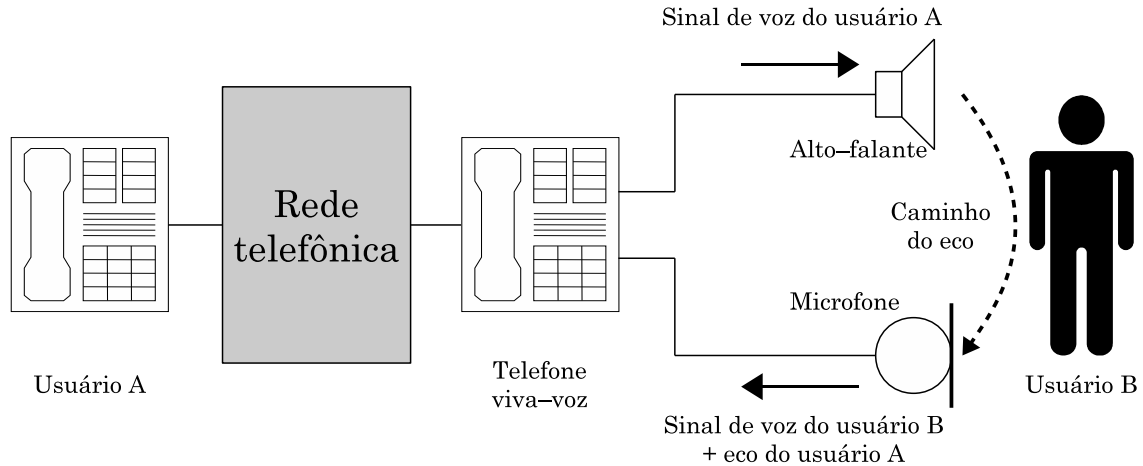


Figura 2.4: Eco acústico em telefone viva-voz.

2.2 Sistemas de Cancelamento de Eco

Uma primeira abordagem utilizada para controle do eco foi permitir apenas a comunicação *half-duplex*¹, através de utilização de supressores de eco [15]. O supressor de eco é um dispositivo que, ao detectar a presença de voz em um sentido do circuito, suprime (ou atenua fortemente) o sinal na outra direção. Essas interrupções temporárias na transmissão podem causar o efeito de “picotamento” nos sinais de voz dos usuários, prejudicando assim a clareza e a naturalidade da conversação.

Os canceladores de eco são desenvolvidos visando obter uma solução mais apropriada para o problema do eco [1]. Nos canceladores de eco, uma réplica sintética do sinal de eco é criada e subtraída do sinal de voz contaminado pelo eco. Essa subtração permite eliminar o eco sem interromper o caminho da voz dos usuários, preservando assim a qualidade da conversação.

Na Figura 2.5, um diagrama de blocos que representa a geração do eco em uma ligação telefônica é mostrado. Aqui, $x(n)$ representa o sinal de voz do usuário que escuta o eco (chamado na literatura usuário no *far-end*), enquanto $v(n)$ é o sinal de voz do usuário do outro lado da linha (chamado usuário no *near-end*) [16]. Geralmente, considera-se que o caminho percorrido pelo sinal de voz que retorna como eco pode ser descrito por um sistema linear. Na Figura 2.5, a resposta ao impulso desse sistema é denotada $h(n)$ e o sinal de eco, $d(n)$. O sinal recebido pelo usuário no *far-end* é a soma do eco $d(n)$ com a voz do usuário no *near-end* $v(n)$, o qual é representado na Figura 2.5 por $s(n)$.

Caso seja possível estimar a resposta ao impulso do sistema, pode-se obter uma

¹Um sistema *half-duplex*, apesar de ser bidirecional, não permite a comunicação nos dois sentidos simultaneamente. Ou seja, enquanto um lado opera como transmissor o outro necessariamente opera como receptor.

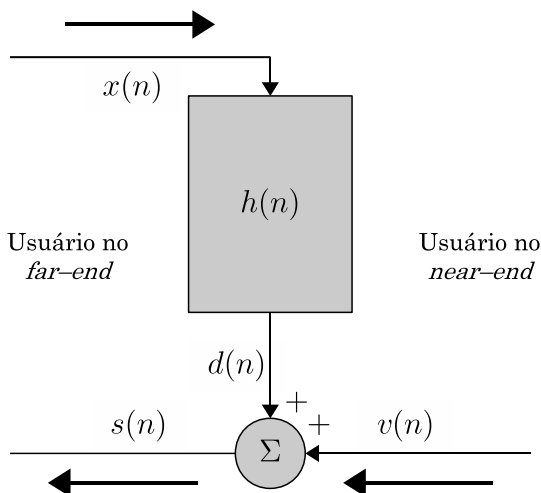


Figura 2.5: Diagrama de blocos da geração de eco em uma ligação telefônica.

réplica do sinal de eco. Para tal, essa estimativa [denotada por $\hat{h}(n)$] é utilizada como resposta ao impulso de um filtro. Excitando esse filtro pelo sinal $x(n)$, obtém-se uma estimativa do sinal de eco [denotada por $\hat{d}(n)$] que é subtraída do sinal $s(n)$. Tal procedimento é ilustrado na Figura 2.6.

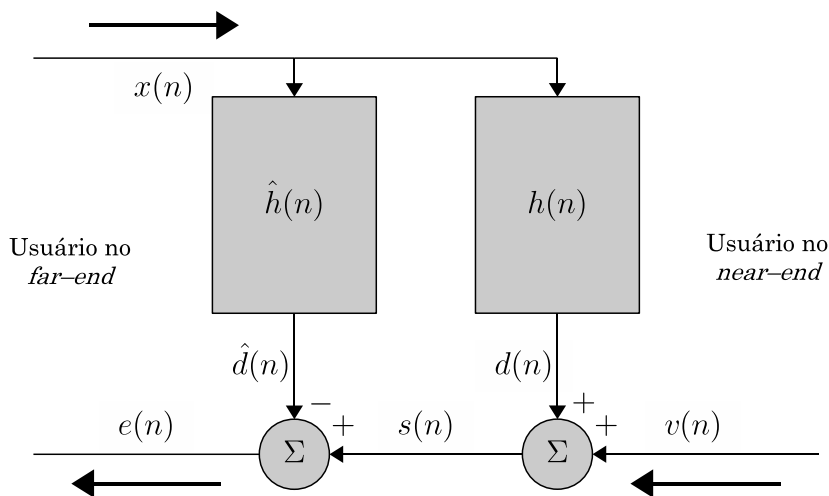


Figura 2.6: Diagrama de blocos simplificado de um sistema de cancelamento de eco.

Se $\hat{h}(n) \approx h(n)$, então $\hat{d}(n) \approx d(n)$ e o eco pode ser cancelado através de subtração. Nesses casos, o sinal residual $e(n)$ que é transmitido para o usuário no *far-end* é praticamente igual ao sinal de voz do usuário do *near-end* $v(n)$.

Nas estratégias de cancelamento de eco usualmente consideradas, a resposta ao impulso $h(n)$ é estimada em tempo real a partir dos sinais $x(n)$ e $d(n)$, utilizando um algoritmo de filtragem adaptativa [1]. O algoritmo adaptativo tem como objetivo alterar (ou adaptar) de forma iterativa $\hat{h}(n)$ visando gradualmente reduzir a diferença entre o sinal de eco e a sua estimativa. Dessa forma, o problema do cancelamento de eco pode ser

interpretado como um problema de identificação de sistemas [3].

Existem disponíveis na literatura diversos algoritmos de filtragem adaptativa, com diferentes características de desempenho, estabilidade numérica e complexidade computacional. Alguns dos algoritmos mais utilizados em aplicações de cancelamento de eco são discutidos no Capítulo 3.

2.2.1 Detecção de *Double-Talk*

Em um sistema de cancelamento de eco, não se tem acesso diretamente ao sinal de eco $d(n)$, que é utilizado na adaptação dos coeficientes do filtro. O sinal disponível é $s(n)$, que contém, além do sinal de eco $d(n)$, o sinal de voz do usuário no *near-end* $v(n)$.

As situações em que existe atividade de voz dos dois usuários [ou seja, $x(n) \neq 0$ e $v(n) \neq 0$] são chamadas *double-talk*. Nessas situações, a voz do usuário no *near-end* atua como uma perturbação na adaptação dos coeficientes do filtro, prejudicando assim a eficácia do cancelamento de eco [17]. A maneira mais usual de contornar tal problema é reduzir ou parar completamente a adaptação quando é detectada atividade do usuário no *near-end*. Para tal, é utilizado um detector de *double-talk* (*DTD*, *double-talk detector*).

O desempenho de um cancelador de eco é impactado caso existam falhas na detecção de *double-talk*. Um tipo comum de falha é o falso alarme, que ocorre quando o detector acusa a presença de *double-talk* em um instante em que não há atividade do usuário do *near-end* [$v(n) = 0$]. Outro tipo de falha, denominada perda, ocorre quando *double-talk* não é identificado em um instante em que há atividade dos dois usuários. Geralmente, existe uma relação de compromisso entre baixa probabilidade de falsos alarmes e de perdas.

2.3 Considerações

Neste capítulo, foram apresentados os tipos de eco mais comuns em ligações telefônicas: o eco de linha e o eco acústico. Em determinadas situações, a presença de eco pode trazer desconforto aos usuários e prejudicar a qualidade da conversação.

Uma estrutura básica de cancelamento de eco é discutida, na qual um filtro adaptativo é utilizado para gerar uma réplica do sinal de eco. Dessa forma, o cancelamento é obtido através de subtração. Para um funcionamento adequado do cancelador de eco, deve existir uma detecção eficaz das situações nas quais existe atividade do usuário no *near-end*. Para tal, são empregados algoritmos de detecção de *double-talk*. Falhas na detecção de *double-talk* podem impactar a adaptação dos coeficientes do filtro, prejudicando assim o processo de cancelamento do eco.

3 Algoritmos de Filtragem Adaptativa

Neste capítulo, é apresentada uma introdução à filtragem adaptativa. Algumas das abordagens geralmente consideradas no desenvolvimento dos algoritmos adaptativos são discutidas. Os algoritmos *least-mean-square* (LMS) e *recursive least-squares* (RLS) (e algumas das suas variações), muito usados na prática, são também analisados.

Um sistema adaptativo é um sistema cuja estrutura é alterável ou ajustável tal que o seu desempenho melhore através do contato com o ambiente ao seu redor [18]. Dessa forma, a propriedade essencial dos sistemas adaptativos é que o seu desempenho seja variante e auto-ajustável. Essa propriedade é desejável nos casos em que não é possível prever, *a priori*, as condições nas quais um determinado sistema opera e as características dos sinais de entrada desse sistema.

Os filtros adaptativos representam uma classe particular dos sistemas adaptativos. Em geral, a resposta ao impulso de um filtro adaptativo é alterada de forma iterativa, visando aproximar a saída desse filtro de um determinado sinal desejado. Na Figura 3.1, é mostrado um diagrama de blocos de uma aplicação típica de filtragem adaptativa. O sinal de excitação é representado na figura por $x(n)$ e a resposta ao impulso do filtro adaptativo no instante n , por $\hat{h}_k(n)$. Tal resposta é adaptada de modo que a saída do filtro, denotada por $\hat{d}(n)$, aproxime-se do sinal desejado [representado na figura por $d(n)$]. Dessa forma, o sinal de erro $e(n)$ é gradativamente reduzido.

Diversos algoritmos de filtragem adaptativa são apresentados na literatura. Algumas das características mais importantes desses algoritmos são [3]:

Velocidade de convergência. Definida como o número de iterações necessárias para que o filtro adaptativo opere em regime permanente.

Desajuste. Medida quantitativa do erro em regime permanente, quando comparado com uma solução ótima.

Rastreamento. Medida da capacidade do algoritmo de “rastrear” variações estatísticas

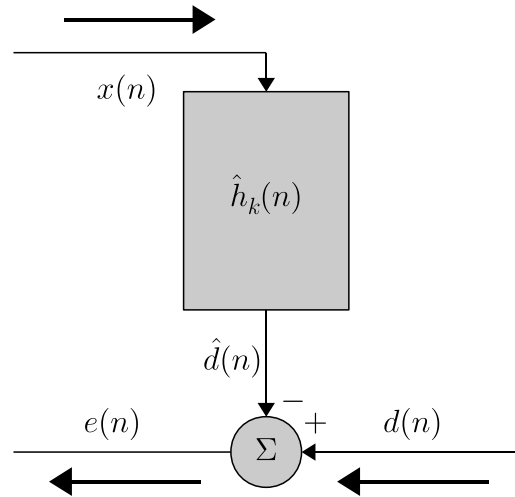


Figura 3.1: Diagrama em blocos do filtro adaptativo.

em ambientes não-estacionários.

Robustez. Um filtro é considerado robusto quando pequenos distúrbios (ou seja, distúrbios com pequena energia) resultam em pequenos erros de estimação.

Complexidade computacional. Reflexo da quantidade de operações aritméticas (adições, multiplicações e divisões) e da quantidade de memória requeridas pelo algoritmo.

Robustez numérica. Sensibilidade do algoritmo a erros causados pela implementação, considerando aritmética de precisão finita e quantização dos sinais de entrada.

Na escolha de um algoritmo de filtragem adaptativa, deve ser considerada a estrutura do filtro: de resposta ao impulso infinita (*IIR*, *infinite impulse response*) ou finita (*FIR*, *finite impulse response*) [19]. Essa escolha é ditada principalmente por critérios práticos [3]. Os filtros FIR são inerentemente estáveis, enquanto os filtros IIR podem ser instáveis para determinados conjuntos de coeficientes. Ainda que seja possível projetar filtros IIR estáveis, garantir sua estabilidade em um processo adaptativo não é uma tarefa trivial. É por esse motivo que a maioria das implementações práticas de filtros adaptativos utiliza estruturas FIR, ainda que estruturas IIR exijam, para uma dada ordem, menor complexidade computacional. Neste trabalho, apenas filtros adaptativos FIR são considerados.

Existem diversas possibilidades para estabelecer o critério de desempenho segundo o qual o algoritmo de filtragem adaptativa atualiza (ou adapta) seus coeficientes. Em geral, costuma-se adotar a minimização de uma função custo que dependa do sinal de erro como

critério de desempenho [20]. Na escolha dessa função, devem ser levados em conta fatores como facilidade de tratamento matemático e complexidade computacional.

A seguir, são apresentadas duas abordagens utilizadas no desenvolvimento de algoritmos adaptativos. Na primeira, a função custo considerada é o valor médio do erro quadrático. Essa abordagem dá origem ao filtro de Wiener, do qual deriva o algoritmo *least-mean-square* (LMS). Na segunda, a função custo considerada é uma soma ponderada de amostras passadas do erro quadrático. Dessa abordagem, obtém-se o algoritmo *recursive least-squares* (RLS).

3.1 Abordagem do Erro Quadrático Médio

A abordagem do erro quadrático médio (MSE, *mean-square error*) tem como grande vantagem a facilidade no tratamento matemático [3], [18]. A função custo que se deseja minimizar é o valor médio do erro quadrático. Assim,

$$J_{\text{MSE}} = \text{E} [e^2(n)] \quad (3.1)$$

onde $\text{E}(\cdot)$ denota o operador valor esperado. É possível determinar um filtro linear ótimo (não adaptativo) que corresponda ao mínimo de (3.1), denominado filtro de Wiener. A partir do filtro de Wiener, são derivados alguns algoritmos de filtragem adaptativa, tal como o LMS.

3.1.1 Filtro de Wiener

O filtro de Wiener é o filtro linear que corresponde ao conjunto de coeficientes que oferece o menor valor possível para o erro quadrático médio J_{MSE} . Na derivação do filtro de Wiener, considera-se inicialmente que o filtro ilustrado na Figura 3.1 é invariante e que os sinais envolvidos são estacionários e têm média zero. Assim, a saída $\hat{d}(n)$ do filtro é obtida como

$$\hat{d}(n) = \sum_{k=0}^{N-1} \hat{h}_k x(n-k) = \hat{\mathbf{h}}^T \mathbf{x}(n) \quad (3.2)$$

onde $\mathbf{x}(n) = [x(n) \ x(n-1) \ \cdots \ x(n-N+1)]^T$ representa o vetor de excitação e $\hat{\mathbf{h}} = [\hat{h}_0 \ \hat{h}_1 \ \cdots \ \hat{h}_{N-1}]^T$, o vetor de coeficientes. Dessa forma, o sinal de erro pode ser escrito como

$$e(n) = d(n) - \hat{d}(n) = d(n) - \hat{\mathbf{h}}^T \mathbf{x}(n) \quad (3.3)$$

e (3.1) pode ser expressa como função de $\hat{\mathbf{h}}$. Assim,

$$\begin{aligned} J_{\text{MSE}}(\hat{\mathbf{h}}) &= \text{E} \left\{ \left[d(n) - \hat{\mathbf{h}}^T \mathbf{x}(n) \right]^2 \right\} \\ &= \text{E} [d^2(n)] - 2 \text{E} [d(n) \hat{\mathbf{h}}^T \mathbf{x}(n)] + \text{E} [\hat{\mathbf{h}}^T \mathbf{x}(n) \mathbf{x}^T(n) \hat{\mathbf{h}}] \\ &= \text{E} [d^2(n)] - 2 \hat{\mathbf{h}}^T \text{E} [\mathbf{x}(n) d(n)] + \hat{\mathbf{h}}^T \text{E} [\mathbf{x}(n) \mathbf{x}^T(n)] \hat{\mathbf{h}}. \end{aligned} \quad (3.4)$$

Agora, denotando por σ_d^2 a variância de $d(n)$

$$\sigma_d^2 = \text{E} [d^2(n)], \quad (3.5)$$

por $\mathbf{r}_{\mathbf{x}d}$ o vetor $N \times 1$ de correlação cruzada entre $\mathbf{x}(n)$ e $d(n)$

$$\mathbf{r}_{\mathbf{x}d} = \text{E} [\mathbf{x}(n) d(n)] \quad (3.6)$$

e por $\mathbf{R}_{\mathbf{x}}$ a matriz $N \times N$ de autocorrelação de $\mathbf{x}(n)$

$$\mathbf{R}_{\mathbf{x}} = \text{E} [\mathbf{x}(n) \mathbf{x}^T(n)], \quad (3.7)$$

reescreve-se (3.4) como

$$J_{\text{MSE}}(\hat{\mathbf{h}}) = \sigma_d^2 - 2 \hat{\mathbf{h}}^T \mathbf{r}_{\mathbf{x}d} + \hat{\mathbf{h}}^T \mathbf{R}_{\mathbf{x}} \hat{\mathbf{h}}. \quad (3.8)$$

Observa-se que $J_{\text{MSE}}(\hat{\mathbf{h}})$ depende quadraticamente dos coeficientes de $\hat{\mathbf{h}}$ e, portanto, apresenta um único ponto de mínimo, denotado por $\hat{\mathbf{h}}_{\text{MSE}}$. Para determinar esse ponto, define-se o operador gradiente ∇ , cujo k -ésimo componente é dada por

$$\nabla_k = \frac{\partial}{\partial \hat{h}_k}, \quad k = 0, 1, \dots, N-1.$$

Aplicando o operador ∇ em (3.8), obtém-se

$$\begin{aligned} \nabla J_{\text{MSE}}(\hat{\mathbf{h}}) &= \begin{bmatrix} \frac{\partial}{\partial \hat{h}_0} J_{\text{MSE}}(\hat{\mathbf{h}}) \\ \frac{\partial}{\partial \hat{h}_1} J_{\text{MSE}}(\hat{\mathbf{h}}) \\ \vdots \\ \frac{\partial}{\partial \hat{h}_{N-1}} J_{\text{MSE}}(\hat{\mathbf{h}}) \end{bmatrix} \\ &= -2 \mathbf{r}_{\mathbf{x}d} + 2 \mathbf{R}_{\mathbf{x}} \hat{\mathbf{h}}. \end{aligned} \quad (3.9)$$

No ponto em que $J_{\text{MSE}}(\hat{\mathbf{h}})$ assume o seu valor mínimo, os componentes do vetor gradiente se anulam. Assim, denotando por $\hat{\mathbf{h}}_{\text{MSE}}$ o vetor de coeficientes que corresponde ao mínimo

de $J_{\text{MSE}}(\hat{\mathbf{h}})$, verifica-se que

$$\nabla J_{\text{MSE}}(\hat{\mathbf{h}}_{\text{MSE}}) = -2\mathbf{r}_{\text{xd}} + 2\mathbf{R}_{\mathbf{x}}\hat{\mathbf{h}}_{\text{MSE}} = \mathbf{0}_N \quad (3.10)$$

onde $\mathbf{0}_N$ denota um vetor de zeros com dimensão N . Reescrevendo (3.10) como

$$\mathbf{R}_{\mathbf{x}}\hat{\mathbf{h}}_{\text{MSE}} = \mathbf{r}_{\text{xd}} \quad (3.11)$$

obtém-se a equação de Wiener-Hopf. Assumindo que $\mathbf{R}_{\mathbf{x}}$ seja não-singular, $\hat{\mathbf{h}}_{\text{MSE}}$ pode ser obtido por

$$\hat{\mathbf{h}}_{\text{MSE}} = \mathbf{R}_{\mathbf{x}}^{-1}\mathbf{r}_{\text{xd}}. \quad (3.12)$$

O vetor $\hat{\mathbf{h}}_{\text{MSE}}$ é o vetor de coeficientes que apresenta o mínimo erro quadrático médio. Na literatura, esse conjunto de coeficientes é usualmente denominado filtro de Wiener. Observa-se que a determinação do filtro de Wiener depende do conhecimento da matriz de autocorrelação do sinal de entrada $\mathbf{x}(n)$ e do vetor de correlação cruzada entre $\mathbf{x}(n)$ e $d(n)$.

3.1.2 Cálculo do Filtro de Wiener Através do Método *Steepest Descent*

É possível ainda calcular $\hat{\mathbf{h}}_{\text{MSE}}$ sem a aplicação direta de (3.12), através de um procedimento iterativo de otimização. Nesse processo, parte-se de um valor inicial arbitrário $\hat{\mathbf{h}}^{(0)}$. Na j -ésima iteração, o vetor de coeficientes assume um valor particular $\hat{\mathbf{h}}^{(j)} = [\hat{h}_0^{(j)} \ \hat{h}_1^{(j)} \ \dots \ \hat{h}_{N-1}^{(j)}]^T$, que é utilizado para calcular $\hat{\mathbf{h}}^{(j+1)}$. Com o aumento do número de iterações, espera-se que $\hat{\mathbf{h}}^{(j)}$ aproxime-se de $\hat{\mathbf{h}}_{\text{MSE}}$. O procedimento iterativo aqui considerado é método *steepest descent* (SD), que é uma técnica bem conhecida de otimização.

Na iteração j , a função custo pode ser escrita como

$$J_{\text{MSE}}^{(j)} = \sigma_d^2 - 2 \left[\hat{\mathbf{h}}^{(j)} \right]^T \mathbf{r}_{\text{xd}} + \left[\hat{\mathbf{h}}^{(j)} \right]^T \mathbf{R}_{\mathbf{x}} \hat{\mathbf{h}}^{(j)} \quad (3.13)$$

e o seu gradiente,

$$\nabla J_{\text{MSE}}^{(j)} = -2\mathbf{r}_{\text{xd}} + 2\mathbf{R}_{\mathbf{x}}\hat{\mathbf{h}}^{(j)}. \quad (3.14)$$

De acordo com o método SD, o valor do vetor de coeficientes na iteração $j + 1$ é obtido incrementando o valor atual $\hat{\mathbf{h}}^{(j)}$ na direção oposta àquela apontada pelo vetor gradiente.

Assim,

$$\hat{\mathbf{h}}^{(j+1)} = \hat{\mathbf{h}}^{(j)} + \frac{1}{2}\mu[-\nabla J_{\text{MSE}}^{(j)}] \quad (3.15)$$

onde μ é uma constante positiva e o fator $\frac{1}{2}$ é incluído por simplicidade matemática. Assim,

substituindo (3.14) em (3.15), obtém-se a equação recursiva do método SD. Portanto,

$$\hat{\mathbf{h}}^{(j+1)} = \hat{\mathbf{h}}^{(j)} + \mu \left[\mathbf{r}_{xd} - \mathbf{R}_x \hat{\mathbf{h}}^{(j)} \right]. \quad (3.16)$$

Dado que o valor de μ seja adequadamente escolhido [3], a solução obtida pelo método SD converge para o filtro de Wiener. Dessa forma,

$$\lim_{j \rightarrow \infty} \hat{\mathbf{h}}^{(j)} = \hat{\mathbf{h}}_{\text{MSE}}. \quad (3.17)$$

3.1.3 Algoritmo *Least-Mean-Square* (LMS)

Nos casos em que as estatísticas dos sinais $x(n)$ e $s(n)$ são conhecidas, o filtro que minimiza o erro quadrático médio pode ser calculado diretamente através da equação de Wiener-Hopf ou iterativamente utilizando o método SD. No entanto, na prática nem sempre é possível conhecer tais estatísticas *a priori*. Mais ainda, em muitos casos os sinais envolvidos não são estacionários, fazendo com que as estatísticas variem com o tempo.

Assim, é desejável obter um algoritmo que dependa apenas dos sinais $x(n)$ e $s(n)$, e que tenha a capacidade de se ajustar a variações estatísticas desses sinais. Para tal, são definidas as estimativas instantâneas da matriz de autocorrelação e do vetor de correlação cruzada. Assim,

$$\hat{\mathbf{R}}_x(n) = \mathbf{x}(n)\mathbf{x}^T(n) \quad (3.18)$$

e

$$\hat{\mathbf{r}}_{xd}(n) = \mathbf{x}(n)d(n). \quad (3.19)$$

Dessa forma, utilizando (3.18) e (3.19), uma versão simplificada do método SD pode ser obtida. Considerando que é realizada uma iteração do método para cada nova amostra de $x(n)$ e $d(n)$, o vetor de coeficientes nesse caso pode ser escrito como $\hat{\mathbf{h}}(n) = [\hat{h}_0(n) \ \hat{h}_1(n) \ \cdots \ \hat{h}_{N-1}(n)]^T$. Assim, a estimativa no vetor gradiente para o instante n é

$$\begin{aligned} \hat{\nabla} J_{\text{MSE}}(n) &= -2\mathbf{x}(n)d(n) + 2\mathbf{x}(n)\mathbf{x}^T(n)\hat{\mathbf{h}}(n) \\ &= -2\mathbf{x}(n) \left[d(n) - \hat{d}(n) \right] \\ &= -2\mathbf{x}(n)e(n) \end{aligned} \quad (3.20)$$

com

$$e(n) = d(n) - \hat{d}(n) = d(n) - \mathbf{x}^T(n)\hat{\mathbf{h}}(n). \quad (3.21)$$

Substituindo $\nabla J_{\text{MSE}}(n)$ por $\hat{\nabla} J_{\text{MSE}}(n)$ em (3.15), obtém-se a seguinte equação recursiva:

$$\hat{\mathbf{h}}(n+1) = \hat{\mathbf{h}}(n) + \mu \mathbf{x}(n)e(n).$$

O algoritmo obtido, definido por (3.21) e (3.1.3), é denominado *least-mean-square* (LMS). Observa-se que $\hat{\mathbf{h}}(n)$ caracteriza um filtro adaptativo, pois os seus coeficientes são modificados de forma iterativa a partir dos dados de entrada. Dessa forma, não é necessária qualquer informação *a priori* sobre as características dos sinais envolvidos.

Em aplicações práticas, os valores dos coeficientes do filtro adaptativo são geralmente inicializados com zero. Assim,

$$\hat{\mathbf{h}}(0) = \mathbf{0}_N . \quad (3.22)$$

Na Tabela 3.1, são mostradas as equações que caracterizam o algoritmo LMS. Geralmente, a complexidade computacional de um algoritmo é medida através do número de multiplicações requeridas pelo algoritmo. No caso do LMS, conforme observado na Tabela 3.1, são necessárias $2N$ multiplicações por iteração.

Tabela 3.1: Algoritmo LMS
Inicialização do algoritmo
$\hat{\mathbf{h}}(0) = \mathbf{0}_N$
Atualização dos coeficientes
$e(n) = d(n) - \mathbf{x}^T(n)\hat{\mathbf{h}}(n)$
$\hat{\mathbf{h}}(n+1) = \hat{\mathbf{h}}(n) + \mu \mathbf{x}(n)e(n)$

O algoritmo LMS é muito utilizado em aplicações práticas de filtragem adaptativa. Isso se deve principalmente à sua robustez, simplicidade e baixa complexidade computacional. No entanto, em determinadas situações, a velocidade de convergência do algoritmo LMS é baixa [3]. Isso acontece principalmente quando as amostras do sinal de entrada são fortemente correlacionadas, como, por exemplo, para sinais de voz.

3.2 Abordagem dos Mínimos Quadrados

Uma outra abordagem usualmente considerada no desenvolvimento de algoritmos adaptativos é a dos mínimos quadrados (LS). Esse critério dá origem ao algoritmo RLS, que é também utilizado em diversas aplicações práticas.

Como no caso do algoritmo LMS, considera-se aqui que o vetor de coeficientes assume para cada n um valor particular $\hat{\mathbf{h}}(n) = [\hat{h}_0(n) \ \hat{h}_1(n) \ \cdots \ \hat{h}_{N-1}(n)]^T$. Dado que são conhecidos os valores de $x(1), x(2), \dots, x(n)$ e $d(1), d(2), \dots, d(n)$, pode-se definir um erro de estimação considerando o valor atual de $\hat{\mathbf{h}}(n)$ e amostras passadas dos sinais de entrada

$x(i)$ e $d(i)$, com $i = 1, 2, \dots, n$. Esse erro de estimação pode ser escrito como

$$\begin{aligned} e_i(n) &= d(i) - \sum_{k=0}^{N-1} \hat{h}_k(n) x(i-k) \\ &= d(i) - \hat{\mathbf{h}}^T(n) \mathbf{x}(i). \end{aligned} \quad (3.23)$$

Dessa forma, o valor de $e_i(n)$ depende da amostra passada $d(i)$, do vetor de amostras passadas $\mathbf{x}(i) = [x(i) \ x(i-1) \ \dots \ x(i-N+1)]^T$ e do vetor atual de coeficientes $\hat{\mathbf{h}}(n)$. Geralmente, considera-se que o sinal de entrada é pré-janelado [ou seja, $x(n) = 0$ para $n \leq 0$]. Na abordagem dos mínimos quadrados, a função custo considerada é a soma exponencialmente ponderada dos erros de estimação ao quadrado:

$$J_{\text{LS}}(n) = \sum_{i=1}^n \lambda^{n-i} e_i^2(n). \quad (3.24)$$

onde $0 < \lambda \leq 1$ é denominado fator de esquecimento. O caso particular em que $\lambda = 1$ corresponde à memória infinita. Já para $\lambda < 1$, observa-se que os valores mais recentes dos sinais de entrada têm peso maior do que valores passados. De maneira geral, o inverso de $1 - \lambda$ fornece uma medida da memória do método [3]. Substituindo (3.23) em (3.24), tem-se

$$\begin{aligned} J_{\text{LS}}(n) &= \sum_{i=1}^n \lambda^{n-i} \left[d(i) - \hat{\mathbf{h}}^T(n) \mathbf{x}(i) \right]^2 \\ &= \sum_{i=1}^n \lambda^{n-i} \left[d^2(i) - 2d(i) \hat{\mathbf{h}}^T(n) \mathbf{x}(i) + \hat{\mathbf{h}}(n)^T \mathbf{x}(i) \mathbf{x}^T(i) \hat{\mathbf{h}}(n) \right]. \end{aligned} \quad (3.25)$$

Como no caso do filtro de Wiener, deseja-se determinar o vetor de coeficientes que anule o gradiente da função custo. O gradiente de (3.25) é

$$\nabla J_{\text{LS}}(n) = 2 \sum_{i=1}^n \lambda^{n-i} \left[-d(i) \mathbf{x}(i) + \mathbf{x}(i) \mathbf{x}^T(i) \hat{\mathbf{h}}(n) \right]. \quad (3.26)$$

Considerando que todos os componentes do vetor gradiente se anulam, verifica-se que

$$\sum_{i=1}^n \lambda^{n-i} \left[-d(i) \mathbf{x}(i) + \mathbf{x}(i) \mathbf{x}^T(i) \hat{\mathbf{h}}_{\text{LS}}(n) \right] = 0, \quad (3.27)$$

ou seja,

$$\sum_{i=1}^n \left[\lambda^{n-i} \mathbf{x}(i) \mathbf{x}^T(i) \right] \hat{\mathbf{h}}_{\text{LS}}(n) = \sum_{i=1}^n \lambda^{n-i} \mathbf{x}(i) d(i). \quad (3.28)$$

onde $\hat{\mathbf{h}}_{\text{LS}}(n)$ é o vetor de coeficientes que corresponde à solução de mínimos quadrados.

Agora, definindo a matriz $N \times N$ de autocorrelação amostral do sinal de entrada

$$\begin{aligned}\tilde{\mathbf{R}}_{\mathbf{x}}(n) &= \sum_{i=1}^n \lambda^{n-i} \mathbf{x}(i) \mathbf{x}^T(i) \\ &= \mathbf{x}(n) \mathbf{x}^T(n) + \lambda \tilde{\mathbf{R}}_{\mathbf{x}}(n-1)\end{aligned}\quad (3.29)$$

e o vetor $N \times 1$ de correlação cruzada amostral entre o sinal de entrada e o sinal desejado como

$$\begin{aligned}\tilde{\mathbf{r}}_{\mathbf{x}d}(n) &= \sum_{i=1}^n \lambda^{n-i} \mathbf{x}(i) d(i) \\ &= \mathbf{x}(n) d(n) + \lambda \tilde{\mathbf{r}}_{\mathbf{x}d}(n-1)\end{aligned}\quad (3.30)$$

reescreve-se (3.28) como

$$\tilde{\mathbf{R}}_{\mathbf{x}}(n) \hat{\mathbf{h}}_{\text{LS}}(n) = \tilde{\mathbf{r}}_{\mathbf{x}d}(n). \quad (3.31)$$

A expressão (3.31) é equivalente à equação de Wiener-Hopf, para o caso dos mínimos quadrados. Admitindo que $\tilde{\mathbf{R}}_{\mathbf{x}}(n)$ seja não-singular, tem-se

$$\hat{\mathbf{h}}_{\text{LS}}(n) = \tilde{\mathbf{R}}_{\mathbf{x}}^{-1}(n) \tilde{\mathbf{r}}_{\mathbf{x}d}(n). \quad (3.32)$$

Dessa forma, pode-se usar (3.32) para obter o vetor de coeficientes que corresponde, para todo n , ao mínimo de (3.24).

3.2.1 Algoritmo *Recursive Least-Squares* (RLS)

Uma desvantagem da utilização de (3.32) para determinar $\hat{\mathbf{h}}_{\text{LS}}(n)$ é a necessidade de calcular a inversa de $\tilde{\mathbf{R}}_{\mathbf{x}}(n)$ para cada nova amostra de $x(n)$ e $d(n)$. Na prática, essa operação pode demandar elevada carga computacional, principalmente, se a ordem N for elevada. É possível, no entanto, obter um procedimento para determinar recursivamente $\tilde{\mathbf{R}}_{\mathbf{x}}^{-1}(n)$. Para tal, lança-se mão do lema da inversão de matrizes. Sejam duas matrizes definidas positivas $N \times N$ \mathbf{A} e \mathbf{B} relacionadas por

$$\mathbf{A} = \mathbf{B}^{-1} + \mathbf{C} \mathbf{D}^{-1} \mathbf{C}^T \quad (3.33)$$

onde \mathbf{C} é uma matriz $N \times M$ e \mathbf{D} é uma matriz $M \times M$. De acordo com o lema da inversão de matrizes, o inverso de \mathbf{A} pode ser escrito como

$$\mathbf{A}^{-1} = \mathbf{B} - \mathbf{B} \mathbf{C} (\mathbf{D} + \mathbf{C}^T \mathbf{B} \mathbf{C})^{-1} \mathbf{C}^T \mathbf{B}. \quad (3.34)$$

Agora, igualando (3.29) a (3.33), verifica-se que

$$\begin{aligned}\mathbf{A} &= \tilde{\mathbf{R}}_{\mathbf{x}}(n) \\ \mathbf{B}^{-1} &= \lambda \tilde{\mathbf{R}}_{\mathbf{x}}(n-1) \\ \mathbf{C} &= \mathbf{x}(n) \\ \mathbf{D} &= 1\end{aligned}$$

Dessa forma, utilizando (3.34), pode-se escrever $\tilde{\mathbf{R}}_{\mathbf{x}}(n)^{-1}$ como

$$\tilde{\mathbf{R}}_{\mathbf{x}}^{-1}(n) = \frac{1}{\lambda} \tilde{\mathbf{R}}_{\mathbf{x}}^{-1}(n-1) - \frac{1}{\lambda^2} \left[\frac{\tilde{\mathbf{R}}_{\mathbf{x}}^{-1}(n-1) \mathbf{x}(n) \mathbf{x}^T(n) \tilde{\mathbf{R}}_{\mathbf{x}}^{-1}(n-1)}{1 + \frac{1}{\lambda} \mathbf{x}^T(n) \tilde{\mathbf{R}}_{\mathbf{x}}^{-1}(n-1) \mathbf{x}(n)} \right] \quad (3.35)$$

Por simplicidade matemática, define-se o vetor $N \times 1$

$$\mathbf{k}(n) = \frac{\tilde{\mathbf{R}}_{\mathbf{x}}^{-1}(n-1) \mathbf{x}(n)}{\lambda + \mathbf{x}^T(n) \tilde{\mathbf{R}}_{\mathbf{x}}^{-1}(n-1) \mathbf{x}(n)} \quad (3.36)$$

e (3.35) pode agora ser reescrita como

$$\tilde{\mathbf{R}}_{\mathbf{x}}^{-1}(n) = \frac{1}{\lambda} [\mathbf{I} - \mathbf{k}(n) \mathbf{x}^T(n)] \tilde{\mathbf{R}}_{\mathbf{x}}^{-1}(n-1). \quad (3.37)$$

O vetor $\mathbf{k}(n)$ é geralmente denominado na literatura ganho de Kalman [3]. Observa-se que (3.36) e (3.37) permitem o cálculo recursivo de $\tilde{\mathbf{R}}_{\mathbf{x}}^{-1}(n)$ sem a necessidade da inversão explícita de $\tilde{\mathbf{R}}_{\mathbf{x}}(n)$.

Utilizando o ganho de Kalman, é possível também obter uma expressão recursiva para o vetor de coeficientes $\hat{\mathbf{h}}_{\text{LS}}(n)$. Para tal, reescreve-se (3.36) como

$$\begin{aligned}\mathbf{k}(n) &= \frac{1}{\lambda} \left[\tilde{\mathbf{R}}_{\mathbf{x}}^{-1}(n-1) \mathbf{x}(n) - \mathbf{k}(n) \mathbf{x}^T(n) \tilde{\mathbf{R}}_{\mathbf{x}}^{-1}(n-1) \mathbf{x}(n) \right] \\ &= \left\{ \frac{1}{\lambda} [\mathbf{I} - \mathbf{k}(n) \mathbf{x}^T(n)] \tilde{\mathbf{R}}_{\mathbf{x}}^{-1}(n-1) \right\} \mathbf{x}(n).\end{aligned} \quad (3.38)$$

Substituindo a expressão entre chaves em (3.38) por (3.37), obtém-se

$$\mathbf{k}(n) = \tilde{\mathbf{R}}_{\mathbf{x}}^{-1}(n) \mathbf{x}(n). \quad (3.39)$$

Assim, aplicando (3.30) e (3.39) em (3.32), obtém-se a seguinte expressão recursiva para

$\hat{\mathbf{h}}_{\text{LS}}(n)$:

$$\begin{aligned}
\hat{\mathbf{h}}_{\text{LS}}(n) &= \tilde{\mathbf{R}}_{\mathbf{x}}^{-1}(n) [\lambda \tilde{\mathbf{r}}_{\text{xd}}(n-1) + \mathbf{x}(n)d(n)] \\
&= \lambda \tilde{\mathbf{R}}_{\mathbf{x}}^{-1}(n) \tilde{\mathbf{r}}_{\text{xd}}(n-1) + \tilde{\mathbf{R}}_{\mathbf{x}}^{-1}(n) \mathbf{x}(n)d(n) \\
&= [\mathbf{I} - \mathbf{k}(n)\mathbf{x}^{\text{T}}(n)] \tilde{\mathbf{R}}_{\mathbf{x}}^{-1}(n-1) \tilde{\mathbf{r}}_{\text{xd}}(n-1) + \mathbf{k}(n)d(n) \\
&= [\mathbf{I} - \mathbf{k}(n)\mathbf{x}^{\text{T}}(n)] \hat{\mathbf{h}}_{\text{LS}}(n-1) + \mathbf{k}(n)d(n) \\
&= \hat{\mathbf{h}}_{\text{LS}}(n-1) + \mathbf{k}(n) [d(n) - \mathbf{x}^{\text{T}}(n)\hat{\mathbf{h}}_{\text{LS}}(n-1)] \\
&= \hat{\mathbf{h}}_{\text{LS}}(n-1) + \mathbf{k}(n)\epsilon(n)
\end{aligned} \tag{3.40}$$

onde $\epsilon(n)$ é o erro *a priori*, definido como

$$\epsilon(n) = d(n) - \mathbf{x}^{\text{T}}(n)\hat{\mathbf{h}}_{\text{LS}}(n-1). \tag{3.41}$$

As equações (3.36), (3.41), (3.40) e (3.37) definem o algoritmo *recursive least-squares* (RLS). Para inicializar o algoritmo, geralmente considera-se [20]

$$\tilde{\mathbf{R}}_{\mathbf{x}}^{-1}(0) = \delta^{-1}\mathbf{I}_N \tag{3.42}$$

onde \mathbf{I}_N denota a matriz identidade de ordem N e δ é uma constante positiva. Já o vetor de coeficientes, assim como no algoritmo LMS, é geralmente inicializado como um vetor de zeros. Na Tabela 3.2, são mostradas as equações que caracterizam o algoritmo RLS.

Tabela 3.2: Algoritmo RLS	
Inicialização do algoritmo	
$\hat{\mathbf{h}}_{\text{LS}}(0) = \mathbf{0}_N$	
$\tilde{\mathbf{R}}_{\mathbf{x}}^{-1}(0) = \delta^{-1}\mathbf{I}_N$	
Atualização de $\mathbf{k}(n)$ e $\tilde{\mathbf{R}}_{\mathbf{x}}^{-1}(n)$	
$\mathbf{k}(n) = \frac{\tilde{\mathbf{R}}_{\mathbf{x}}^{-1}(n-1)\mathbf{x}(n)}{\lambda + \mathbf{x}^{\text{T}}(n)\tilde{\mathbf{R}}_{\mathbf{x}}^{-1}(n-1)\mathbf{x}(n)}$	
$\tilde{\mathbf{R}}_{\mathbf{x}}^{-1}(n) = \frac{1}{\lambda} [\mathbf{I} - \mathbf{k}(n)\mathbf{x}^{\text{T}}(n)] \tilde{\mathbf{R}}_{\mathbf{x}}^{-1}(n-1)$	
Adaptação do vetor de coeficientes	
$\epsilon(n) = d(n) - \mathbf{x}^{\text{T}}(n)\hat{\mathbf{h}}_{\text{LS}}(n-1)$	
$\hat{\mathbf{h}}_{\text{LS}}(n) = \hat{\mathbf{h}}_{\text{LS}}(n-1) + \mathbf{k}(n)\epsilon(n)$	

A principal vantagem do algoritmo RLS em relação ao LMS é que sua velocidade de convergência é, tipicamente, uma ordem de grandeza maior [3]. No entanto, apesar de

evitar a inversão explícita da matriz de autocorrelação, a complexidade computacional do algoritmo RLS é ainda substancialmente superior à do LMS. Na formulação apresentada, o RLS possui complexidade $\mathcal{O}(N^2)$, fazendo com que implementações de filtros adaptativos de ordem elevada demandem grande carga computacional.

A seguir, são apresentadas algumas importantes relações matemáticas que dizem respeito às variáveis do algoritmo RLS. Em (3.41), o erro *a priori* é calculado a partir do valor do vetor de coeficientes no instante $n - 1$. O erro *a priori*, em geral, é diferente do erro *a posteriori*

$$e(n) = d(n) - \mathbf{x}^T(n)\hat{\mathbf{h}}_{\text{LS}}(n) \quad (3.43)$$

cujos cálculos dependem do vetor de coeficientes no instante n . No entanto, é possível estabelecer uma relação entre $e(n)$ e $\epsilon(n)$. Substituindo (3.40) em (3.43), tem-se

$$\begin{aligned} e(n) &= d(n) - \mathbf{x}^T(n) \left[\hat{\mathbf{h}}_{\text{LS}}(n-1) + \mathbf{k}(n)\epsilon(n) \right] \\ &= d(n) - \mathbf{x}^T(n)\hat{\mathbf{h}}_{\text{LS}}(n-1) - \mathbf{x}^T(n)\mathbf{k}(n)\epsilon(n) \\ &= \epsilon(n) - \mathbf{x}^T(n)\mathbf{k}(n)\epsilon(n) \\ &= [1 - \mathbf{x}^T(n)\mathbf{k}(n)] \epsilon(n). \end{aligned} \quad (3.44)$$

Definindo

$$\gamma(n) = 1 - \mathbf{x}^T(n)\mathbf{k}(n) \quad (3.45)$$

observa-se que

$$\gamma(n) = \frac{e(n)}{\epsilon(n)}. \quad (3.46)$$

A variável $\gamma(n)$ é geralmente denominada na literatura de fator de conversão [3].

A soma dos erros quadráticos $J_{\text{LS}}(n)$ assume seu valor mínimo quando o vetor de coeficientes assume o valor particular $\hat{\mathbf{h}}_{\text{LS}}(n)$ dado por (3.32). Utilizando as definições de $\tilde{\mathbf{R}}_{\mathbf{x}}(n)$ e $\mathbf{r}_{\text{xd}}(n)$ e substituindo (3.32) em (3.25), o valor mínimo de $J_{\text{LS}}(n)$ [aqui denotado por $J_{\text{LS}}^{(\text{min})}(n)$] pode ser escrito como

$$\begin{aligned} J_{\text{LS}}^{(\text{min})}(n) &= \sum_{i=1}^n \lambda^{n-i} d^2(i) - 2\hat{\mathbf{h}}_{\text{LS}}^T(n)\mathbf{r}_{\text{xd}}(n) + \hat{\mathbf{h}}_{\text{LS}}^T(n)\tilde{\mathbf{R}}_{\mathbf{x}}(n)\hat{\mathbf{h}}_{\text{LS}}(n) \\ &= \sum_{i=1}^n \lambda^{n-i} d^2(i) - 2\hat{\mathbf{h}}_{\text{LS}}^T(n)\mathbf{r}_{\text{xd}}(n) + \hat{\mathbf{h}}_{\text{LS}}^T(n)\tilde{\mathbf{R}}_{\mathbf{x}}(n)\tilde{\mathbf{R}}_{\mathbf{x}}^{-1}(n)\mathbf{r}_{\text{xd}}(n) \\ &= \sum_{i=1}^n \lambda^{n-i} d^2(i) - \hat{\mathbf{h}}_{\text{LS}}^T(n)\mathbf{r}_{\text{xd}}(n). \end{aligned} \quad (3.47)$$

Pode-se ainda obter uma equação recursiva para $J_{\text{LS}}^{(\text{min})}(n)$ substituindo (3.30), (3.40) e

(3.39) em (3.47). Assim,

$$\begin{aligned}
J_{\text{LS}}^{(\min)}(n) &= \sum_{i=1}^{n-1} \lambda^{n-i} d^2(i) + d^2(n) - \lambda \mathbf{r}_{\mathbf{x}d}^{\text{T}}(n-1) \hat{\mathbf{h}}_{\text{LS}}(n-1) - \lambda \mathbf{r}_{\mathbf{x}d}^{\text{T}}(n-1) \mathbf{k}(n) \epsilon(n) \\
&\quad - \mathbf{x}^{\text{T}}(n) \hat{\mathbf{h}}_{\text{LS}}(n-1) d(n) - \mathbf{x}^{\text{T}}(n) \mathbf{k}(n) \epsilon(n) d(n) \\
&= \lambda J_{\text{LS}}^{(\min)}(n-1) + \epsilon(n) \left[\lambda \mathbf{r}_{\mathbf{x}d}^{\text{T}}(n-1) + \mathbf{x}^{\text{T}}(n) d(n) \right] \mathbf{k}(n) \\
&\quad + d(n) \left[d(n) - \mathbf{x}^{\text{T}}(n) \hat{\mathbf{h}}_{\text{LS}}(n-1) \right] \\
&= \lambda J_{\text{LS}}^{(\min)}(n-1) + d(n) \epsilon(n) - \epsilon(n) \mathbf{r}_{\mathbf{x}d}^{\text{T}}(n) \tilde{\mathbf{R}}_{\mathbf{x}}^{-1}(n) \mathbf{x}(n) \\
&= \lambda J_{\text{LS}}^{(\min)}(n-1) + \epsilon(n) \left[d(n) - \hat{\mathbf{h}}_{\text{LS}}^{\text{T}}(n) \mathbf{x}(n) \right] \\
&= \lambda J_{\text{LS}}^{(\min)}(n-1) + \epsilon(n) e(n). \tag{3.48}
\end{aligned}$$

Na literatura, o algoritmo RLS é muitas vezes escrito em termos do ganho de Kalman normalizado, definido como

$$\tilde{\mathbf{k}}(n) = \frac{\mathbf{k}(n)}{\gamma(n)}. \tag{3.49}$$

3.2.2 Algoritmo *Fast Transversal Filter* (FTF)

Conforme anteriormente apresentado, a principal desvantagem do algoritmo RLS é a sua elevada complexidade computacional. No entanto, existem disponíveis na literatura diversas estratégias para implementação do RLS que apresentam menor complexidade. Algumas estratégias tiram proveito da estrutura transversal do filtro adaptativo, apresentando complexidade $\mathcal{O}(N)$. Dentre essas, a de menor complexidade é o *fast transversal filter* (FTF) [5], que tem complexidade $7N$.

No algoritmo FTF, tira-se proveito da estrutura particular do vetor de entrada $\mathbf{x}(n)$ para os filtros transversais FIR. Nesses casos, pode-se relacionar o valor atual de $\mathbf{x}(n)$ e o seu valor passado $\mathbf{x}(n-1)$ através de

$$\begin{bmatrix} x(n) \\ \mathbf{x}(n-1) \end{bmatrix} = \mathbf{x}_{N+1}(n) = \begin{bmatrix} \mathbf{x}(n) \\ x(n-N) \end{bmatrix} \tag{3.50}$$

onde $\mathbf{x}_{N+1}(n)$ denota o vetor de entrada aumentado. Observa-se que (3.50) representa um aumento seguido de uma diminuição da ordem de $\mathbf{x}(n)$. Nessa mesma linha, pode-se definir também a matriz de autocorrelação amostral aumentada

$$\tilde{\mathbf{R}}_{\mathbf{x}_{N+1}}(n) = \sum_{i=1}^n \lambda^{n-i} \mathbf{x}_{N+1}(i) [\mathbf{x}_{N+1}(i)]^{\text{T}} \tag{3.51}$$

o ganho de Kalman aumentado

$$\mathbf{k}_{N+1}(n) = \left[\tilde{\mathbf{R}}_{\mathbf{x}_{N+1}}(n) \right]^{-1} \mathbf{x}_{N+1}(n) \quad (3.52)$$

e o fator de conversão aumentado

$$\gamma_{N+1}(n) = 1 - [\mathbf{x}_{N+1}(n)]^T \mathbf{k}_{N+1}(n). \quad (3.53)$$

Nesse caso, o ganho de Kalman aumentado normalizado é

$$\tilde{\mathbf{k}}_{N+1}(n) = \frac{\mathbf{k}_{N+1}(n)}{\gamma_{N+1}(n)}. \quad (3.54)$$

Para o FTF, o ganho de Kalman é obtido recursivamente de maneira similar à considerada em (3.50). O vetor aumentado $\mathbf{k}_{N+1}(n)$ é calculado a partir de $\mathbf{k}(n-1)$, enquanto $\mathbf{k}(n)$ é obtido através de uma operação de redução da ordem de $\mathbf{k}_{N+1}(n)$. Essa seqüência de operações é ilustrada na Figura 3.2 [21]. Os vetores auxiliares $\mathbf{a}(n)$ e $\mathbf{b}(n)$ mostrados na figura são, respectivamente, filtros preditores *forward* e *backward*. O preditor *forward* $\mathbf{a}(n)$ é utilizado no aumento da ordem de $\mathbf{k}(n-1)$, enquanto o preditor *backward* $\mathbf{b}(n)$ é utilizado na redução de ordem de $\mathbf{k}_{N+1}(n)$. As operações de aumento e de redução da ordem do ganho de Kalman e a atualização dos preditores são discutidas na seqüência.

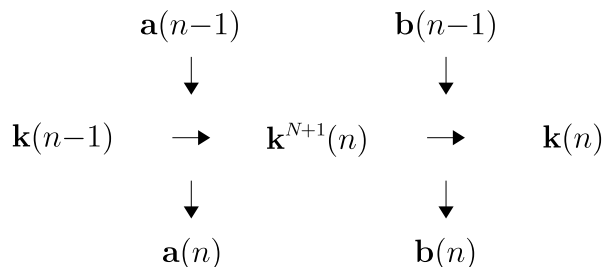


Figura 3.2: Ilustração do procedimento de atualização do ganho de Kalman no FTF.

3.2.2.1 Filtro Preditor Linear *Forward*

O filtro preditor *forward* $\mathbf{a}(n)$ tem como objetivo estimar a amostra atual do sinal de entrada $x(n)$ a partir do vetor de N amostras passadas desse sinal, denotado por $\mathbf{x}(n-1) = [x(n-1) \ x(n-2) \ \cdots \ x(n-N)]^T$. Considerando o critério dos mínimos quadrados para determinar $\mathbf{a}(n)$, o equacionamento obtido no desenvolvimento do algoritmo RLS pode ser utilizado. No caso do preditor *forward*, o vetor de entrada considerado é $\mathbf{x}(n-1)$, enquanto o sinal desejado é $x(n)$. Assim, modificando (3.32), o filtro preditor $\mathbf{a}(n)$ pode ser determinado através de

$$\mathbf{a}(n) = \tilde{\mathbf{R}}_{\mathbf{x}}^{-1}(n-1) \tilde{\mathbf{r}}_{\mathbf{x}\mathbf{x}}^f(n) \quad (3.55)$$

onde

$$\tilde{\mathbf{r}}_{\mathbf{x}\mathbf{x}}^f(n) = \sum_{i=1}^n \lambda^{n-i} \mathbf{x}(i-1)x(i) \quad (3.56)$$

denota a correlação cruzada amostral entre $\mathbf{x}(n-1)$ e $x(n)$. Adaptando (3.40) ao caso do preditor *forward*, obtém-se a equação recursiva de atualização de $\mathbf{a}(n)$. Assim,

$$\mathbf{a}(n) = \mathbf{a}(n-1) + \mathbf{k}(n-1)\phi(n) \quad (3.57)$$

onde $\phi(n)$ é erro de estimação *a priori* do filtro *forward*, dado por

$$\phi(n) = x(n) - \mathbf{a}^T(n-1)\mathbf{x}(n-1). \quad (3.58)$$

Já o erro da estimação *a posteriori* do preditor *forward* é

$$f(n) = x(n) - \mathbf{a}^T(n)\mathbf{x}(n-1) \quad (3.59)$$

$$= \gamma(n-1)\phi(n). \quad (3.60)$$

A soma dos erros quadráticos para o preditor *forward* é obtida adaptando (3.47) e (3.48). Portanto,

$$\begin{aligned} J_f(n) &= \sum_{i=1}^n \lambda^{n-i} x^2(i) - \mathbf{a}^T(n)\tilde{\mathbf{r}}_{\mathbf{x}\mathbf{x}}^f(n) \\ &= r_{00} - \mathbf{a}^T(n)\tilde{\mathbf{r}}_{\mathbf{x}\mathbf{x}}^f(n) \end{aligned} \quad (3.61)$$

$$= \lambda J_f(n-1) + \phi(n)f(n) \quad (3.62)$$

com

$$r_{00} = \sum_{i=1}^n \lambda^{n-i} x^2(i). \quad (3.63)$$

Utilizando (3.50), (3.56) e (3.63), pode-se reescrever $\tilde{\mathbf{R}}_{\mathbf{x}_{N+1}}(n)$ como

$$\begin{aligned} \tilde{\mathbf{R}}_{\mathbf{x}_{N+1}}(n) &= \sum_{i=1}^n \lambda^{n-i} \left\{ \begin{bmatrix} x(i) \\ \mathbf{x}(i-1) \end{bmatrix} \begin{bmatrix} x(i) & \mathbf{x}^T(i-1) \end{bmatrix} \right\} \\ &= \sum_{i=1}^n \lambda^{n-i} \left\{ \begin{bmatrix} x^2(i) & x(i)\mathbf{x}^T(i-1) \\ x(i)\mathbf{x}(i-1) & \mathbf{x}(i-1)\mathbf{x}^T(i-1) \end{bmatrix} \right\} \\ &= \begin{bmatrix} r_{00} & [\tilde{\mathbf{r}}_{\mathbf{x}\mathbf{x}}^f(n)]^T \\ \tilde{\mathbf{r}}_{\mathbf{x}\mathbf{x}}^f(n) & \tilde{\mathbf{R}}_{\mathbf{x}}(n-1) \end{bmatrix}. \end{aligned} \quad (3.64)$$

O complemento de Schur da matriz $\tilde{\mathbf{R}}_{\mathbf{x}_{N+1}}(n)$ é

$$s = r_{00} - [\tilde{\mathbf{r}}_{\mathbf{x}\mathbf{x}}^f(n)]^T \tilde{\mathbf{R}}_{\mathbf{x}}^{-1}(n-1)\tilde{\mathbf{r}}_{\mathbf{x}\mathbf{x}}^f(n). \quad (3.65)$$

Substituindo (3.55) e (3.61) em (3.65), verifica-se que

$$\begin{aligned} s &= r_{00} - \mathbf{a}^T(n) \tilde{\mathbf{r}}_{\mathbf{x}\mathbf{x}}^f(n) \\ &= J_f(n) \end{aligned} \quad (3.66)$$

e, utilizando (3.55), a inversa de $\tilde{\mathbf{R}}_{\mathbf{x}_{N+1}}(n)$ pode ser escrita como

$$\begin{aligned} \left[\tilde{\mathbf{R}}_{\mathbf{x}_{N+1}}(n) \right]^{-1} &= \frac{1}{J_f(n)} \begin{bmatrix} 1 & -\mathbf{a}^T(n) \\ -\mathbf{a}(n) & J_f(n) \tilde{\mathbf{R}}_{\mathbf{x}}^{-1}(n-1) + \mathbf{a}(n) \mathbf{a}^T(n) \end{bmatrix} \\ &= \begin{bmatrix} 0 & \mathbf{0}_N^T \\ \mathbf{0}_N & \tilde{\mathbf{R}}_{\mathbf{x}}^{-1}(n-1) \end{bmatrix} + \frac{1}{J_f(n)} \tilde{\mathbf{a}}(n) \tilde{\mathbf{a}}^T(n) \end{aligned} \quad (3.67)$$

com

$$\tilde{\mathbf{a}}(n) = \begin{bmatrix} 1 \\ -\mathbf{a}(n) \end{bmatrix}. \quad (3.68)$$

Pós-multiplicando (3.67) por $\mathbf{x}_{N+1}(n)$ e utilizando (3.52), (3.39) e (3.59), obtém-se

$$\begin{aligned} \mathbf{k}_{N+1}(n) &= \begin{bmatrix} 0 & \mathbf{0}_N^T \\ \mathbf{0}_N & \tilde{\mathbf{R}}_{\mathbf{x}}^{-1}(n-1) \end{bmatrix} \begin{bmatrix} x(n) \\ \mathbf{x}(n-1) \end{bmatrix} + \frac{\tilde{\mathbf{a}}(n) \tilde{\mathbf{a}}^T(n)}{J_f(n)} \begin{bmatrix} x(n) \\ \mathbf{x}(n-1) \end{bmatrix} \\ &= \begin{bmatrix} 0 \\ \mathbf{k}(n-1) \end{bmatrix} + \frac{f(n) \tilde{\mathbf{a}}(n)}{J_f(n)}. \end{aligned} \quad (3.69)$$

Agora, pré-multiplicando (3.69) por $\mathbf{x}_{N+1}(n)$ e substituindo (3.45) e (3.53), verifica-se que

$$\gamma_{N+1}(n) = \gamma(n-1) - \frac{f^2(n)}{J_f(n)}. \quad (3.70)$$

Substituindo (3.60) e (3.62) em (3.70), obtém-se

$$\begin{aligned} \gamma_{N+1}(n) &= \gamma(n-1) - \frac{f^2(n)}{J_f(n)} \\ &= \gamma(n-1) \left[1 - \frac{f(n) \phi(n)}{J_f(n)} \right] \end{aligned} \quad (3.71)$$

$$= \gamma(n-1) \lambda \frac{J_f(n-1)}{J_f(n)}. \quad (3.72)$$

Dessa forma, a expressão anterior pode ser utilizada para determinar o valor do fator de conversão aumentado.

Agora, substituindo (3.57) em (3.69) e rearranjando os termos, obtém-se

$$\mathbf{k}_{N+1}(n) = \left[1 - \frac{f(n) \phi(n)}{J_f(n)} \right] \begin{bmatrix} 0 \\ \mathbf{k}(n-1) \end{bmatrix} + \frac{f(n) \tilde{\mathbf{a}}(n-1)}{J_f(n)}. \quad (3.73)$$

Utilizando (3.49), (3.54) e (3.71), reescreve-se (3.73) como

$$\begin{aligned}\gamma_{N+1}(n)\tilde{\mathbf{k}}_{N+1}(n) &= \left[1 - \frac{f(n)\phi(n)}{J_f(n)}\right] \gamma(n-1) \begin{bmatrix} 0 \\ \tilde{\mathbf{k}}(n-1) \end{bmatrix} + \frac{f(n)\tilde{\mathbf{a}}(n-1)}{J_f(n)} \\ &= \gamma_{N+1}(n) \begin{bmatrix} 0 \\ \tilde{\mathbf{k}}(n-1) \end{bmatrix} + \frac{f(n)\tilde{\mathbf{a}}(n-1)}{J_f(n)}.\end{aligned}\quad (3.74)$$

Finalmente, dividindo os dois lados de (3.74) por $\gamma_{N+1}(n)$ e substituindo (3.72), pode-se obter $\tilde{\mathbf{k}}_{N+1}(n)$. Assim,

$$\tilde{\mathbf{k}}_{N+1}(n) = \begin{bmatrix} 0 \\ \tilde{\mathbf{k}}(n-1) \end{bmatrix} + \frac{\phi(n)}{\lambda J_f(n-1)} \tilde{\mathbf{a}}(n-1). \quad (3.75)$$

Então, utilizando (3.72) e (3.75), obtém-se $\gamma_{N+1}(n)$ e $\tilde{\mathbf{k}}_{N+1}(n)$. A seguir, utiliza-se um preditor *backward* para realizar a redução de ordem dessas variáveis.

3.2.2.2 Filtro Preditor Linear *Backward*

O filtro preditor *backward* é denotado por $\mathbf{b}(n)$. Seu objetivo é estimar a amostra passada $x(n-N)$ do sinal de entrada, a partir das N amostras mais recentes $x(n), x(n-1), \dots, x(n-N+1)$. Assim, nesse caso, o vetor de entrada é o próprio $\mathbf{x}(n)$ e o sinal desejado é $x(n-N)$. A equação utilizada para obter o vetor $\mathbf{b}(n)$ ótimo segundo o critério dos mínimos quadrados é

$$\mathbf{b}(n) = \tilde{\mathbf{R}}_{\mathbf{x}}^{-1}(n) \tilde{\mathbf{r}}_{\mathbf{x}\mathbf{x}}^b(n) \quad (3.76)$$

onde

$$\tilde{\mathbf{r}}_{\mathbf{x}\mathbf{x}}^b(n) = \sum_{i=1}^n \lambda^{n-i} \mathbf{x}(i) x(i-N) \quad (3.77)$$

denota a correlação cruzada amostral entre $\mathbf{x}(n)$ e $x(n-N)$. Nesse caso, a atualização de $\mathbf{b}(n)$ é obtida através de

$$\mathbf{b}(n) = \mathbf{b}(n-1) + \mathbf{k}(n)\beta(n) \quad (3.78)$$

onde $\beta(n)$ é erro de estimação *a priori* do filtro *backward*

$$\beta(n) = x(n-N) - \mathbf{b}^T(n-1)\mathbf{x}(n). \quad (3.79)$$

Já o erro da estimação *a posteriori* do preditor *backward* é

$$b(n) = x(n-N) - \mathbf{b}^T(n)\mathbf{x}(n) \quad (3.80)$$

$$= \gamma(n)\beta(n). \quad (3.81)$$

A soma dos erros quadráticos é

$$\begin{aligned} J_b(n) &= \sum_{i=1}^n \lambda^{n-i} x^2(i-N) - \mathbf{b}^T(n) \tilde{\mathbf{r}}_{\mathbf{x}\mathbf{x}}^b(n) \\ &= r_{NN} - \mathbf{b}^T(n) \tilde{\mathbf{r}}_{\mathbf{x}\mathbf{x}}^b(n) \end{aligned} \quad (3.82)$$

$$= \lambda J_b(n-1) + \beta(n) b(n) \quad (3.83)$$

com

$$r_{NN} = \sum_{i=1}^n \lambda^{n-i} x^2(i-N). \quad (3.84)$$

Utilizando (3.50), (3.77) e (3.84), pode-se reescrever $\tilde{\mathbf{R}}_{\mathbf{x}_{N+1}}(n)$ como

$$\begin{aligned} \tilde{\mathbf{R}}_{\mathbf{x}_{N+1}}(n) &= \sum_{i=1}^n \lambda^{n-i} \left\{ \begin{bmatrix} \mathbf{x}(i) \\ x(i-N) \end{bmatrix} \begin{bmatrix} \mathbf{x}(i) & x(i-N) \end{bmatrix} \right\} \\ &= \sum_{i=1}^n \lambda^{n-i} \left\{ \begin{bmatrix} \mathbf{x}(i)\mathbf{x}^T(i) & x(i-N)\mathbf{x}^T(i) \\ x(i-N)\mathbf{x}(i) & x^2(i-N) \end{bmatrix} \right\} \\ &= \begin{bmatrix} \tilde{\mathbf{R}}_{\mathbf{x}}(n) & \tilde{\mathbf{r}}_{\mathbf{x}\mathbf{x}}^b(n) \\ [\tilde{\mathbf{r}}_{\mathbf{x}\mathbf{x}}^b(n)]^T & r_{NN} \end{bmatrix}. \end{aligned} \quad (3.85)$$

Então, utilizando (3.76) e (3.82), o seu complemento de Schur é

$$\begin{aligned} s &= r_{NN} - [\tilde{\mathbf{r}}_{\mathbf{x}\mathbf{x}}^b(n)]^T \tilde{\mathbf{R}}_{\mathbf{x}}^{-1}(n) \tilde{\mathbf{r}}_{\mathbf{x}\mathbf{x}}^b(n) \\ &= r_{NN} - \mathbf{b}^T(n) \tilde{\mathbf{r}}_{\mathbf{x}\mathbf{x}}^b(n) \\ &= J_b(n). \end{aligned} \quad (3.86)$$

A inversa de $\tilde{\mathbf{R}}_{\mathbf{x}_{N+1}}(n)$ pode então ser escrita como

$$\begin{aligned} \left[\tilde{\mathbf{R}}_{\mathbf{x}_{N+1}}(n) \right]^{-1} &= \frac{1}{J_b(n)} \begin{bmatrix} J_b(n) \tilde{\mathbf{R}}_{\mathbf{x}}^{-1}(n) + \mathbf{b}(n)\mathbf{b}^T(n) & -\mathbf{b}(n) \\ -\mathbf{b}^T(n) & 1 \end{bmatrix} \\ &= \begin{bmatrix} \tilde{\mathbf{R}}_{\mathbf{x}}^{-1}(n) & \mathbf{0}_N \\ \mathbf{0}_N^T & 0 \end{bmatrix} + \frac{1}{J_b(n)} \tilde{\mathbf{b}}(n) \tilde{\mathbf{b}}^T(n) \end{aligned} \quad (3.87)$$

com

$$\tilde{\mathbf{b}}(n) = \begin{bmatrix} -\mathbf{b}(n) \\ 1 \end{bmatrix}. \quad (3.88)$$

Pós-multiplicando (3.87) por $\mathbf{x}_{N+1}(n)$ e substituindo (3.52), (3.39) e (3.80), obtém-se

$$\begin{aligned}\mathbf{k}_{N+1}(n) &= \begin{bmatrix} \tilde{\mathbf{R}}_{\mathbf{x}}^{-1}(n) & \mathbf{0}_N \\ \mathbf{0}_N^T & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x}(n) \\ x(n-N) \end{bmatrix} + \frac{\tilde{\mathbf{b}}(n)\tilde{\mathbf{b}}^T(n)}{J_b(n)} \begin{bmatrix} \mathbf{x}(n) \\ x(n-N) \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{k}(n) \\ 0 \end{bmatrix} + \frac{b(n)\tilde{\mathbf{b}}(n)}{J_b(n)}\end{aligned}\quad (3.89)$$

que pode ser reescrita em termos do ganho de Kalman normalizado como

$$\gamma_{N+1}(n)\tilde{\mathbf{k}}_{N+1}(n) = \gamma(n) \begin{bmatrix} \mathbf{k}(n) \\ 0 \end{bmatrix} + \frac{b(n)\tilde{\mathbf{b}}(n)}{J_b(n)}. \quad (3.90)$$

Igualando o último componente dos vetores nos dois lados de (3.90) e denotando por $\tilde{k}_{N+1}(n)$ o último elemento de $\tilde{\mathbf{k}}_{N+1}(n)$, verifica-se que

$$\tilde{k}_{N+1}(n) = \frac{b(n)}{\gamma_{N+1}(n)J_b(n)}. \quad (3.91)$$

Agora, pré-multiplicando (3.89) por $\mathbf{x}_{N+1}(n)$, utilizando (3.81) e (3.45) e rearranjando os termos, obtém-se

$$\gamma_{N+1}(n) = \gamma(n) - \frac{b^2(n)}{J_b(n)}. \quad (3.92)$$

Substituindo (3.81), (3.83) e (3.91) em (3.92), obtém-se

$$\begin{aligned}\gamma_{N+1}(n) &= \gamma(n) - \frac{b^2(n)}{J_b(n)} \\ &= \gamma(n) \left[1 - \frac{b(n)\beta(n)}{J_b(n)} \right]\end{aligned}\quad (3.93)$$

$$= \gamma(n) \left[1 - \tilde{k}_{N+1}(n)\gamma_{N+1}(n)\beta(n) \right]. \quad (3.94)$$

Assim, isolando $\gamma(n)$ na expressão anterior, tem-se

$$\gamma(n) = \frac{\gamma_{N+1}(n)}{1 - \tilde{k}_{N+1}(n)\gamma_{N+1}(n)\beta(n)}. \quad (3.95)$$

Dessa forma, (3.95) permite determinar o fator de conversão $\gamma(n)$ a partir da redução da ordem de $\gamma_{N+1}(n)$. Agora, substituindo (3.78) e (3.93) em (3.90) e rearranjando os termos, obtém-se

$$\begin{aligned}\gamma_{N+1}(n)\tilde{\mathbf{k}}_{N+1}(n) &= \gamma(n) \left[1 - \frac{b(n)\beta(n)}{J_b(n)} \right] \begin{bmatrix} \mathbf{k}(n) \\ 0 \end{bmatrix} + \frac{b(n)\tilde{\mathbf{b}}(n-1)}{J_b(n)} \\ &= \gamma_{N+1}(n) \begin{bmatrix} \tilde{\mathbf{k}}(n) \\ 0 \end{bmatrix} + \frac{b(n)\tilde{\mathbf{b}}(n-1)}{J_b(n)}.\end{aligned}\quad (3.96)$$

Dividindo os dois lados de (3.96) por $\gamma_{N+1}(n)$ e substituindo (3.91), verifica-se que

$$\begin{bmatrix} \tilde{\mathbf{k}}(n) \\ 0 \end{bmatrix} = \tilde{\mathbf{k}}_{N+1}(n) - \tilde{k}_{N+1}(n)\tilde{\mathbf{b}}(n-1). \quad (3.97)$$

Dessa forma, pode-se utilizar (3.97) para determinar o valor atualizado de $\tilde{\mathbf{k}}(n)$ a partir do vetor aumentado $\tilde{\mathbf{k}}_{N+1}(n)$.

É possível ainda obter uma expressão alternativa à (3.79) para o erro de estimação *a priori* do filtro *backward*. Para tal, utilizando (3.83), observa-se que (3.93) pode ser reescrita como

$$\gamma_{N+1}(n) = \gamma(n)\lambda \frac{J_b(n-1)}{J_b(n)}. \quad (3.98)$$

Substituindo (3.98) em (3.91) e rearranjando os termos, obtém-se

$$\beta(n) = \lambda J_b(n-1)\tilde{k}_{N+1}(n). \quad (3.99)$$

3.2.2.3 Adaptação dos Coeficientes

Utilizando (3.97), é possível determinar o valor do ganho de Kalman normalizado. Esse vetor pode então ser utilizado na determinação recursiva do vetor de coeficientes. Escrevendo (3.40) em termos de $\tilde{\mathbf{k}}(n)$ e substituindo (3.46), obtém-se

$$\begin{aligned} \hat{\mathbf{h}}_{\text{LS}}(n) &= \hat{\mathbf{h}}_{\text{LS}}(n-1) + \tilde{\mathbf{k}}(n)\gamma(n)\epsilon(n) \\ &= \hat{\mathbf{h}}_{\text{LS}}(n-1) + \tilde{\mathbf{k}}(n)e(n). \end{aligned} \quad (3.100)$$

Na Tabela (3.3), são apresentadas as equações que caracterizam as três etapas do algoritmo FTF, a saber: predição *forward*, predição *backward* e adaptação dos coeficientes. Também são mostrados os valores utilizados na inicialização do algoritmo [5]. Observa-se que a complexidade computacional total do FTF é $7N$.

3.2.3 Algoritmo *Stabilized Fast Transversal Filter* (SFTF)

O principal atrativo do algoritmo FTF, em relação ao RLS, é que a sua complexidade computacional é significativamente menor. No entanto, o FTF é reconhecido ser numericamente instável, quando implementado com aritmética de precisão finita [22].

Uma modificação do FTF, denominada *stabilized* FTF (SFTF) [22], visa melhorar as suas características numéricas. A idéia básica do SFTF é tirar proveito do fato de que certas quantidades, no FTF, podem ser calculadas de mais de uma maneira. Assim, comparando o resultado obtido por diferentes expressões, pode-se realizar medidas instantâneas dos erros numéricos presentes no algoritmo. Essas medidas, então, são reali-

Tabela 3.3: Algoritmo FTF

Inicialização do algoritmo
$\mathbf{a}(0) = \mathbf{0}_N \quad \mathbf{b}(0) = \mathbf{0}_N \quad \tilde{\mathbf{k}}(0) = \mathbf{0}_N$ $J_f(0) = \delta\lambda^N \quad J_b(0) = \delta \quad \gamma(0) = 1$
Predição <i>forward</i> - Cálculo de $\gamma_{N+1}(n)$ e $\tilde{\mathbf{k}}_{N+1}(n)$
$\phi(n) = x(n) - \mathbf{a}^T(n-1)\mathbf{x}(n-1)$ $f(n) = \gamma(n-1)\phi(n)$ $J_f(n) = \lambda J_f(n-1) + \phi(n)f(n)$ $\gamma_{N+1}(n) = \lambda \frac{J_f(n-1)}{J_f(n)}$ $\tilde{\mathbf{k}}_{N+1}(n) = \begin{bmatrix} 0 \\ \tilde{\mathbf{k}}(n-1) \end{bmatrix} + \frac{\phi(n)}{\lambda J_f(n-1)} \begin{bmatrix} 1 \\ -\mathbf{a}(n-1) \end{bmatrix}$ $\mathbf{a}(n) = \mathbf{a}(n-1) + \tilde{\mathbf{k}}(n-1)f(n)$
Predição <i>backward</i> - Cálculo de $\gamma(n)$ e $\tilde{\mathbf{k}}(n)$
$\beta(n) = \lambda J_b(n-1)\tilde{k}_{N+1}(n)$ $\gamma(n) = \frac{\gamma_{N+1}(n)}{1 - \tilde{k}_{N+1}(n)\gamma_{N+1}(n)\beta(n)}$ $b(n) = \gamma(n)\beta(n)$ $J_b(n) = \lambda J_b(n-1) + \beta(n)b(n)$ $\begin{bmatrix} \tilde{\mathbf{k}}(n) \\ 0 \end{bmatrix} = \tilde{\mathbf{k}}_{N+1}(n) - \tilde{k}_{N+1}(n) \begin{bmatrix} -\mathbf{b}(n-1) \\ 1 \end{bmatrix}$ $\mathbf{b}(n) = \mathbf{b}(n-1) + \tilde{\mathbf{k}}(n)b(n)$
Atualização dos coeficientes - Cálculo de $e(n)$ e $\hat{\mathbf{h}}_{\text{LS}}(n)$
$\epsilon(n) = d(n) - \mathbf{x}^T(n)\hat{\mathbf{h}}_{\text{LS}}(n-1)$ $e(n) = \gamma(n)\epsilon(n)$ $\hat{\mathbf{h}}_{\text{LS}}(n) = \hat{\mathbf{h}}_{\text{LS}}(n-1) + \tilde{\mathbf{k}}(n)e(n)$

mentadas, visando modificar a dinâmica da propagação dos erros numéricos e estabilizar os modos instáveis do FTF [22].

No SF-FTF, apenas a predição *backward* do algoritmo FTF é modificada. Em particular, são consideradas duas maneiras diferentes de calcular o erro de estimação *a priori* do preditor *backward* $\beta(n)$. A primeira é a mesma expressão usada no FTF, que é dada por

$$\beta_s(n) = \lambda J_b(n-1)\tilde{k}_{N+1}(n) \quad (3.101)$$

onde o subscrito s denota manipulação escalar [3]. Já a segunda é a própria definição de $\beta(n)$. Assim,

$$\beta_f(n) = x(n - N) - \mathbf{b}^T(n - 1)\mathbf{x}(n) \quad (3.102)$$

onde o subscrito f denota uma operação de filtragem. O valor final da variável $\beta(n)$ é obtido utilizando uma combinação convexa de $\beta_s(n)$ e $\beta_f(n)$. Portanto,

$$\beta(n) = \beta_s(n) + K [\beta_f(n) - \beta_s(n)] \quad (3.103)$$

onde K é uma constante que determina o peso relativo de $\beta_s(n)$ e $\beta_f(n)$. Dado que o valor de $\beta(n)$ é utilizado em várias expressões no FTF, utiliza-se em cada uma delas um valor particular $\beta_i(n)$ que é calculado considerando uma constante K_i . Dessa forma, o ajuste dos valores individuais das constantes K_i permite obter graus de liberdade adicionais no ajuste da propagação dos erros do FTF.

No SFTF, utiliza-se também uma expressão alternativa para o fator de conversão $\gamma(n)$, dada por

$$\gamma_a(n) = \lambda^N \frac{J_b(n)}{J_f(n)}. \quad (3.104)$$

Essa expressão apresenta um mecanismo diferente de propagação de erros numéricos do que a expressão utilizada no FTF [22], i.e.,

$$\gamma_s(n) = \frac{\gamma_{N+1}(n)}{1 - \tilde{k}_{N+1}(n)\gamma_{N+1}(n)\beta(n)}. \quad (3.105)$$

O valor de $\gamma(n)$ é utilizado em diferentes expressões do algoritmo FTF. Dessa forma, no SFTF, utiliza-se o valor de $\gamma_a(n)$ em determinados pontos e o de $\gamma_s(n)$ em outros.

Na Tabela 3.4, são mostradas as equações utilizadas na predição *backward* algoritmo SFTF. A inicialização, a predição *forward* e a adaptação dos coeficientes são realizadas de maneira idêntica à do FTF [considerando $\gamma(n) = \gamma_a(n)$]. Observa-se que o cálculo de $\beta_f(n)$ através de (3.102) faz com que a complexidade computacional total do SFTF seja $8N$.

3.2.4 Algoritmo *Fast Newton Transversal Filter* (FNNTF)

O algoritmo *Fast Newton Transversal Filter* (FNNTF) [6] é uma versão modificada do algoritmo FTF, que permite certa liberdade no ajuste do compromisso entre o seu desempenho e a sua complexidade computacional. Em geral, as características do FNNTF (desempenho e complexidade) ficam entre aquelas dos algoritmos LMS e FTF.

A equação de adaptação dos coeficientes do LMS e do RLS pode ser escrita como

Tabela 3.4: Algoritmo SFTF

Predição *backward* do algoritmo SFTF

$$\begin{aligned} \beta_s(n) &= \lambda J_b(n-1) \tilde{k}_{N+1}(n) \\ \beta_f(n) &= x(n-N) - \mathbf{b}^T(n-1) \mathbf{x}(n) \\ \beta_1(n) &= \beta_s(n) + K_1 [\beta_f(n) - \beta_s(n)] \\ \beta_2(n) &= \beta_s(n) + K_2 [\beta_f(n) - \beta_s(n)] \\ \gamma_s(n) &= \frac{\gamma_{N+1}(n)}{1 - \tilde{k}_{N+1}(n) \gamma_{N+1}(n) \beta_f(n)} \\ b_1(n) &= \gamma_s(n) \beta_1(n) \\ b_2(n) &= \gamma_s(n) \beta_2(n) \\ J_b(n) &= \lambda J_b(n-1) + \beta_2(n) b_2(n) \\ \gamma_a(n) &= \lambda^N \frac{J_b(n)}{J_f(n)} \\ \begin{bmatrix} \tilde{\mathbf{k}}(n) \\ 0 \end{bmatrix} &= \tilde{\mathbf{k}}_{N+1}(n) - \tilde{k}_{N+1}(n) \begin{bmatrix} -\mathbf{b}(n-1) \\ 1 \end{bmatrix} \\ \mathbf{b}(n) &= \mathbf{b}(n-1) + \tilde{\mathbf{k}}(n) b_1(n) \end{aligned}$$

um caso particular da seguinte equação geral:

$$\hat{\mathbf{h}}(n) = \hat{\mathbf{h}}(n-1) + \mathbf{R}^{-1}(n) \mathbf{x}(n) e(n) \quad (3.106)$$

onde $\mathbf{R}(n)$ é uma matriz $N \times N$. No caso do LMS, a matriz $\mathbf{R}(n)$ considerada é

$$\mathbf{R}(n) = \mathbf{R} = \frac{1}{\mu} \mathbf{I}_N. \quad (3.107)$$

Já no caso do RLS, a matriz considerada é a matriz de autocorrelação amostral, definida em (3.29). Assim,

$$\mathbf{R}(n) = \tilde{\mathbf{R}}_{\mathbf{x}}(n). \quad (3.108)$$

Utilizando o algoritmo FTF, o mesmo vetor de coeficientes obtido pelo RLS é determinado sem a necessidade do cálculo explícito de $\tilde{\mathbf{R}}_{\mathbf{x}}(n)$, através do uso de preditores *forward* e *backward*.

A idéia básica do FNFTF é que, nos casos em que o sinal de entrada $x(n)$ pode ser representando por um processo auto-regressivo de ordem L [AR(L), com $L < N$], a matriz $N \times N$ de autocorrelação amostral $\tilde{\mathbf{R}}_{\mathbf{x}}(n)$ pode ser obtida estendendo a matriz $L \times L$ de autocorrelação amostral $\tilde{\mathbf{R}}_{\mathbf{x}_L}(n)$. Em muitos casos, sinais de interesse prático (como, por exemplo, sinais de voz) podem ser representados como um processo AR(L). Dessa forma, em aplicações de cancelamento de eco, o FNFTF permite uma redução da complexidade

computacional (em comparação ao FTF) sem grande prejuízo no desempenho.

De maneira similar ao que ocorre no algoritmo RLS, o cálculo explícito de $\tilde{\mathbf{R}}_{\mathbf{x}_L}(n)$ e a extensão de $\tilde{\mathbf{R}}_{\mathbf{x}_L}(n)$ para $\tilde{\mathbf{R}}_{\mathbf{x}}(n)$ podem ser evitados. Nesse caso, assim como no FTF, são utilizados preditores *forward* e *backward*. Como o sinal de entrada $x(n)$ pode ser representado por um processo AR(L), esses preditores possuem ordem L . O cálculo dos coeficientes desses preditores é efetuado utilizando o equacionamento das predições *forward* e *backward* do FTF (ou da sua versão numericamente mais estável, o SFTF).

No cálculo do ganho de Kalman, os preditores obtidos (de ordem L) são estendidos com $N - L$ zeros. Assim, o ganho de Kalman normalizado aumentado é obtido como

$$\tilde{\mathbf{k}}_{N+1}(n) = \begin{bmatrix} 0 \\ \tilde{\mathbf{k}}(n-1) \end{bmatrix} + \frac{\phi(n)}{\lambda J_f(n-1)} \begin{bmatrix} 1 \\ -\mathbf{a}(n-1) \\ \mathbf{0}_{N-L} \end{bmatrix} \quad (3.109)$$

e o ganho de Kalman normalizado, como

$$\begin{bmatrix} \tilde{\mathbf{k}}(n) \\ 0 \end{bmatrix} = \tilde{\mathbf{k}}_{N+1}(n) - \frac{\beta(n_c)}{\lambda J_b(n_c-1)} \begin{bmatrix} \mathbf{0}_{N-L} \\ -\mathbf{b}(n_c-1) \\ 1 \end{bmatrix} \quad (3.110)$$

com

$$n_c = n - N + L. \quad (3.111)$$

As variáveis $\phi(n)$ e $\beta(n)$ representam, respectivamente, o erro de estimação a priori dos preditores *forward* e *backward* e $J_f(n)$ e $J_b(n)$ denotam, respectivamente, a soma dos erros quadráticos dos preditores *forward* e *backward*.

Na Tabela 3.5, são mostradas as equações que definem o algoritmo FNFTF e as variáveis obtidas utilizando a predição *forward* e *backward* do algoritmo FTF. Observando a Tabela 3.3, verifica-se que as predições do FTF possuem complexidade $6L$. O cálculo do ganho de Kalman não requer multiplicações adicionais, enquanto a adaptação dos coeficientes possui complexidade $2N$. Dessa forma, o FNFTF possui complexidade total $2N + 6L$. Verifica-se que, para $L = 0$, a complexidade é a mesma do algoritmo LMS, enquanto para $L = N$, obtém-se o algoritmo FTF. Dessa forma, escolhendo $0 < L < N$, pode-se obter um compromisso entre a simplicidade computacional do LMS e o desempenho do FTF.

Uma desvantagem do algoritmo FNFTF em relação ao FTF é que devem ser armazenados $L \times (N - L)$ valores adicionais na memória, devido ao uso do vetor $\mathbf{b}(n_c - 1)$ em (3.110). É possível eliminar essa necessidade de memória adicional se for utilizado um segundo algoritmo FTF, em paralelo, cujo sinal de entrada é $x(n_c)$. Nesse caso, no entanto, a complexidade computacional total torna-se $2N + 12L$ [23].

Tabela 3.5: Algoritmo FNTF

Inicialização do algoritmo		
$\tilde{\mathbf{k}}(0) = \mathbf{0}_N \quad \gamma(0) = 1$		
Variáveis obtidas do algoritmo FTF (ou SFTF)		
$\mathbf{a}(n-1)$	$J_f(n-1)$	$\phi(n)$
$\mathbf{b}(n_c-1)$	$J_b(n_c-1)$	$\beta(n_c)$
Cálculo do ganho de Kalman		
$\tilde{\mathbf{k}}_{N+1}(n) = \begin{bmatrix} 0 \\ \tilde{\mathbf{k}}(n-1) \end{bmatrix} + \frac{\phi(n)}{\lambda J_f(n-1)} \begin{bmatrix} 1 \\ -\mathbf{a}(n-1) \\ \mathbf{0}_{N-L} \end{bmatrix}$		
$\begin{bmatrix} \tilde{\mathbf{k}}(n) \\ 0 \end{bmatrix} = \tilde{\mathbf{k}}_{N+1}(n) - \frac{\beta(n_c)}{\lambda J_b(n_c-1)} \begin{bmatrix} \mathbf{0}_{N-L} \\ -\mathbf{b}(n_c-1) \\ 1 \end{bmatrix}$		
Cálculo do fator de conversão		
$\gamma(n) = \gamma(n-1) - \frac{\phi^2(n)}{\lambda J_f(n-1)} + \frac{\beta^2(n_c)}{\lambda J_b(n_c-1)}$		
Atualização dos coeficientes		
$\epsilon(n) = d(n) - \mathbf{x}^T(n) \hat{\mathbf{h}}_{\text{LS}}(n-1)$		
$e(n) = \gamma(n) \epsilon(n)$		
$\hat{\mathbf{h}}_{\text{LS}}(n) = \hat{\mathbf{h}}_{\text{LS}}(n-1) + \tilde{\mathbf{k}}(n) e(n)$		

3.3 Considerações

Neste capítulo, foram apresentados conceitos básicos de filtragem adaptativa e alguns dos algoritmos adaptativos mais utilizados em aplicações práticas. Um deles é o algoritmo LMS, cujas principais características são a sua robustez, simplicidade e baixa complexidade computacional ($2N$). No entanto, em alguns casos, o LMS pode apresentar baixa velocidade de convergência. Outro algoritmo bastante utilizado é o RLS, que apresenta, em geral, velocidade de convergência uma ordem de grandeza maior do que a do LMS. No entanto, o RLS apresenta elevada complexidade computacional [$\mathcal{O}(N^2)$].

Uma estratégia eficiente de implementação do RLS é o algoritmo FTF, que tira proveito da estrutura transversal do filtro adaptativo e apresenta complexidade $7N$. Uma desvantagem do FTF é que esse algoritmo pode apresentar instabilidades numéricas em implementações práticas. Uma modificação do FTF, denominada SFTF, visa melhorar suas características numéricas, com um pequeno aumento de complexidade ($8N$).

Outra modificação do FTF é o algoritmo FNTF. Nesse algoritmo, considera-se que

o sinal de entrada pode ser representado por um processo $AR(L)$. Dessa forma, obtém-se uma redução na ordem dos preditores do FTF, reduzindo assim a complexidade computacional para $2N + 6L$. Ajustando o valor de L de zero até N , obtém-se um compromisso entre a baixa complexidade do LMS e o desempenho superior do FTF.

4 Detecção de Double-Talk

Neste capítulo, são discutidas algumas estratégias utilizadas para a detecção de *double-talk* em uma ligação telefônica. O procedimento geral de detecção de *double-talk* e alguns algoritmos utilizados para tal função são apresentados e discutidos.

4.1 Detectores de *Double-Talk*

Na Figura 4.1, é mostrada a estrutura básica de um cancelador de eco. Como discutido nos Capítulos 2 e 3, o algoritmo adaptativo em um cancelador utiliza o sinal de voz do usuário do *far-end* $x(n)$ e o seu eco $d(n)$ para refinar a medida $\hat{h}(n)$ da resposta ao impulso $h(n)$. No entanto, não se tem acesso diretamente ao sinal de eco $d(n)$. O sinal disponível para o cancelador de eco é $s(n)$, resultado da soma de $d(n)$ com o sinal de voz do usuário no *near-end* $v(n)$. Assim, deve existir um mecanismo de controle que diminua ou interrompa a adaptação dos coeficientes nos períodos em que existe, simultaneamente, eco do usuário no *far-end* e sinal de voz do usuário no *near-end*. A parte principal desse mecanismo de controle é o *detector de double-talk* (DTD) [17].

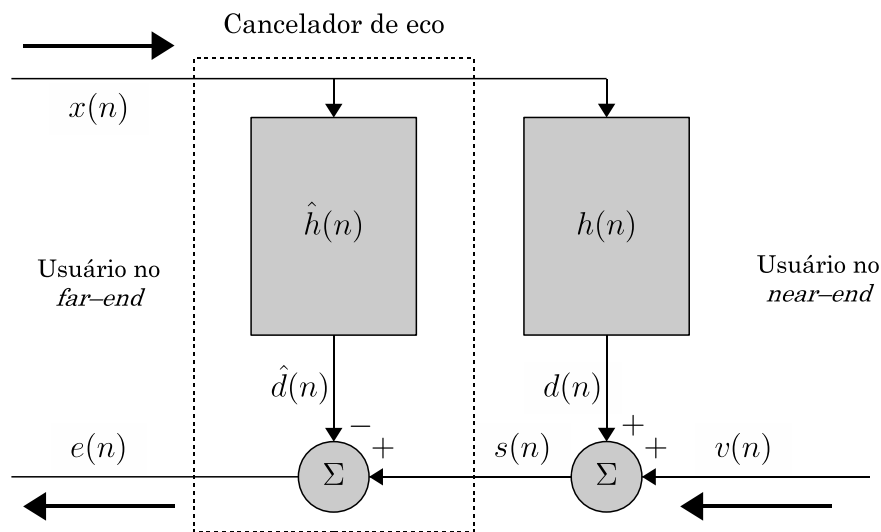


Figura 4.1: Estrutura básica de um cancelador de eco.

Existem diferentes métodos para detecção de *double-talk* propostos na literatura.

Em geral, esses métodos operam como descrito a seguir:

1. Uma variável de decisão $\xi(n)$ é calculada, utilizando os sinais disponíveis $x(n)$, $s(n)$, $e(n)$ e $\hat{d}(n)$, e os coeficientes da resposta ao impulso $\hat{h}(n)$.
2. A variável $\xi(n)$ é comparada a um limiar pré-definido T e *double-talk* é assumido quando $\xi(n) < T$.
3. Uma vez que *double-talk* é assumido, a adaptação dos coeficientes é interrompida por um número mínimo de amostras N_{hold} .
4. Se a condição $\xi(n) < T$ se mantiver falsa por mais do que N_{hold} amostras, a adaptação dos coeficientes recomeça.

Algumas abordagens para detecção de *double-talk* podem falhar sob certas condições na prática. Em particular, para algumas técnicas, não se pode dispor de um limiar T que seja independente das características do sistema (como, por exemplo, potência dos sinais envolvidos ou atenuação da resposta ao impulso). Dessa forma, o valor de T deve ser ajustado para cada caso, o que pode ser indesejável em certas aplicações.

Uma outra condição que pode levar à falha é a dependência do algoritmo de detecção de *double-talk* do valor dos coeficientes do filtro adaptativo $\hat{h}(n)$. Quando essa dependência existe, alguns fatores devem ser levados em conta, a saber:

- Uma falha na detecção de *double-talk* pode fazer o algoritmo adaptativo divergir, o que por consequência pode impactar a detecção de *double-talk* nos instantes seguintes.
- Uma mudança abrupta na resposta ao impulso do sistema pode ser interpretada de forma errada como sendo *double-talk*, impedindo a adaptação do filtro à nova condição.
- Na inicialização do cancelador de eco [ou seja, quando $\hat{h}(n) = 0$], deve existir uma estratégia que permita a adaptação do filtro antes do início da detecção de *double-talk*.

De maneira geral, o desempenho de um detector de *double-talk* pode ser caracterizado pelas seguintes grandezas:

- *Probabilidade de falso alarme* P_f . Probabilidade de falsa detecção de *double-talk*, ou seja, de *double-talk* ser declarado quando não há atividade do usuário do *near-end* [$v(n) = 0$].

- *Probabilidade de detecção* P_d . Probabilidade da correta detecção de *double-talk*, ou seja, de *double-talk* ser declarado quando existe atividade do usuário do *near-end* [$v(n) \neq 0$].
- *Probabilidade de perda* $P_m = 1 - P_d$. Probabilidade de *double-talk* não ser detectado quando existe atividade do usuário do *near-end* [$v(n) \neq 0$].

Tipicamente, existe uma relação de compromisso entre a baixa probabilidade de falso alarme P_f e a baixa probabilidade de perda P_m , que é controlada pelo valor do limiar T . Em sistemas de cancelamento de eco, a penalidade associada às perdas é elevada quando comparada à dos falsos alarmes, já que essas situações podem acarretar divergência do filtro adaptativo. Já os falsos alarmes causam, em geral, uma redução na velocidade de convergência do filtro.

Algumas técnicas conhecidas para detecção de *double-talk* são discutidas na seqüência.

4.2 Algoritmo Geigel

O algoritmo *Geigel* [7] é um algoritmo clássico e bastante simples para detecção de *double-talk*. Nesse algoritmo, assume-se que a potência média do sinal de eco é menor do que as dos sinais de voz dos usuários do *far-end* e do *near-end*. Dessa forma, nas situações em que o sinal $s(n)$ recebido pelo cancelador de eco tem baixa potência [em relação ao sinal $x(n)$], pode-se considerar que não há atividade do usuário do *near-end* (e, portanto, não há *double-talk*).

A variável de detecção considerada é

$$\xi(n)_G = \frac{\|\mathbf{x}(n)\|_\infty}{|s(n)|} \quad (4.1)$$

onde $\mathbf{x}(n) = [x(n) \ x(n-1) \ \dots \ x(n-N+1)]^T$ é o vetor de amostras do *far-end*, N denota o número de amostras passadas consideradas no cálculo de $\xi(n)_G$ e $\|\cdot\|_\infty$ caracteriza a norma infinita de um vetor. Geralmente, o valor de N corresponde à ordem do filtro adaptativo considerado, visando assim abranger todo o tempo de atraso do caminho do eco.

O limiar T é escolhido de forma a compensar um valor específico de atenuação média do caminho do eco. Uma desvantagem do algoritmo Geigel é que, em muitas situações, não é possível conhecer *a priori* esse valor de atenuação.

4.3 Método da Correlação Cruzada

Um outro método de detecção de *double-talk* utiliza a correlação entre o vetor de amostras do *far-end* $\mathbf{x}(n)$ e o sinal do *near-end* $s(n)$ para detecção de *double-talk* [10]. A medida de correlação considerada é o vetor de coeficientes de correlação cruzada, dado por

$$\begin{aligned} \mathbf{c}_{\mathbf{x}s} &= \frac{\mathbb{E}[\mathbf{x}(n)s(n)]}{\sqrt{\mathbb{E}[x^2(n)]\mathbb{E}[s^2(n)]}} \\ &= \frac{\mathbf{r}_{\mathbf{x}s}}{\sigma_x \sigma_s} \end{aligned} \quad (4.2)$$

onde $\mathbb{E}(\cdot)$ denota o valor esperado, $\mathbf{r}_{\mathbf{x}s} = \mathbb{E}[\mathbf{x}(n)s(n)]$ é o vetor de correlação cruzada entre $\mathbf{x}(n)$ e $s(n)$ e σ_x e σ_s são os desvios-padrão de $x(n)$ e $s(n)$, respectivamente.

O coeficiente de correlação cruzada aproxima-se da unidade quando a correlação é máxima e de zero quando é mínima. Assim, se os valores dos componentes de $\mathbf{c}_{\mathbf{x}s}$ são baixos [indicando baixa correlação entre $x(n)$ e $s(n)$], pode-se assumir que o sinal $s(n)$ não é composto inteiramente por eco (o que caracteriza uma situação de *double-talk*). Dessa maneira, a variável de detecção do método da correlação cruzada é definida como

$$\begin{aligned} \xi_{\text{CC}} &= \|\mathbf{c}_{\mathbf{x}s}\|_{\infty} \\ &= \frac{\|\mathbf{r}_{\mathbf{x}s}\|_{\infty}}{\sigma_x \sigma_s}. \end{aligned} \quad (4.3)$$

Em aplicações práticas, as grandezas estatísticas utilizadas em (4.2) devem ser estimadas a partir de $x(n)$ e $s(n)$, que geralmente são não-estacionários. Essas grandezas podem ser obtidas recursivamente como segue:

$$\tilde{\mathbf{r}}_{\mathbf{x}s}(n) = \mathbf{x}(n)s(n) + \lambda\tilde{\mathbf{r}}_{\mathbf{x}s}(n-1) \quad (4.4)$$

$$\tilde{\sigma}_x^2(n) = x^2(n) + \lambda\tilde{\sigma}_x^2(n-1) \quad (4.5)$$

$$\tilde{\sigma}_s^2(n) = s^2(n) + \lambda\tilde{\sigma}_s^2(n-1) \quad (4.6)$$

onde $0 \leq \lambda \leq 1$ é o fator de esquecimento. Assim, a variável de detecção pode ser obtida como

$$\begin{aligned} \xi_{\text{CC}}(n) &= \|\mathbf{c}_{\mathbf{x}s}(n)\|_{\infty} \\ &= \frac{\|\tilde{\mathbf{r}}_{\mathbf{x}s}(n)\|_{\infty}}{\tilde{\sigma}_x(n)\tilde{\sigma}_s(n)}. \end{aligned} \quad (4.7)$$

A principal desvantagem do método da correlação cruzada é que o valor ótimo do limiar T pode ter grande variação dependendo das condições de operação [12].

4.4 Método da Coerência Espectral

Em vez de utilizar diretamente a correlação cruzada, é possível utilizar a coerência espectral entre os sinais $x(n)$ e $s(n)$ como um indicativo da presença de *double-talk* [9].

A coerência espectral entre os sinais $x(n)$ e $s(n)$ é definida como

$$\gamma_{xs}^2(\omega) = \frac{|\Phi_{xs}(\omega)|^2}{\Phi_x(\omega)\Phi_s(\omega)} \quad (4.8)$$

onde $\Phi_x(\omega)$ e $\Phi_s(\omega)$ representam, respectivamente, a densidade espectral de potência de $x(n)$ e $s(n)$,

$$\Phi_x(\omega) = \sum_{k=-\infty}^{\infty} r_x(k)e^{-j\omega k} \quad (4.9)$$

e

$$\Phi_s(\omega) = \sum_{k=-\infty}^{\infty} r_s(k)e^{-j\omega k}, \quad (4.10)$$

$\Phi_{xs}(\omega)$ é a densidade espectral cruzada de potência entre $x(n)$ e $s(n)$,

$$\Phi_{xs}(\omega) = \sum_{k=-\infty}^{\infty} r_{xs}(k)e^{-j\omega k} \quad (4.11)$$

e $r_x(k)$, $r_s(k)$ e $r_{xs}(k)$ denotam, respectivamente, as autocorrelações de $x(n)$ e $s(n)$ e a correlação cruzada entre esses dois sinais. Assim como o coeficiente de correlação cruzada, a coerência espectral aproxima-se da unidade quando a correlação entre $x(n)$ e $s(n)$ é elevada e de zero quando a correlação é baixa. Dessa forma, valores baixos de $\gamma_{xs}^2(\omega)$ podem ser utilizados como um indicativo da presença de *double-talk*.

No método da coerência espectral, a variável de decisão considerada é uma média de valores de $\gamma_{xs}^2(\omega)$ avaliados em determinadas frequências. Assim,

$$\xi_{\text{CE}} = \frac{1}{M} \sum_{m=0}^{M-1} \gamma_{xs}^2(\omega_m) \quad (4.12)$$

onde $\omega_0, \omega_1, \dots, \omega_{M-1}$ representam frequências de interesse.

Em aplicações práticas, a coerência espectral deve ser estimada a partir dos sinais $x(n)$ e $s(n)$ recebidos pelo cancelador de eco. Uma possível abordagem é estimar a coerência espectral através do processamento de blocos de amostras dos sinais de entrada. Os i -ésimos blocos de amostras são denotados por $x_B^i(n)$ e $s_B^i(n)$ e são formados segmentando os sinais $x(n)$ e $s(n)$ por uma janela de L amostras. Tipicamente, cada novo bloco contém 5 a 10 novas amostras. Para cada bloco i , são estimadas as densidades espectrais

de potência $\Phi_x^i(\omega)$, $\Phi_s^i(\omega)$ e $\Phi_{xs}^i(\omega)$ através do método multi-janelas [24]. Portanto,

$$\hat{\Phi}_x^i(\omega) = \frac{1}{K} \sum_{k=0}^{K-1} \frac{|X_k^i(\omega)|^2}{\lambda_k} \quad (4.13)$$

$$\hat{\Phi}_s^i(\omega) = \frac{1}{K} \sum_{k=0}^{K-1} \frac{|S_k^i(\omega)|^2}{\lambda_k} \quad (4.14)$$

$$\hat{\Phi}_{xs}^i(\omega) = \frac{1}{K} \sum_{k=0}^{K-1} \frac{S_k^i(\omega) [X_k^i(\omega)]^*}{\lambda_k} \quad (4.15)$$

onde $X_k^i(\omega)$ e $S_k^i(\omega)$ denotam, respectivamente, o k -ésimo autoespectro dos blocos de amostras $x_B^i(n)$ e $s_B^i(n)$, λ_k é o autovalor associado ao k -ésimo autoespectro e K representa o número de janelas consideradas. Os k -ésimos autoespectros $X_k^i(\omega)$ e $S_k^i(\omega)$ são, respectivamente, dados por

$$X_k^i(\omega) = \sum_{n=0}^{L-1} x_B^i(n) \phi_k(n) e^{-j\omega n} \quad (4.16)$$

$$S_k^i(\omega) = \sum_{n=0}^{L-1} s_B^i(n) \phi_k(n) e^{-j\omega n} \quad (4.17)$$

onde $\phi_k(n)$ representa a seqüência de Slepian de ordem k [24]. As seqüências de Slepian têm a propriedade de apresentarem seu espectro maximamente concentrado em uma determinada largura de banda W , atendendo à seguinte equação:

$$\lambda_k \phi_k(n) = \sum_{m=0}^{L-1} \left\{ \frac{\text{sen} [2\pi W(n-m)]}{\pi W(n-m)} \phi_k(m) \right\}. \quad (4.18)$$

Em geral, são utilizados

$$K = \lfloor 2LW \rfloor \quad (4.19)$$

autoespectros no cálculo das densidades espectrais.

Na prática, (4.16) e (4.17) podem ser calculadas de forma eficiente através da FFT. Assim, a estimativa da coerência espectral para o i -ésimo bloco de amostras pode ser escrita como

$$[\gamma_{xs}^i(k)]^2 = \frac{|\hat{\Phi}_{xs}^i(k)|^2}{\hat{\Phi}_x^i(k) \hat{\Phi}_s^i(k)} \quad (4.20)$$

onde k denota o índice da FFT. Nesse caso, a variável de detecção é obtida por

$$\xi_{\text{CE}}^i = \frac{1}{M} \sum_{m=0}^{M-1} [\gamma_{xs}^i(k_m)]^2 \quad (4.21)$$

onde k_0, k_1, \dots, k_{M-1} denotam os índices da FFT que correspondem às freqüências de

interesse. Usualmente, considera-se uma faixa de frequência de aproximadamente 300 Hz a 2000 Hz para o cálculo de ξ_{CE}^i .

As principais desvantagens do método da coerência espectral para a detecção de *double-talk* consistem no seu elevado custo computacional (principalmente para valores grandes de L) e a dificuldade na determinação de um valor do limiar T que apresente desempenho adequado para uma grande variedade de condições de operação.

4.5 Método da Correlação Cruzada Normalizada

O método da correlação cruzada normalizada (NCC) para detecção de *double-talk* é baseado no método da correlação entre $\mathbf{x}(n)$ e $s(n)$. A sua principal vantagem é obter um valor constante para $\xi(n)$ quando $v(n) = 0$, independente de quaisquer características do sistema.

Da Figura 4.1, observa-se que o sinal $s(n)$ recebido pelo cancelador de eco é

$$s(n) = d(n) + v(n). \quad (4.22)$$

Dessa forma, a variância do sinal $s(n)$ é

$$\sigma_s^2 = \sigma_d^2 + \sigma_v^2. \quad (4.23)$$

Já a variância do sinal $d(n)$ pode ser escrita como

$$\sigma_d^2 = \mathbf{h}^T \mathbf{R}_x \mathbf{h} \quad (4.24)$$

onde o vetor $\mathbf{h} = [h(0) \ h(1) \ \dots \ h(N-1)]^T$ caracteriza a resposta ao impulso do sistema. Dado que $d(n) = \mathbf{h}^T \mathbf{x}(n)$, tem-se

$$\mathbf{h} = \mathbf{R}_x^{-1} \mathbf{r}_{xd}. \quad (4.25)$$

Assim, pode-se reescrever (4.24) como

$$\sigma_d^2 = \mathbf{r}_{xd}^T \mathbf{R}_x^{-1} \mathbf{r}_{xd}. \quad (4.26)$$

Então, assumindo que o sinal de entrada $x(n)$ e o sinal de voz do *near-end* $v(n)$ são não-correlacionados, obtém-se

$$\begin{aligned} \mathbf{r}_{xs} &= E[\mathbf{x}(n)s(n)] = E\{\mathbf{x}(n)[d(n) + v(n)]\} \\ &= E[\mathbf{x}(n)d(n)] \\ &= \mathbf{r}_{xd}. \end{aligned} \quad (4.27)$$

Assim, σ_d^2 pode ser expresso como

$$\sigma_d^2 = \mathbf{r}_{\mathbf{x}s}^T \mathbf{R}_{\mathbf{x}}^{-1} \mathbf{r}_{\mathbf{x}s}. \quad (4.28)$$

A variável de detecção do NCC é obtida calculando-se a razão entre σ_d^2 e σ_s^2 . Portanto,

$$\xi_{\text{NCC}} = \frac{\sigma_d^2}{\sigma_s^2} = \frac{\mathbf{r}_{\mathbf{x}s}^T \mathbf{R}_{\mathbf{x}}^{-1} \mathbf{r}_{\mathbf{x}s}}{\sigma_s^2}. \quad (4.29)$$

O cálculo de ξ_{NCC} , de acordo com (4.29), depende apenas dos sinais $x(n)$ e $s(n)$, que são os sinais disponíveis para o cancelador de eco.

Agora, reescrevendo (4.29) como

$$\xi_{\text{NCC}} = \frac{\sigma_d^2}{\sigma_s^2} = \frac{\sigma_d^2}{\sigma_d^2 + \sigma_v^2} \quad (4.30)$$

observa-se que, quando existe *double-talk* [$v(n) \neq 0$], o denominador de (4.30) torna-se maior do que seu numerador. Dessa forma, a detecção de *double-talk* pode ser realizada considerando $T = 1$. Contudo, costuma-se na prática utilizar um valor $T < 1$, visando compensar possíveis erros de estimação das grandezas estatísticas bem como a presença de ruído.

Assim como no método da correlação cruzada, as grandezas estatísticas requeridas para o cálculo de ξ_{NCC} podem ser estimadas recursivamente. As estimativas do vetor de correlação cruzada entre $\mathbf{x}(n)$ e $s(n)$ e da variância do sinal $s(n)$ são obtidas, respectivamente, por (4.4) e (4.6). Já a estimativa da matriz de autocorrelação de $\mathbf{x}(n)$ pode ser determinada através de (3.29). Dessa forma, a variável de detecção pode ser obtida como

$$\xi_{\text{NCC}}(n) = \frac{\mathbf{r}_{\mathbf{x}s}(n)^T \mathbf{R}_{\mathbf{x}}^{-1}(n) \mathbf{r}_{\mathbf{x}s}(n)}{\sigma_s^2(n)}. \quad (4.31)$$

Para valores elevados de N , a inversão da matriz $\mathbf{R}_{\mathbf{x}}(n)$ pode acarretar elevada carga computacional. Assim, visando reduzir a complexidade da implementação, pode-se evitar o cálculo de $\mathbf{R}_{\mathbf{x}}^{-1}(n)$ assumindo que o filtro adaptativo tenha convergido. Nesse caso, tem-se

$$\mathbf{h} = \mathbf{R}_{\mathbf{x}}^{-1} \mathbf{r}_{\mathbf{x}s} \approx \hat{\mathbf{h}}(n) \quad (4.32)$$

onde $\hat{\mathbf{h}}(n) = [\hat{h}_0(n) \ \hat{h}_1(n) \ \cdots \ \hat{h}_{N-1}(n)]^T$ é o vetor de coeficientes do filtro adaptativo. Assim, é possível aproximar (4.31) por

$$\xi_{\text{NCC}}(n) \approx \frac{\mathbf{r}_{\mathbf{x}s}^T(n) \hat{\mathbf{h}}(n)}{\sigma_s^2(n)}. \quad (4.33)$$

Assumindo ainda a convergência do filtro adaptativo, uma outra estratégia para simplificar

a implementação do NCC é utilizar a réplica do eco $\hat{d}(n)$ no cálculo da variância do sinal $d(n)$. Assim,

$$\sigma_d^2(n) \approx \sigma_{\hat{d}}^2(n) = \hat{d}^2(n) + \lambda \sigma_d^2(n-1). \quad (4.34)$$

Então, substituindo (4.34) em (4.31), obtém-se a aproximação

$$\xi_{\text{NCC}}(n) \approx \frac{\sigma_{\hat{d}}^2(n)}{\sigma_s^2(n)}. \quad (4.35)$$

O cálculo aproximado de $\xi_{\text{NCC}}(n)$ através de (4.33) ou (4.35) é sedutor do ponto de vista de complexidade computacional. No entanto, em ambos os casos, a qualidade da aproximação (e, portanto, o desempenho do detector de *double-talk*) depende de a convergência dos coeficientes do filtro adaptativo ter sido alcançada.

4.6 Resultados de Simulação

Nesta seção, são apresentados resultados experimentais visando avaliar o desempenho das técnicas apresentadas para detecção de *double-talk*. A maneira escolhida aqui para essa avaliação é através do valor de P_m obtido para um dado valor de P_f . Nos experimentos realizados, a probabilidade de falso alarme P_f é medida como

$$P_f = \frac{N_{x,\text{DT}}}{N_x} \quad (4.36)$$

onde N_x é o número de amostras em que há somente atividade do usuário do *far-end* [$x(n) \neq 0$ e $v(n) = 0$] e $N_{x,\text{DT}}$ é o número de amostras desse conjunto nas quais é declarado *double-talk*. Já a probabilidade de perda P_m é medida como

$$P_m = 1 - \frac{N_{x,v,\text{DT}}}{N_{x,v}} \quad (4.37)$$

onde $N_{x,v}$ é o número de amostras em que há atividade dos usuários do *far-end* e do *near-end* [$x(n) \neq 0$ e $v(n) \neq 0$] e $N_{x,v,\text{DT}}$ é o número de amostras desse conjunto nas quais é declarado *double-talk*. Nos experimentos realizados, o valor de T é ajustado de modo a obter $P_f = 0, 1$.

Quando se comparam diferentes técnicas para detecção de *double-talk*, é interessante medir P_m para diferentes valores da razão *near-end* para *far-end* (NFR), a qual caracteriza uma medida da energia do sinal do *near-end* em relação à do *far-end*. Assim,

$$\text{NFR} = 10 \log_{10} \left(\frac{\sigma_v^2}{\sigma_y^2} \right). \quad (4.38)$$

Nos testes realizados, foram utilizado 10 pares de arquivos de voz como sinais de

far-end e *near-end*. Esses arquivos, extraídos de uma base de dados de sinais de voz, são amostrados em 8 kHz e possuem em média um minuto de duração. Para cada par de arquivos, os diferentes valores de NFR são obtidos aplicando um ganho ao sinal correspondente ao *near-end*. Já os sinais de eco são obtidos através da convolução dos sinais referentes ao *far-end* com diferentes vetores de coeficientes. Cada um desses vetores corresponde às primeiras $N_h = 256$ amostras de uma resposta ao impulso medida (através de um processo de identificação de sistemas) em uma situação real de eco de linha.

O desempenho do método da correlação cruzada e do NCC depende do fator de esquecimento λ utilizado na estimação recursiva das estatísticas dos sinais envolvidos. Na Figura 4.2, é mostrada a curva de probabilidade de perda P_m obtida pelo NCC, considerando diversos valores de λ . Nesse experimento, realizado com um par de arquivos de *far-end* e *near-end*, a variável de detecção é calculada através de (4.31) e são considerados $N = 256$, $P_f = 0,1$ e NFR = -5 dB. Observa-se que os melhores resultados são obtidos para valores de λ inferiores a 0,996. Essa característica é também observada considerando outros valores de NFR e P_f . Assim, nos experimentos realizados, é utilizado $\lambda = 0,995$.

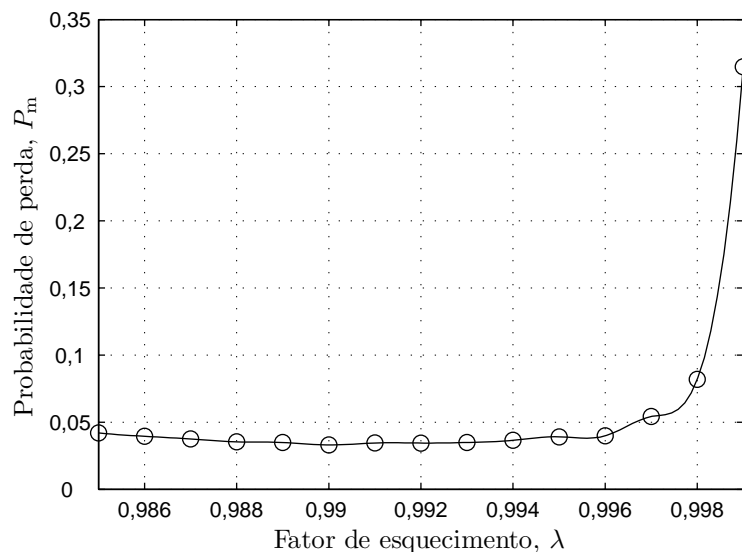


Figura 4.2: Influência do fator de esquecimento λ no método da correlação cruzada normalizada para NFR = -5 dB e probabilidade de falso alarme $P_f = 0,1$.

Na Figura 4.3, é mostrada uma comparação entre os desempenhos do algoritmo de Geigel, do método da correlação cruzada, do método da coerência espectral e do NCC. São considerados valores de NFR entre -20 dB e 10 dB, que correspondem a valores usualmente observados em aplicações práticas. Para cada método, é determinado um valor para o limiar T de modo que, levando em conta todos os valores de NFR, obtenha-se na média $P_f = 0,1$. No método da coerência espectral, são utilizados blocos com $L = 256$

amostras, tomando 128 novas amostras por bloco e $W = 0,02$. A faixa de frequências considerada no cálculo da variável de detecção desse método é de 280 Hz até 2200 Hz.

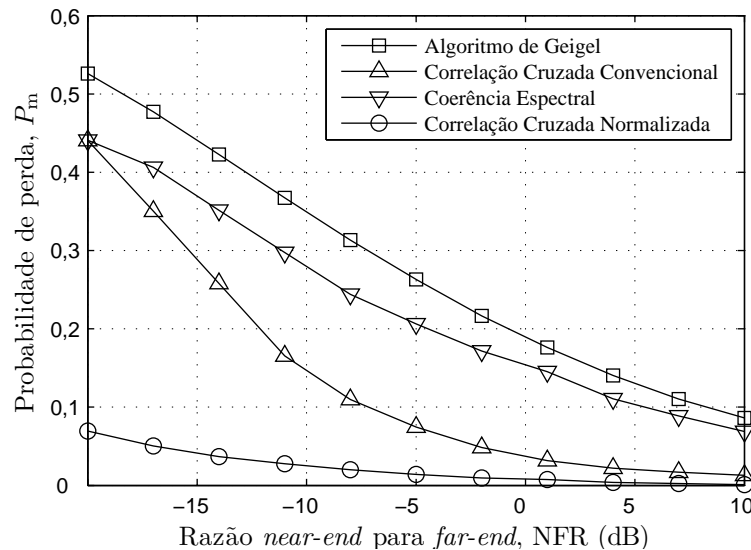


Figura 4.3: Comparação dos valores de P_m obtidos por diferentes técnicas, considerando uma probabilidade de falso alarme $P_f = 0,1$.

Observa-se que o método do NCC apresenta melhor desempenho para todos os valores de NFR considerados. O método da correlação cruzada convencional obtém resultados próximos aos obtidos pelo NCC apenas para valores elevados de NFR (aproximadamente, maior do que 1 dB), enquanto o método da coerência espectral e o algoritmo de Geigel apresentam, de maneira geral, os piores desempenhos. Na Tabela 4.1, são mostrados os valores de T utilizados nos experimentos para cada uma das técnicas. Verifica-se que o valor de T obtido experimentalmente para o NCC é muito próximo de 1, que é o limiar teórico ótimo para esse método.

Tabela 4.1: Valor do Limiar T Obtido para Diferentes Técnicas

Método	T
Algoritmo de Geigel	19,520
Correlação Cruzada	0,661
Coerência Espectral	0,274
Correlação Cruzada Normalizada	0,996

4.7 Considerações

Neste capítulo, foram discutidas algumas técnicas disponíveis na literatura para a detecção de *double-talk* em sistemas de cancelamento de eco. A mais simples delas é o

algoritmo de Geigel, que considera que existe uma atenuação fixa entre o sinal do usuário no *far-end* e o seu eco. Uma desvantagem desse algoritmo é que nem sempre se conhece *a priori* esse valor de atenuação. Já nos métodos da correlação cruzada e da coerência espectral, considera-se que a correlação entre os sinais $x(n)$ e $s(n)$ é baixa quando $s(n)$ é contaminado pelo sinal de voz do usuário do *near-end*, indicando assim a presença de *double-talk*. No entanto, nesses dois métodos, não existe um limiar de detecção que seja independente das características dos sinais e dos sistemas. O método da coerência espectral tem ainda como desvantagem uma elevada complexidade computacional.

O método da correlação cruzada normalizada (NCC) tem como grande diferencial apresentar um limiar de detecção que não depende das características do sistema. No entanto, a implementação direta desse método requer a inversão de uma matriz $N \times N$ para todo n , que pode acarretar elevado custo computacional para valores elevados de N . O cálculo da inversa dessa matriz pode ser evitado considerando algumas simplificações, que, no entanto, tornam a variável de detecção dependente do valor dos coeficientes do filtro adaptativo do cancelador de eco.

5 Implementação Eficiente do Método da Correlação Cruzada Normalizada

Nesse capítulo, é discutida uma estratégia eficiente para a implementação do método da correlação cruzada. A principal vantagem dessa estratégia é depender apenas do conhecimento dos sinais do *far-end* e do *near-end*, não levando em conta outras variáveis do sistema. Na abordagem proposta, uma versão de baixa complexidade do algoritmo RLS é utilizada para efetuar uma parte importante dos cálculos, eliminando assim a necessidade de algumas aproximações usuais.

5.1 Abordagem Considerada

Conforme discutido no Capítulo 4, a principal vantagem do método da correlação cruzada normalizada (NCC) para a detecção de *double-talk* é que se pode determinar um limiar de detecção ótimo não dependente das características do sistema e dos sinais envolvidos. Para obter a variável de detecção do NCC, é necessário determinar a inversa da matriz de autocorrelação do sinal de entrada, o que pode demandar elevada carga computacional. Existem disponíveis na literatura algumas formas simplificadas do NCC que evitam o cálculo da inversa de $\mathbf{R}_x(n)$. Nessas estratégias, no entanto, existe a dependência do vetor de coeficientes do filtro adaptativo $\hat{\mathbf{h}}(n)$. Tal dependência pode, em aplicações práticas, gerar uma série de problemas, principalmente na fase inicial de operação do cancelador de eco e nas situações em que existe uma modificação no caminho do eco. Dessa forma, é desejável obter uma maneira eficiente de calcular a variável de detecção do NCC diretamente a partir de (4.31), sem a dependência de $\hat{\mathbf{h}}(n)$. Um possível caminho é obter, de maneira recursiva, diretamente a matriz $\mathbf{R}_x^{-1}(n)$, evitando assim a inversão explícita de $\mathbf{R}_x(n)$. Para tal, como visto no Capítulo 3, pode ser utilizada a atualização recursiva de $\mathbf{R}_x^{-1}(n)$ do algoritmo RLS, através de (3.37). Ainda assim, tal procedimento requer uma complexidade computacional elevada, pois a implementação de (3.37) e (4.31) demanda $\mathcal{O}(N^2)$ multiplicações.

Uma alternativa mais interessante para determinar $\xi_{\text{NCC}}(n)$ pode ser obtida considerando o vetor de coeficientes $\hat{\mathbf{h}}_{\text{LS}}(n)$ obtido pelo algoritmo RLS. Considerando os sinais de entrada $x(n)$ e $s(n)$, o RLS atualiza de maneira recursiva o vetor de coeficientes que obedece à equação

$$\hat{\mathbf{h}}_{\text{LS}}(n) = \mathbf{R}_{\mathbf{x}}^{-1}(n) \mathbf{r}_{\mathbf{x}s}(n). \quad (5.1)$$

A relação (5.1) se mantém não importando a natureza dos sinais $x(n)$ e $s(n)$. Dessa forma, tanto nas situações em que $s(n)$ é composto apenas pelo eco quanto nas situações em que existe *double-talk*, o vetor de coeficientes $\hat{\mathbf{h}}_{\text{LS}}(n)$ é exatamente o produto de $\mathbf{R}_{\mathbf{x}}^{-1}(n)$ por $\mathbf{r}_{\mathbf{x}s}(n)$. Esse mesmo resultado é utilizado no cálculo da variável de detecção do NCC, que é então escrita como

$$\xi_{\text{NCC}}(n) = \frac{\mathbf{r}_{\mathbf{x}s}^{\text{T}}(n) \hat{\mathbf{h}}_{\text{LS}}(n)}{\sigma_y^2(n)}. \quad (5.2)$$

A utilização de (5.2) para o cálculo de $\xi_{\text{NCC}}(n)$ é interessante do ponto de vista prático, pois versões de complexidade reduzida do RLS podem ser utilizadas como um método eficiente para o cálculo do vetor $\hat{\mathbf{h}}_{\text{LS}}(n)$. Dessa forma, elimina-se a necessidade do cálculo explícito da matriz $\mathbf{R}_{\mathbf{x}}^{-1}(n)$.

É importante observar que aqui, os coeficientes determinados pelo algoritmo RLS não são interpretados como aqueles de um filtro adaptativo em uma aplicação convencional de cancelamento de eco. Dado que tais coeficientes são adaptados inclusive durante os instantes de *double-talk*, não há garantia de que $\hat{\mathbf{h}}_{\text{LS}}(n)$ estime adequadamente o caminho do eco.

Assim, o algoritmo proposto para detecção de *double-talk* pode ser implementado seguindo os passos:

1. Cálculo de $\hat{\mathbf{h}}_{\text{RLS}}(n)$ utilizando uma versão de baixa complexidade do algoritmo RLS.
2. Atualização de $\mathbf{r}_{\mathbf{x}y}(n)$ e $\sigma_y^2(n)$ usando (4.4) e (4.6).
3. Cálculo de $\xi_{\text{NCC}}(n)$ através de (5.2).

Um possível caminho para a implementação do NCC é utilizar o algoritmo SFTF para o cálculo de $\hat{\mathbf{h}}_{\text{RLS}}(n)$. Esse algoritmo possui baixa complexidade computacional, em relação ao RLS convencional, e suas características numéricas são melhores do que as do FTF. No entanto, o algoritmo SFTF pode ainda apresentar instabilidade numérica caso o valor de λ seja muito pequeno [22]. Em [22], é mostrado que um comportamento numérico satisfatório é obtido se

$$1 \geq \lambda \geq \lambda_{\text{MIN}} \quad (5.3)$$

com

$$\lambda_{\text{MIN}} = 1 - \frac{0,4}{N}. \quad (5.4)$$

Para $N = 256$, tem-se $\lambda_{\text{MIN}} \approx 0,9985$. No entanto, observa-se pela Figura 4.2 que a faixa de valores para λ definida em (5.3) não corresponde à de melhor desempenho. A abordagem aqui escolhida para obter estabilidade numérica sem comprometer o desempenho é utilizar, para o cálculo de $\hat{\mathbf{h}}_{\text{LS}}(n)$, um fator de esquecimento, designado como λ_{RLS} , diferente daquele denotado por λ , usado em (4.4) e (4.6). Os valores considerados nos experimentos são $\lambda_{\text{RLS}} = 0,9999$ e $\lambda = 0,995$.

Na Figura 5.1, é mostrada uma comparação de desempenho entre a implementação do NCC considerando o algoritmo SFTF e a implementação considerando o cálculo explícito de $\mathbf{R}_{\mathbf{x}}^{-1}(n)$ (denominada implementação convencional). A metodologia aqui utilizada é a mesma considerada na Seção 4.6. Verifica-se que o uso de λ_{RLS} no cálculo de $\hat{\mathbf{h}}_{\text{LS}}(n)$ através do algoritmo SFTF não traz grande impacto ao desempenho do método.

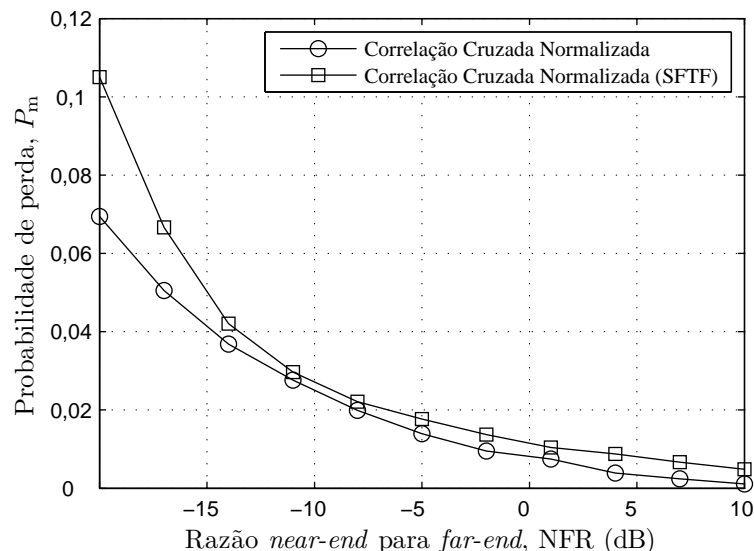


Figura 5.1: Comparação dos valores de P_m obtidos pela implementação proposta do NCC, com $\lambda_{\text{RLS}} = 0,9999$ e $\lambda = 0,995$, e o algoritmo convencional, considerando uma probabilidade de falso alarme $P_f = 0,1$.

5.2 Implementação do NCC Utilizando o Algoritmo FNTF

Uma alternativa para reduzir a complexidade computacional da implementação proposta é utilizar o algoritmo FNTF para o cálculo de $\hat{\mathbf{h}}_{\text{LS}}(n)$. Nesse caso, um parâmetro que influencia o desempenho do método é a ordem N_p dos preditores. Na Figura 5.2,

é mostrada a curva da probabilidade de perda P_m em função de N_p , considerando um experimento com um par de arquivos de *far-end* e *near-end*. Observa-se que a redução da ordem dos preditores não prejudica substancialmente o desempenho do método. Mais ainda, o melhor caso corresponde a $N_p = 0$. Essa característica é também observada considerando outros valores de NFR e P_f . Assim, esse valor de N_p é utilizado nos experimentos que seguem.

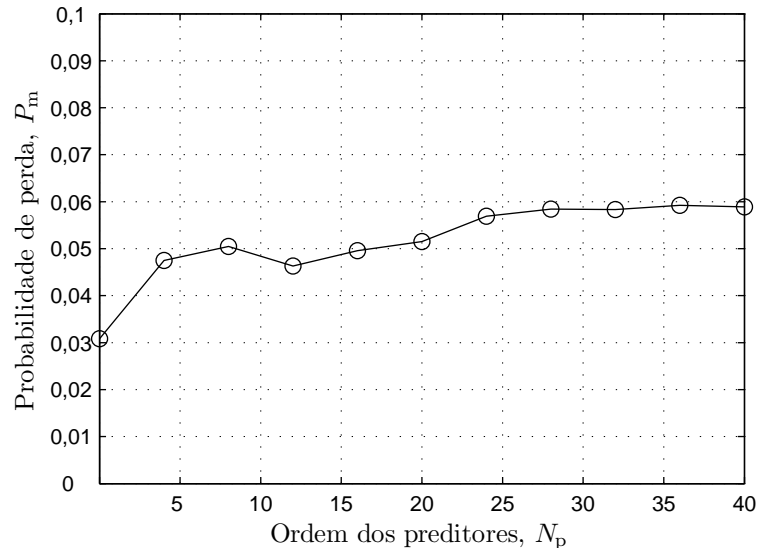


Figura 5.2: Influência da ordem dos preditores N_p , considerando NFR = -5 dB e $P_f = 0, 1$.

Na Figura 5.3, é mostrada uma comparação da implementação proposta, utilizando o algoritmo FNFF com $N_p = 0$, com a implementação convencional. Verifica-se que o desempenho nos dois casos é bastante próximo. Já na Tabela 5.1, é mostrada a complexidade computacional de cada uma das etapas da implementação do NCC proposta, considerando o algoritmo FNTF com $N_p = 0$. Verifica-se que a complexidade computacional total da implementação proposta é $5N$.

Tabela 5.1: Complexidade Computacional da Implementação Proposta

Etapas	Complexidade
Cálculo de $\hat{\mathbf{h}}_{\text{RLS}}(n)$	$2N$
Atualização recursiva de $\mathbf{r}_{xy}(n)$ e $\sigma_y^2(n)$	$2N$
Cálculo de $\xi_{\text{NCC}}(n)$	N
TOTAL	$5N$

É possível reduzir a complexidade do NCC se forem assumidas algumas suposições sobre os sinais envolvidos. Em particular, considerando que sinais de voz têm a maior parte de sua energia concentrada em baixas frequências, pode-se (em determinados casos)

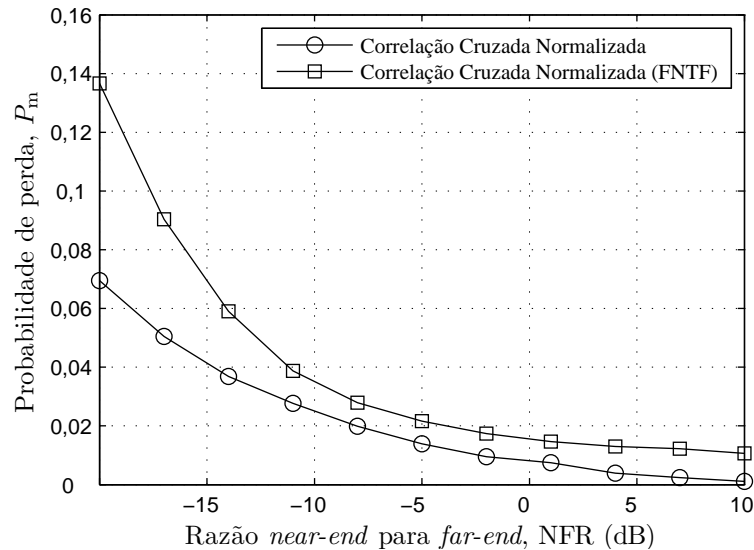


Figura 5.3: Comparação dos valores de P_m obtidos pelo NCC considerando o algoritmo FNMF, com $N_p = 0$, em relação ao NCC convencional.

realizar a detecção de *double-talk* usando uma taxa de amostragem menor do que a taxa do sistema, sem grande prejuízo de desempenho. Assim, a dimensão N dos vetores utilizados na detecção de *double-talk* reduz-se para

$$N_r = \frac{N}{L} \quad (5.5)$$

onde L caracteriza o fator de redução da taxa de amostragem. Considerando ainda que o número total de operações é também reduzido por um fator L (devido à redução na taxa de amostragem), a complexidade total do método torna-se

$$C_r = \frac{5N}{L^2} \quad (5.6)$$

Na Figura 5.4, é mostrada uma comparação do desempenho do NCC baseado no FNMF com diferentes fatores de redução de taxa de amostragem do sistema. Observa-se que até o valor da frequência de amostragem $f_s = 2$ kHz (que corresponde ao fator de redução $L = 4$), não há perda significativa de desempenho.

5.3 Considerações

Neste capítulo, foi apresentada uma implementação eficiente do método da correlação cruzada normalizada (NCC) para detecção de *double-talk* que não depende dos coeficientes do filtro de cancelamento de eco. Na implementação proposta, uma parte importante dos cálculos é efetuada através do algoritmo RLS. Assim, utilizando o al-

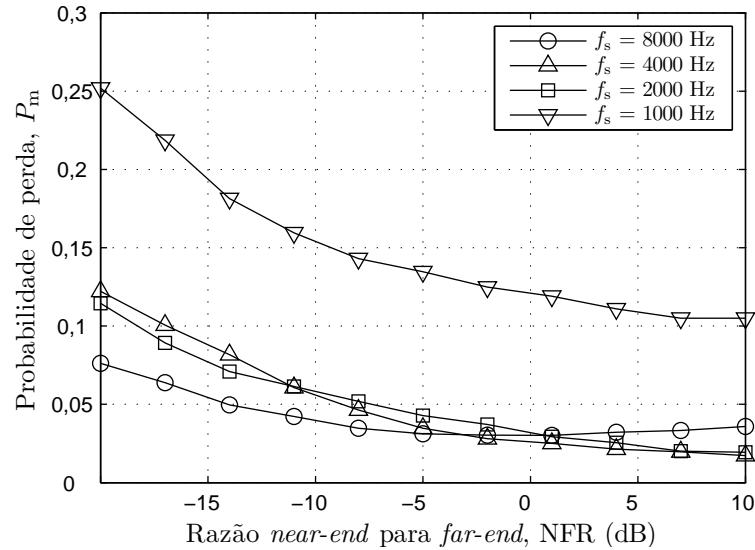


Figura 5.4: Probabilidade de perda P_m para diferentes frequências de amostragem.

goritmo FNTF (uma versão de baixa complexidade do RLS), com preditores de ordem zero, o NCC pode ser implementado com complexidade de $5N$. Pode-se adicionalmente reduzir a complexidade computacional se a detecção de *double-talk* for realizada usando uma taxa de amostragem inferior à do sistema. Resultados de simulação mostraram que a implementação proposta apresenta um desempenho equivalente ao da implementação convencional do método da correlação cruzada normalizada envolvendo a inversão de uma matriz $N \times N$. Além do mais, observa-se que a detecção de *double-talk* pode ser realizada usando uma taxa de amostragem tão baixa quanto $f_s = 2$ kHz, sem prejuízo significativo ao desempenho. Nessa situação, a complexidade da abordagem proposta pode ser tão baixa quanto $0,3N$.

6 Considerações Finais

Neste trabalho, foi apresentada uma abordagem eficiente para a implementação do método da correlação cruzada normalizada (NCC) para detecção de *double-talk* em sistemas de cancelamento de eco. Neste capítulo, os principais pontos do trabalho são discutidos e algumas propostas para trabalhos futuros são sugeridas.

6.1 Sumário e Discussão dos Resultados

No Capítulo 2, é apresentada uma introdução geral aos sistemas de cancelamento de eco. São discutidos os tipos de eco mais relevantes em telefonia, que são o eco de linha e o eco acústico. A abordagem usualmente considerada para solucionar o problema do eco é a utilização de um cancelador de eco. Nesses sistemas, uma réplica do sinal de eco é subtraída do sinal de voz contaminado pelo eco. Dessa forma, não há a interrupção do caminho da voz dos usuários, preservando assim a qualidade da conversação. Em geral, a resposta ao impulso do caminho do eco é estimada utilizando um algoritmo de filtragem adaptativa. O algoritmo adaptativo tem como objetivo obter de forma iterativa uma estimativa da resposta ao impulso do caminho do eco, visando gradualmente reduzir a diferença entre o sinal de eco e a sua réplica. No entanto, a adaptação do filtro deve ser interrompida nos instantes em que há fala nos dois lados da linha (*double-talk*). Dessa forma, é indispensável o uso de um detector de *double-talk* associado aos sistemas de cancelamento de eco.

No Capítulo 3, são apresentados alguns algoritmos de filtragem adaptativa usualmente considerados em aplicações práticas. O algoritmo mais popular é o LMS, cujas principais características são a sua robustez, simplicidade e baixa complexidade computacional. No entanto, em determinadas situações, o LMS pode exibir baixa velocidade de convergência. Outro algoritmo bastante utilizado em aplicações práticas é o RLS. Em relação ao LMS, o RLS apresenta velocidade de convergência substancialmente maior. A desvantagem desse algoritmo, porém, é a sua elevada complexidade computacional. Existem, no entanto, algumas estratégias de implementação do RLS que apresentam menor

complexidade. Uma delas é o algoritmo FTF, que tira proveito da estrutura transversal do filtro adaptativo. No entanto, o FTF é reconhecido ser numericamente instável, quando implementado com precisão finita. Uma modificação do FTF, denominada SFTF, visa melhorar as suas características numéricas com um pequeno aumento de complexidade. Outra modificação do FTF é o algoritmo FNFTF. Nesse algoritmo, considera-se que o sinal de entrada pode ser representado por um processo auto-regressivo $AR(L)$. Assim, ajustando o valor de L de zero até N , obtém-se um compromisso entre a baixa complexidade do LMS e o desempenho superior do FTF.

No Capítulo 4, são mostradas algumas técnicas disponíveis na literatura para a detecção de *double-talk*. A mais simples delas é o algoritmo de Geigel, que é baseado na comparação de nível entre os sinais do *far-end* e do *near-end*. No entanto, nesse algoritmo, deve-se conhecer *a priori* o valor da atenuação entre o sinal do *far-end* e o seu eco, o que nem sempre é possível na prática. Nos métodos da correlação cruzada e da coerência espectral, considera-se que a correlação entre os sinais do *far-end* e do *near-end* é baixa quando existe *double-talk*. Uma desvantagem desses métodos é que não existe um limiar de detecção que seja independente das características do sistema e dos sinais envolvidos. O método da correlação cruzada normalizada (NCC) tem como principal vantagem apresentar um limiar de detecção que não depende dessas características. A desvantagem desse método é que a sua implementação envolve a inversão de uma matriz $N \times N$, o que pode acarretar elevado custo computacional. O cálculo dessa matriz inversa pode ser evitado considerando algumas simplificações, que fazem com que o cálculo da variável de detecção passe a depender dos coeficientes do filtro adaptativo do cancelador de eco. Resultados de simulação mostram o desempenho superior do método da correlação cruzada normalizada em relação aos outros métodos apresentados.

No Capítulo 5, é descrita a implementação proposta do NCC. Nessa implementação, não há a dependência do filtro adaptativo utilizado no cancelamento de eco. A parcela mais significativa do cálculo da variável de detecção do NCC pode ser realizada através de uma versão de baixa complexidade do algoritmo RLS. Utilizando o algoritmo FNFTF e considerando $L = 0$, o NCC pode ser implementado com complexidade de $5N$. Pode-se obter uma redução ainda maior da complexidade se a detecção de *double-talk* for realizada utilizando uma taxa de amostragem inferior à do sistema. Resultados de simulação mostram que o desempenho da implementação proposta é equivalente ao obtido pela implementação direta do NCC (que envolve inversão de uma matriz $N \times N$). Verifica-se ainda que a detecção de *double-talk* pode ser realizada considerando uma taxa de amostragem tão baixa quanto $f_s = 2$ kHz, sem perda significativa de desempenho. Dessa forma, a complexidade da abordagem proposta pode ser tão baixa quanto $0,3N$.

6.2 Propostas para Trabalhos Futuros

Na abordagem considerada neste trabalho para a implementação do NCC, uma parte significativa dos cálculos é realizada pelo algoritmo RLS. Aqui, foi considerado o uso do algoritmo FNTF. No entanto, outras versões do RLS podem ser consideradas. Em particular, podem ser explorados algoritmos que trabalhem em blocos, como o *fast subsampled-updating* SFSTF [25]. Para valores elevados de N , esse algoritmo pode apresentar complexidade computacional menor até do que a do algoritmo LMS. Podem também ser consideradas implementações do NCC diretamente no domínio da frequência, o que poderia reduzir ainda mais a complexidade computacional do método.

Referências

- [1] M. M. Sondhi, “The history of echo cancellation,” *Signal Processing Magazine, IEEE*, vol. 23, no. 5, pp. 95–102, Sept. 2006.
- [2] E. Hänsler and G. Schmidt, *Topics in Acoustic Echo and Noise Control: Selected Methods for the Cancellation of Acoustical Echoes, the Reduction of Background Noise, and Speech Processing (Signals and Communication Technology)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2006.
- [3] S. Haykin, *Adaptive filter theory*, 3rd ed. New Jersey, USA: Prentice Hall, 2000.
- [4] D. L. Duttweiler, “Proportionate normalized least-mean-squares adaptation in echo cancelers,” *IEEE Transactions on Speech and Audio Processing*, vol. 8, no. 5, pp. 508–518, Sep. 2000.
- [5] J. Cioffi and T. Kailath, “Fast recursive least squares transversal filters for adaptive filtering,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 32, no. 2, pp. 304–337, Apr. 1984.
- [6] G. Moustakides and S. Theodoridis, “Fast newton transversal filters—a new class of adaptive estimation algorithms,” *IEEE Transactions on Signal Processing*, vol. 39, no. 10, pp. 2184–2193, Oct. 1991.
- [7] D. L. Duttweiler, “A twelve-channel digital echo canceler,” *IEEE Transactions on Communications*, vol. 26, no. 5, pp. 647–653, 1978.
- [8] H. Ye and B.-X. Wu, “A new double-talk detection algorithm based on the orthogonality theorem,” *IEEE Transactions on Communications*, vol. 39, no. 11, pp. 1542–1545, Nov. 1991.
- [9] T. Gansler, M. Hansson, C.-J. Ivarsson, and G. Salomonsson, “A double-talk detector based on coherence,” *IEEE Transactions on Communications*, vol. 44, no. 11, pp. 1421–1427, Nov. 1996.
- [10] J. H. Cho, D. R. Morgan, and J. Benesty, “An objective technique for evaluating doubletalk detectors in acoustic echo cancelers,” *IEEE Transactions on Speech and Audio Processing*, vol. 7, pp. 718–724, Nov. 1999.
- [11] K. Ochiai, T. Araseki, and T. Ogihara, “Echo canceler with two echo path models,” *IEEE Transactions on Communications*, vol. 25, no. 6, pp. 589–595, Jun 1977.
- [12] J. Benesty, D. R. Morgan, and J. H. Cho, “A new class of doubletalk detectors based on cross-correlation,” *IEEE Transactions on Speech and Audio Processing*, vol. 8, no. 2, pp. 168–172, Mar. 2000.

- [13] M. Waseem, M. Omer, A. Bhatti, and S. Shah, "A new doubletalk detector with fuzzy decision thresholding," in *Proc. IEEE Students Conf.*, vol. 1, Lahore, Pakistan, Aug. 2002, pp. 117–121.
- [14] J. H. James, B. Chen, and L. Garrison, "Implementing VoIP: A voice transmission performance progress report," *IEEE Communications Magazine*, vol. 42, no. 7, pp. 36–41, Jul. 2004.
- [15] *Transmission characteristics and speech quality parameters of hands-free terminals*, Document ITU-T Rec. P.340, International Telecommunication Union (ITU), Geneva, Switzerland, 2000.
- [16] *Digital network echo cancellers*, Document ITU-T Rec. G.168, International Telecommunication Union (ITU), Geneva, Switzerland, 2002.
- [17] L. S. Gay and J. Benesty, *Acoustic Signal Processing for Telecommunication*, 1st ed. Norwell, USA: Kluwer Academic Publishers, 2000.
- [18] B. Widrow and S. D. Stearns, *Adaptive Signal Processing*. New Jersey, USA: Prentice-Hall, 1985.
- [19] A. V. Oppenheim and A. S. Willsky, *Signals & Systems*, 1st ed. New Jersey, USA: Prentice Hall, 1997.
- [20] B. Farhang-Boroujeny, *Adaptive filters: theory and applications*, 1st ed. New York, USA: John Wiley & Sons, 2000.
- [21] G.-O. Glentis, K. Berberidis, and S. Theodoridis, "Efficient least squares adaptive algorithms for FIR transversal filtering," *Signal Processing Magazine, IEEE*, vol. 16, no. 4, pp. 13–41, Jul 1999.
- [22] D. Slock and T. Kailath, "Numerically stable fast transversal filters for recursive least squares adaptive filtering," *IEEE Transactions on Signal Processing*, vol. 39, no. 1, pp. 92–114, Jan. 1991.
- [23] T. Petillon, A. Gilloire, and S. Theodoridis, "The fast newton transversal filter: an efficient scheme for acoustic echo cancellation in mobile radio," *Signal Processing, IEEE Transactions on*, vol. 42, no. 3, pp. 509–518, Mar 1994.
- [24] D. Thomson, "An overview of multiple-window and quadratic-inverse spectrum estimation methods," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. VI, Apr. 1994, pp. 185 – 194.
- [25] K. Maouche and D. Slock, "Fast subsampled-updating stabilized fast transversal filter (FSU SFTF) RLS algorithm for adaptive filtering," *IEEE Transactions on Signal Processing*, vol. 48, no. 8, pp. 2248–2257, Aug. 2000.