Universidade Estadual de Campinas
FACULDADE DE ENGENHARIA ELÉTRICA E DE COMPUTAÇÃO - FEEC
DEPARTAMENTO DE COMUNICAÇÕES - DECOM

Predição de Tráfego Auto-Similar em Redes de Faixa Larga

Marcelo Menezes de Carvalho

Orientador :

Dalton Soares Arantes

Dissertação submetida à Faculdade de Engenharia Elétrica e de Computação - FEEC da Universidade Estadual de Campinas - Unicamp, como parte dos requisitos exigidos para a obtenção do título de Mestre em Engenharia Elétrica.

7 de Julho de 1998.

FICHA CATALOGRÁFICA ELABORADA PELA
BIBLIOTECA DA ÁREA DE ENGENHARIA - BAE - UNICAMP

C253p

Carvalho, Marcelo Menezes de
Predição de tráfego auto-similar em redes de faixa larga. / Marcelo Menezes de Carvalho.--Campinas, SP: [s.n.], 1998.

Orientador: Dalton Soares Arantes.
Dissertação (mestrado) - Universidade Estadual de Campinas, Faculdade de Engenharia Elétrica e de Computação.

1. Fractais. 2. Redes neurais (Computação) 3. Redes locais de computação. 4. Estimativa de parâmetros. 5. Previsão estatística. I. Arantes, Dalton Soares. II. Universidade Estadual de Campinas. Faculdade de Engenharia Elétrica e de Computação. III. Título.

Um passo à frente e você não
está mais no mesmo lugar.

Chico Science

Aos meus pais **José Maria** e **Maria Neusa**.
À minha esposa **Mylène**.

Agradecimentos

Gostaria de expressar a minha eterna gratidão ao **Prof. Dr. Dalton Soares Arantes**, cujo apoio incondicional a mim dispensado foi essencial para a execução deste trabalho. A sua competência, dedicação e impressionante motivação tornaram-se referências para mim, em cuja pessoa pude encontrar não só um grande pesquisador e professor, mas também alguém capaz de semear, nas pessoas que o cercam, os sentimentos da ética, do companheirismo e da cordialidade. Mais do que um orientador, pude conquistar uma grande amizade.

Gostaria de agradecer aos colegas engenheiros **Albanita G. de Oliveira**, **Gustavo Hirchoren**, **Fabbryccio Akkazzha**, **Fernando de Castro** e **Cristina de Castro** as diversas e ricas discussões do nosso grupo.

Aos funcionários do DECOM e da FEEC, especialmente a **Mário Niimi**, por seu suporte no Laboratório de Comunicações, e a **Noêmia**, **Lúcia** e **Celi**, pela paciência e apoio.

Meus sinceros agradecimentos à **Fundação Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - CAPES**, pelo suporte financeiro ao longo deste trabalho.

Ao **Centro de Pesquisas e Desenvolvimento da Telebrás - CPqD/Telebrás**, pela oportunidade de participar do convênio Unicamp/Telebrás.

Aos meus pais **José Maria** e **Maria Neusa**, os grandes responsáveis por eu estar hoje escrevendo essas linhas. Além de todo amor, carinho e compreensão, deram-me o que eles podiam oferecer de melhor : a educação. Aos meus queridos irmãos **Eliane**, **Ana** e **Evandro** agradeço todo apoio e incentivo.

À minha esposa **Mylène**, minha principal fonte de inspiração e companheira de todas as horas, agradeço não só a ajuda prestada na redação deste trabalho, mas, principalmente, o seu amor, paciência e constante incentivo, sem os quais eu jamais poderia superar os momentos mais difíceis desta longa jornada. A ela, o meu eterno amor.

A Deus.

Marcelo Menezes de Carvalho
Campinas, Julho de 1998.

Resumo

Este trabalho tem como objetivo estudar os modelos e a predição de tráfego com características auto-similares (fractais) em redes de comunicações de faixa larga, como por exemplo, a rede ATM (Asynchronous Transfer Mode). Para isto, são estudados os principais modelos de processos estocásticos auto-similares, bem como os métodos utilizados para estimação do grau de auto-similaridade de uma série temporal. São também estudadas as principais conseqüências do fenômeno da auto-similaridade no problema da predição, estimação e controle do tráfego em redes de alta velocidade. Para fins de predição, avalia-se o uso de preditores lineares (filtros FIR) e não-lineares, estes últimos representados pelas redes neurais do tipo perceptron multicamadas FIR e Redes de Funções de Base Radiais (Radial Basis Function). A eficácia dos preditores estudados é analisada através da predição de tráfego real de redes locais Ethernet. Além disso, propõe-se um algoritmo adaptativo, baseado no algoritmo EM (Expectation-Maximization), para estimação dos parâmetros de uma mistura de densidades Gaussianas.

Abstract

The purpose of this work is to study the modeling and prediction of self-similar traffic signals in broadband telecommunications networks (ATM networks, for example). With this in mind, we study the main models of self-similar random processes, as well as the estimation methods for the degree of self-similarity of time series. We also study the consequences of statistic self-similarity in problems like prediction, estimation and traffic control in high-speed networks. For the particular case of prediction, we investigate the use of linear FIR filters and non-linear predictors, the latter being represented by FIR multilayer perceptron and by Radial Basis Function (RBF) neural networks. The performance of predictors is evaluated with real LAN Ethernet traffic. Finally, we propose an adaptive algorithm, based on the EM algorithm (Expectation-Maximization), for real time estimation of the parameters of a mixture of Gaussian densities.

Conteúdo

Lista de Figuras

Lista de Tabelas

Capítulo 1

Introdução

Recentemente, pesquisadores dos laboratórios Bellcore publicaram o que hoje é considerada uma das principais descobertas na área de redes de computadores. Baseados em uma análise estatística rigorosa de medições de tráfego de redes locais Ethernet, eles demonstraram que este tipo de tráfego é estatisticamente *auto-similar* e que nenhum dos modelos de tráfego comumente utilizados é capaz de capturar este comportamento *fractal* [1].

Tradicionalmente, a modelagem de tráfego em telefonia convencional e em redes de computadores tem se baseado, quase que exclusivamente, nos modelos Poissonianos ou, mais genericamente, Markovianos, como por exemplo, o modelo de Poisson puro e os modelos de Poisson Modulados por Markov. Utilizam-se também os modelos de fluxos de fluidos e os modelos de trens de pacotes [1, 3, 4]. Nestes modelos, está implícita a hipótese de que o tráfego possui incrementos *independentes* ou *fracamente correlacionados*[1]. No entanto, as análises das medições de tráfego de redes locais Ethernet constataram a ocorrência do fenômeno de *dependência de longo prazo*, segundo o qual o processo estocástico envolvido possui uma função de autocovariância não somável. Na prática, esta característica é expressa por funções de autocorrelação com decaimento *hiperbólico*, ao contrário de um decaimento *exponencial*, típico dos modelos de tráfego Poissonianos. Além disso, foi constatada a presença de surtos de dados em diversas escalas de tempo, de forma a sugerir uma ausência de um comprimento natural para os surtos de dados. Ao se variar as escalas de tempo de milisegundos a minutos e horas, os surtos consistem sempre de subperíodos de surtos intercalados por outros subperíodos com menos surtos. Juntos, a dependência de longo prazo e a invariância a escalas de tempo consistem em comportamentos típicos de um *processo estocástico auto-similar*, conceito matemático introduzido originalmente por Mandelbrot [5, 6].

Após o anúncio desta constatação, diversos estudos passaram a ser realizados em vários outros tipos de tráfego. Verificou-se que, além do tráfego de redes locais Ethernet, outros tipos de tráfego apresentam esta mesma característica, como o tráfego de redes WAN ("Wide Area Networks") [7], tráfego de vídeo com taxa de bit variável (VBR) [8] e tráfego Internet [9].

A descoberta da auto-similaridade do tráfego de redes locais Ethernet, bem como dos outros tipos de tráfego citados, tem conseqüências não só na modelagem de tráfego, mas também (e principalmente) no projeto, na análise e no controle de redes de alta velocidade baseadas em tráfego de pacotes. Em particular, espera-se que este fenômeno seja decisivo

[1]Podemos definir, basicamente, dois tipos de processos quando trabalhamos com tráfego de redes de computadores : o processo de *contagem* e o processo de *incrementos*. Por processo de contagem entende-se o processo que expressa o acúmulo de dados transmitidos em um dado intervalo de tempo. Por processo de incrementos entende-se a taxa com que ocorre este acúmulo de tráfego (bit/s, pacote/s, etc.)

no comportamento do tráfego *agregado* das emergentes redes ATM, as quais transportarão diversos tipos de tráfego, com diferentes características de taxa de bits e qualidade de serviço.

O modo de transferência conhecido por ATM ("Asynchronous Transfer Mode") consiste em um forte candidato para implementação da futura Rede Digital de Serviços Integrados de Faixa Larga (RDSI-FL) e baseia-se na comutação rápida de pacotes de comprimento fixo, denominados de "células", através da multiplexagem por divisão de tempo assíncrona. Neste modo de transferência, a alocação de banda é feita segundo uma multiplexagem *estatística* dos diversos tipos de tráfego, sendo esta realizada nos *multiplexadores estatísticos* situados nos *comutadores de células ATM*. É a multiplexagem estatística a grande responsável pela agregação dos diversos tipos de tráfego dentro da rede, independentemente de suas características de largura de banda ou qualidade de serviço requerida.

Ainda que os enlaces das redes ATM permitam velocidades altíssimas de transmissão e garantam baixíssimas probabilidades de erro (no que se refere a erros nos bits transmitidos), existe na rede ATM a possibilidade de transbordamento de células nos diversos multiplexadores estatísticos espalhados pela rede. Quando isto ocorre, há perda de células com a conseqüente degradação dos serviços de usuário.

Este transbordamento, no entanto, seria em princípio facilmente evitado caso a hipótese de tráfego com incrementos independentes fosse válida. Pois, sob esta hipótese, é possível se demonstrar uma *uniformização* do tráfego à medida que se intensifica a sua agregação. Daí a importância dos multiplexadores estatísticos para a realização desta agregação. Nestas condições, é possível se demonstrar ainda que a probabilidade de transbordamento dos multiplexadores estatísticos ("buffers") decresce *exponencialmente* com o tamanho d do "buffer"[2], desde que o valor médio do tráfego seja menor do que a capacidade de transmissão C do enlace à saída do multiplexador estatístico. Além disso, o atraso médio de fila tende a um valor constante à medida que o tamanho d tende ao infinito [9]. Na prática, este decaimento exponencial significa a possibilidade de obtenção de probabilidades de transbordamento desprezíveis com o uso de memórias relativamente pequenas.

Porém, caso o tráfego possua dependência de longo prazo, o transbordamento dos multiplexadores estatísticos pode se tornar muito freqüente, principalmente quando se deseja trabalhar com alta eficiência (próximo da capacidade C de transmissão). Foi demonstrado recentemente que, para processos auto-similares, a probabilidade de transbordamento decai apenas *algebricamente* (ou seja, polinomialmente) com o tamanho d do "buffer". Além disso, o atraso médio de fila cresce indefinidamente com d [10].

Uma outra conseqüência da auto-similaridade do tráfego é a dificuldade de estimação confiável dos parâmetros do tráfego (médias, variâncias, etc.). Dependendo do valor do parâmetro de auto-similaridade H, a estimação dos parâmetros torna-se um problema totalmente diferente quando comparado às estimações de parâmetros de tráfego convencional. Como será visto adiante, a estimação confiável da taxa média de tráfego num período de cinco minutos é bastante diferente da estimação confiável num período de cinco horas (para valores altos de H) [3].

Tendo em vista os problemas que surgirão em função da auto-similaridade dos tributários que irão compor o tráfego das redes ATM, uma possível solução seria a busca do conhecimento antecipado do comportamento do tráfego, ou seja, a sua predição. De posse dessa informação, e tendo como base o estado da rede no momento da predição, poderia ser possível uma alocação de recursos mais racional, de forma a reduzir problemas de congestionamento ou perda de

[2] Ao longo do texto, utilizaremos os termos "buffer" e multiplexador estatístico indistintamente.

células. Além disso, seria possível ainda se pensar em um policiamento mais eficaz do tráfego de cada usuário, permitindo um maior controle de admissão e conexão de novos usuários à rede, bem como das conexões já estabelecidas. Por esses motivos, o uso de preditores pode ser considerado como uma ferramenta essencial no projeto, controle e gerenciamento de redes de alta velocidade. Dentro desse contexto, o objetivo principal deste trabalho é, portanto, estudar a predição de tráfego auto-similar e a sua aplicação em redes de alta velocidade, em particular, nas redes ATM.

Assim sendo, descrevemos no Capítulo 2 os princípios básicos da rede ATM. Procuramos expor, de forma suscinta, os principais conceitos e elementos que formam o modo de transferência conhecido por ATM. Além disso, descrevemos também o modelo de referência do protocolo ATM e as classes de serviço vislumbradas para esta rede. O uso de preditores de tráfego pode vir a ser uma ferramenta útil para a garantia da qualidade de serviço de muitas dessas classes, como por exemplo, as classes VBR (Taxa de Bit Variável) e ABR (Taxa de Bit Disponível).

No Capítulo 3 expomos os conceitos relacionados a processos estocásticos auto-similares, definindo o que vem a ser processos estocásticos com dependência de longo prazo e auto-similaridade. Em particular, descrevemos dois dos principais modelos estocásticos que possuem características auto-similares : o *movimento Browniano fracionário* (fBm) e o *processo ARIMA fracionário* (Autoregressivo e Média Móvel Integrado). No final do capítulo, expomos o *processo de recompensas renovadas*, proposto por Mandelbrot, que tem sido estudado para explicação da origem da auto-similaridade do tráfego agregado em redes de pacotes.

No Capítulo 4 enfatizamos a auto-similaridade do tráfego Ethernet. Para isto, descrevemos os principais métodos de estimação do parâmetro de auto-similaridade H (parâmetro de Hurst) de uma série temporal. Em particular, como contribuição, implementamos e aplicamos alguns destes métodos na estimação do parâmetro de Hurst de duas séries temporais representativas de tráfego Ethernet real. Para fins de ilustração, descrevemos o tráfego Browniano fracionário, proposto por Norros [11], e que está sendo muito utilizado na análise teórica dos principais efeitos do fenômeno da auto-similaridade em aspectos relacionados a preenchimento de filas e estimação de parâmetros de tráfego. Ao final, descrevemos um modelo proposto pelos mesmos pesquisadores do Bellcore, com o intuito de explicar a natureza do tráfego gerado por fontes isoladas. Um importante resultado por eles demonstrado mostra que, partindo-se deste modelo, é possível se obter o movimento Browniano fracionário (fBm) quando o tráfego de várias fontes isoladas forem agregados. Este modelo é baseado no processo de recompensas renovadas proposto por Mandelbrot [12].

No Capítulo 5 descrevemos a predição de séries temporais sob a ótica da predição linear, representada pelo algoritmo de mínimos quadrados LMS ("Least Mean Square"), e não-linear, através do uso de redes neurais. Nesta dissertação, restringimos nosso estudo ao uso de redes multicamadas com propagação direta. Em particular, estudamos e implementamos o perceptron multicamadas FIR (resposta ao impulso finita) e as redes com funções de base radiais (RBF). No contexto de redes RBF, descrevemos um modelo de preditor dinâmico, proposto por Haykin e Yee [13], e que foi também implementado para fins de predição. Como contribuições originais, propomos uma modificação do algoritmo EM ("Expectation-Maximization") de Dempster [14] e um algoritmo adaptativo, obtido de forma sistemática a partir do primeiro, o qual denominamos de "algoritmo EM adaptativo". A modificação introduzida no algoritmo EM padrão consiste na introdução de uma função de penalidade cuja finalidade é evitar a ocorrência de singularidades quando aplicarmos o algoritmo EM na estimação de parâmetros de misturas de densidades Gaussianas, um assunto de interesse e de grande valia

na implementação de redes RBF. No caso do algoritmo EM adaptativo, espera-se que este seja aplicável quando estivermos trabalhando em ambientes não-estacionários e onde as estimações necessitem ser realizadas em tempo real, o que não pode ser feito com o algoritmo EM padrão.

No Capítulo 6 expomos os nossos resultados. Inicialmente, porém, propomos um modelo de predição a ser utilizado no contexto de redes ATM. Os resultados que se seguem correspondem à aplicação dos preditores estudados na predição de tráfego Ethernet real. Além disso, realizamos um estudo, através de simulações, do desempenho do algoritmo EM adaptativo proposto comparado ao desempenho do algoritmo EM padrão com função de penalidade.

Capítulo 2

A Rede ATM

2.1 Introdução

Em plena "Era da Informação", os consumidores de serviços de telecomunicações têm se tornado cada vez mais exigentes e seletivos com relação à qualidade, diversidade, velocidade e custo destes serviços. Orientada por essa tendência, a comunidade envolvida com as tecnologias de telecomunicações tem se esforçado na busca de novos serviços que atendam às mais variadas exigências. Serviços como vídeo-sob-demanda, vídeo-conferência, vídeo-fone, HDTV (Televisão de Alta Definição), tele-vendas, interconexão de LAN's, etc., são alguns exemplos dos principais serviços vislumbrados para um futuro bem próximo.

No entanto, tão diversificadas quanto as ofertas de serviços, as características e exigências de cada um deles os fazem totalmente distintos uns dos outros. Por esse motivo, tem havido, mundialmente, um esforço conjunto no sentido de se definir uma plataforma única, universal, flexível o bastante, onde todos os serviços (já existentes e que estejam por vir) possam ser oferecidos satisfatoriamente. Esta plataforma é a chamada **Rede Digital de Serviços Integrados de Faixa Larga (RDSI-FL)** [15, 16] definida há alguns anos pelo CCITT (the International Consultative Committee for Telecommunications and Telegraphy).

De acordo com o CCITT, atualmente denominado de ITU-T (International Telecommunications Union - Telecommunication), a RDSI-FL será uma rede de comunicações digitais baseada em um conjunto de interfaces comuns a todos os usuários (e portanto, padronizada), concebida para suportar qualquer serviço de telecomunicações. Ela será capaz de suportar conexões comutadas, semi-permanentes, permanentes, ponto-a-ponto e ponto-multiponto. Propiciará serviços sob demanda, reservados e permanentes. Será também capaz de suportar tanto os serviços com taxa de bit constante como os serviços com taxa de bit variável. Mais ainda, ela suportará transferências orientadas para conexão e transferências sem conexão ("connectionless transfers") em configurações bidirecionais e/ou unidirecionais. A RDSI-FL terá ainda dispositivos inteligentes com o propósito de fornecerem as características dos serviços e de servirem como ferramentas poderosas para a manutenção, operação, controle e gerenciamento da rede. Uma vez difundida em todo o mundo, ela facilitará a troca de informações entre quaisquer usuários, situados em qualquer lugar do planeta, de uma forma transparente e sem quaisquer limitações no que se refere às distâncias envolvidas.

Graças ao avanço vertiginoso das tecnologias de fabricação de dispositivos semicondutores e de fibras ópticas, um novo modo de transferência[1] tem sido definido para a futura RDSI-FL

[1]De acordo com o ITU-T, a palavra "modo de transferência" é empregada para descrever o conjunto de

[15, 16]. Dentre todos os modos de transferência considerados como possíveis candidatos para a RDSI-FL, o modo de transferência conhecido por ATM ("Asynchronous Transfer Mode" ou Modo de Transferência Assíncrono) foi adotado em 1987, pelo CCITT, como o modo de transferência para implementação da futura RDSI-FL. A escolha deste modo de transferência se deu, principalmente, devido à possibilidade de se transmitir através dele **qualquer** tipo de serviço, independente de suas características intrínsecas, tais como taxa de bit, exigência de qualidade ou natureza do tráfego (presença ou não de surtos). Em geral, espera-se que o modo de transferência ATM apresente as seguintes propriedades :

- Suporte a todos os serviços existentes, assim como àqueles com características ainda desconhecidas e aos que surgirem no futuro;
- Utilização dos recursos da rede da forma mais eficiente possível;
- Minimização da complexidade de comutação;
- Minimização do tempo de processamento nos nós intermediários da rede, a fim de suportarem altas velocidades de transmissão;
- Minimização do número de *buffers* nos nós intermediários da rede, com o objetivo de se limitar o atraso e a complexidade do gerenciamento dos *buffers* ;
- Garantir as exigências de desempenho das aplicações existentes e das que estejam por surgir.

2.2 Os Princípios Básicos da Rede ATM

Em 1990, o CCITT publicou uma série de Recomendações especificando os detalhes do modo de transferência ATM para uso em RDSI-FL. Estas Recomendações foram ainda ampliadas em 1991 e em 1992. Resumiremos, a seguir, as características básicas da rede ATM.

2.2.1 O Transporte de Informação

Por definição , o modo de transferência ATM é baseado na comutação *rápida* de pacotes de comprimento *fixo* através da multiplexagem por divisão de tempo *assíncrona*. O termo *assíncrona*, neste caso, refere-se à inexistência de quadros pré-fixados, repetidos sincronizadamente, para alocação dos pacotes pertencentes a uma mesma conexão (à semelhança do STM — Modo de Transferência Síncrono). Ou seja, embora no ATM exista o conceito de intervalos de tempo de duração fixa ("time slots") — os quais ocorrem de forma *síncrona* e correspondem ao tamanho de uma célula — os pacotes provenientes das diversas fontes são multiplexados de forma *assíncrona*, de forma que cada fonte pode enviar seus dados em qualquer célula que esteja vazia. Não há, portanto, a transmissão de uma unidade fixa, repetitiva (os chamados "frames"), onde a largura de banda é compartilhada de forma fixa entre as diversas conexões da rede. O que ocorre, na verdade, é uma alocação **sob demanda** da largura de banda disponível [9].

No ATM, os pacotes, por serem pequenos, são chamados de *células*. Cada célula consiste de um campo de informação de 48 bytes e um cabeçalho de 5 bytes. O cabeçalho é usado, principalmente, para identificar as células pertencentes a uma mesma conexão e para realizar

técnicas utilizadas em uma rede de telecomunicações relacionadas à transmissão, multiplexagem e comutação.

o encaminhamento apropriado. A Figura 2.1 mostra a estrutura de uma célula ATM. A explicação de alguns dos campos do cabeçalho será feita mais adiante.

Além das células de informação , existem ainda outros tipos de células que foram projetadas para fins de gerenciamento da rede. São as células de gerenciamento de recursos da rede[2].

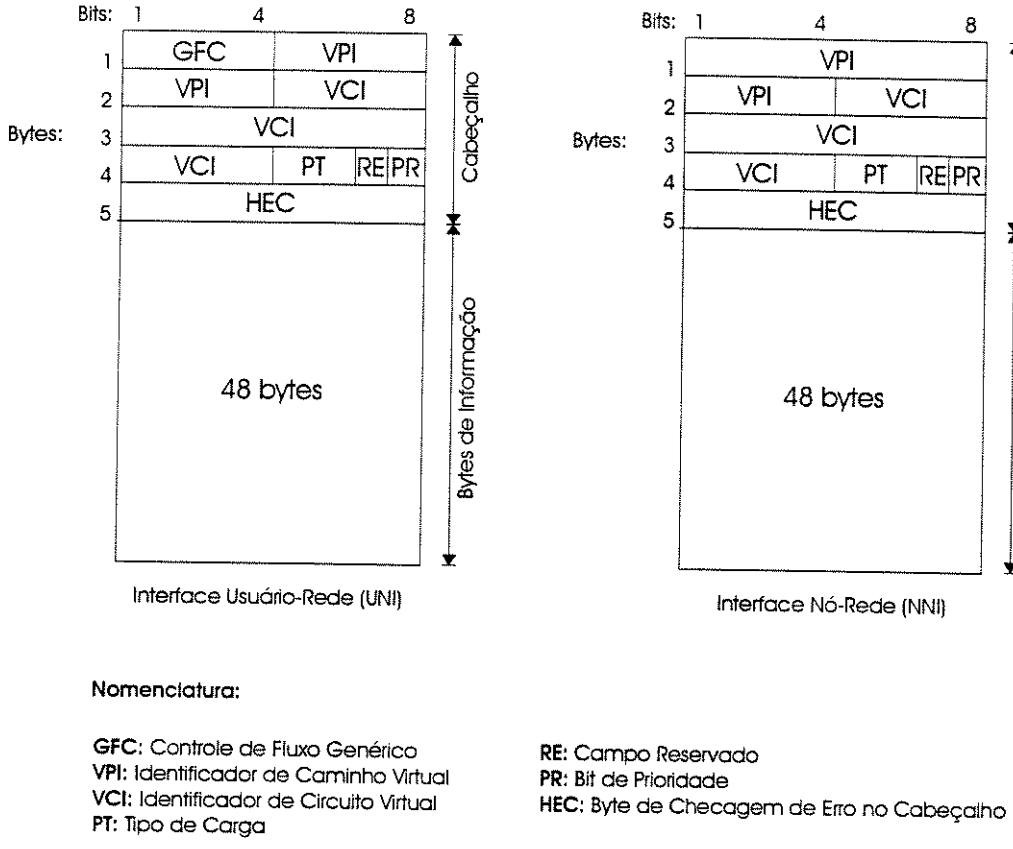

FIGURA 2.1: Estrutura da célula ATM.

2.2.2 O Modo de Transmissão

O modo de transferência ATM é *orientado para conexão*. Isto significa que antes que a informação seja transmitida de um terminal de usuário para a rede, dá-se início a uma fase de estabelecimento de um caminho ponto-a-ponto, chamado de *conexão lógica/virtual*, a fim de que a rede possa reservar os recursos necessários, caso estejam disponíveis. Se não houver recursos disponíveis suficientes, a conexão é recusada para aquele terminal [3]. No ATM, estão previstos dois tipos de conexões : as conexões por *canais virtuais* (VCC) e as conexões por *caminhos virtuais* (VPC).

As conexões por *canais virtuais* são identificadas no subcampo do cabeçalho da célula ATM denominado de **VCI** ("Virtual Channel Identifier"). O **VCI** possui significado apenas local, no enlace entre nós ATM, e será traduzido em cada nó ATM. Cada conexão é caracterizada por um conjunto de valores de VCI que são definidos durante o estabelecimento dessa

[2]Ver seção 2.2.7.

[3]Na verdade, é verificado se existem recursos suficientes para serem compartilhados *estatisticamente*.

conexão. Quando a conexão termina, os valores de **VCI** nos enlaces envolvidos são liberados para serem utilizados por outras conexões. Tipicamente, considera-se um número de canais virtuais, em um mesmo enlace, da ordem de dezenas de milhares. Isto requer um subcampo no cabeçalho de pelo menos 16 bits para o **VCI** .

As conexões por *caminhos virtuais* são identificadas no subcampo do cabeçalho da célula ATM denominado de **VPI** ("Virtual Path Identifier"). Este tipo de conexão foi criado porque espera-se que uma RDSI-FL precise suportar conexões semi-permanentes entre pontos terminais, nas quais será transportado um grande número de conexões simultâneas. Neste tipo de conexão, os recursos da rede são alocados também de forma semi-permanente, para permitir um gerenciamento simples e eficiente desses recursos. O gerenciamento desses caminhos virtuais será desempenhado, tipicamente, não apenas através de uma conexão lógica de baixa taxa de bit, mas também, na maioria dos casos, por um conjunto de conexões lógicas. Portanto, o campo **VPI** é composto de 8 a 12 bits, para permitir 256 a 4096 caminhos virtuais. Cada caminho pode englobar até 64K canais virtuais, identificados por seus respectivos **VCIs**.

Na Figura 2.2 vemos um exemplo de um caminho virtual estabelecido entre o usuário A e o usuário C, transportando duas conexões individuais, cada uma com um VCI separado. Neste exemplo, os valores de VCI utilizados (3 e 4) não são traduzidos nos nós, os quais estão apenas comutando no campo VPI. Além disso, um caminho virtual entre A e B é estabelecido semi-permanentemente, usando-se os valores de VCI iguais a 1, 2 e 3. Podemos notar que o valor de VCI igual a 3 é utilizado duas vezes no enlace entre A e o nó 1. Isto não gera nenhum problema, uma vez que os valores distintos de VPI permitem que os dois pontos terminais (A e 1) diferenciem entre as duas conexões virtuais. No entanto, este exemplo não mostra o caso geral no qual ocorrem mudanças tanto no VPI quanto no VCI.

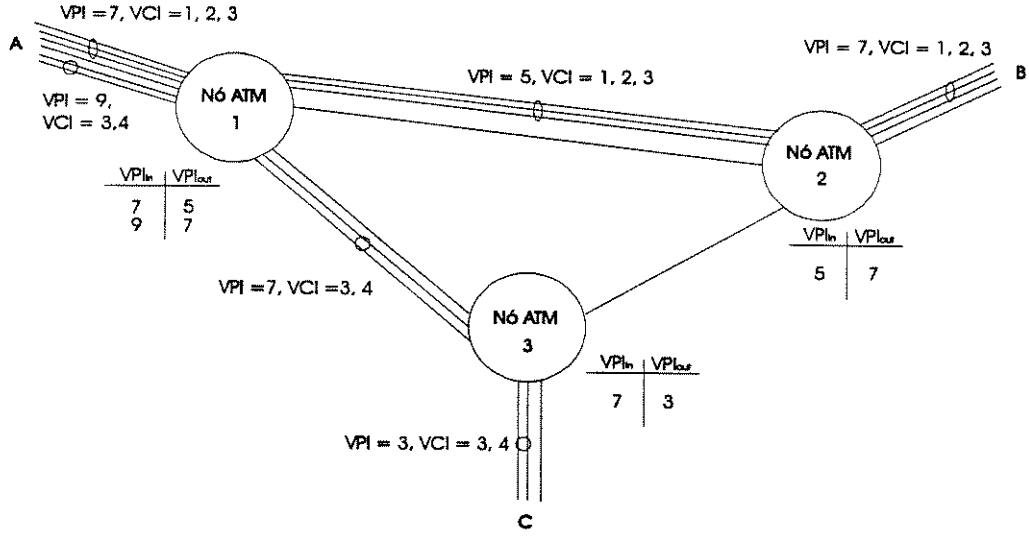

FIGURA 2.2: Exemplo de uso do Identificador de Caminho Virtual.

2.2.3 Recursos da Rede

Como o ATM é *orientado para conexão*, as conexões são estabelecidas de forma semi-permanente ou temporariamente. Este estabelecimento inclui não apenas a alocação de um VCI e/ou VPI, mas também a alocação dos recursos exigidos ao longo da rede e no acesso

do usuário. Estes recursos são expressos em termos de **largura de banda** e **qualidade de serviço (QoS – Quality of Service)**, os quais são negociados pelo usuário durante a fase de estabelecimento da conexão e, possivelmente, durante a transmissão.

Alocação de Banda

A alocação de banda trata da determinação da quantidade de largura de banda dedicada a uma conexão para garantir a qualidade de serviço por ela requerida. Existem duas aproximações alternativas para alocação de banda : a *multiplexagem determinística* e a *multiplexagem estatística.*

Na *multiplexagem determinística* a alocação é baseada na largura de *pico* da banda exigida pela conexão. Um exemplo de aplicação desse tipo de multiplexagem é o modo de transferência conhecido por "Circuit Switching" ou Comutação de Circuitos. Este modo de transferência tem sido usado nas redes telefônicas e ainda é aplicado na RDSI de faixa estreita [15]. Este modo de transferência é baseado no STM — Modo de Transferência Síncrono ("Synchronous Transfer Mode") — no qual a regra de subdivisão e alocação de faixa baseia-se na designação de intervalos de tempo ("time slots"), para cada serviço, dentro de uma estrutura recorrente ("frame") enquanto durar a conexão. Um canal STM (isto é, um "circuito") é identificado pela posição do seu "time slot" dentro do "frame".

A Comutação de Circuitos é bastante inflexível. Uma vez que a duração de um "time slot" é definida, a taxa de bit respectiva fica automaticamente fixada. Por exemplo, no PCM ("Pulse Code Modulation") a duração do "time slot" básico é de 8 bits por 125μs, o que dá origem a canais de 64 Kbit/s. Dessa forma, com apenas um canal básico para transferência de informação, a comutação de circuitos é bastante inadequada para outros tipos de serviços que possuam exigências de taxas de bit diferentes.

Em princípio, a maior taxa de bit envolvida deve ser selecionada como a taxa de bit básica, para ser possível o transporte de todos os serviços. Por exemplo, se a maior taxa de bits dos serviços envolvidos é de 140 Mbit/s, um serviço de 1 kbit/s deve ocupar um canal **completo** de 140 Mbit/s durante toda a duração da conexão, o que significa um grande desperdício de recursos da rede.

Para superar a inflexibilidade de uma única taxa de bit, como é o caso da Comutação de Circuitos, foi desenvolvida uma nova versão chamada de "Multirate Circuit Switching" ou Comutação de Circuitos a Multitaxas [15]. Neste modo de transferência, cada conexão pode alocar um múltiplo n de uma taxa de canal básica, definida previamente. Este número n corresponde ao menor conjunto de canais básicos cuja soma é maior ou igual à taxa de pico da conexão em questão.

Um grande problema deste tipo de alocação é a definição da taxa básica (ou canal básico), pois alguns serviços requerem baixas taxas de bit (por exemplo, voz a 64 Kbit/s) enquanto outros exigem altíssimas taxas de bit (por exemplo, HDTV – "High Definition TV" – a 35Mbit/s). Uma solução para este tipo de problema foi proposta por [17], na qual o quadro de referência básico ("frame") é dividido em intervalos de tempo ("time slots") diferentes, como mostra a Figura 2.3.

Nesta figura, um canal de 139.264 Kbit/s é multiplexado junto com 7 canais de 2048 Kbit/s, 30 canais do tipo B e um canal do tipo D, de 64 Kbit/s, para sinalização. Além disso, um canal de sincronização é definido para permitir a determinação dos limites do quadro. Mesmo assim, grandes quantidades de largura de banda são desperdiçadas, principalmente nas conexões que apresentam uma alta relação *taxa de pico/taxa média de bits.* Neste sentido,

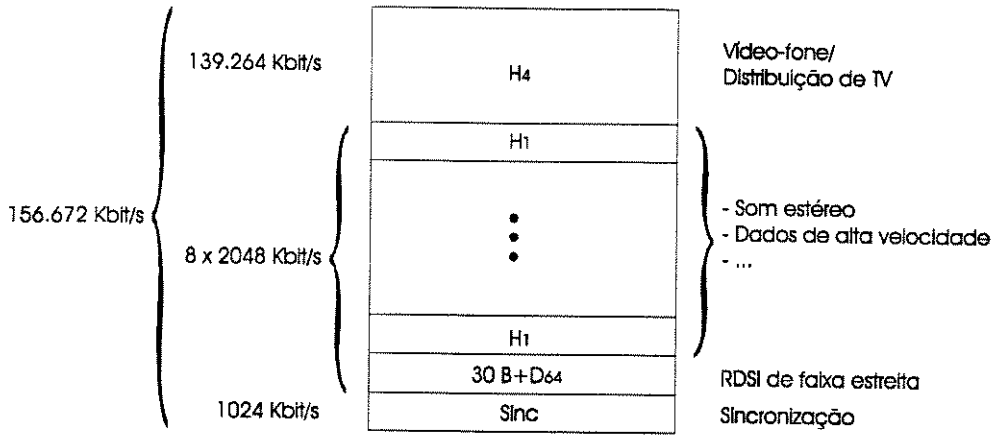

FIGURA 2.3: Comutação de Circuitos a Multitaxas com diferentes canais básicos.

a multiplexagem determinística contrapõe-se à filosofia ATM, pois não permite um compartilhamento eficiente dos recursos da rede entre as diversas conexões existentes.

Um método alternativo para alocação de banda é a *multiplexagem estatística*. Este paradigma, escolhido para as redes ATM, promete ser o grande responsável pela agregação de qualquer tipo de serviço dentro da rede, independentemente de suas características de largura de banda ou qualidade de serviço requerida. De acordo com este princípio, os recursos da rede são compartilhados *estatisticamente* entre as diversas conexões virtuais, ao contrário da multiplexagem determinística, onde os recursos são alocados com *exclusividade* para cada conexão solicitada.

Neste método, a quantidade de largura de banda alocada para uma fonte com Taxa de Bit Variável (VBR)[4], por exemplo, é menor do que a sua taxa de pico, mas necessariamente maior do que sua taxa média de bits. Esta largura de banda alocada é comumente denominada de *largura de banda estatística*. Dessa forma, a soma das taxas de pico das conexões multiplexadas em um enlace pode ser maior do que a largura de banda do enlace, mas a soma de suas larguras de banda estatísticas é menor ou igual à largura de banda do enlace.

A eficiência em largura de banda devido à multiplexagem estatística aumenta à medida que as larguras de banda estatísticas aproximam-se de suas taxas médias, e diminui à medida que se aproximam de suas taxas de pico. Em geral, a multiplexagem estatística permite que mais conexões sejam multiplexadas na rede em relação à multiplexagem determinística, permitindo uma melhor utilização dos recursos da rede.

Para se ter uma idéia do ganho de multiplexagem que é possível se obter com a multiplexagem estatística, consideremos o caso de um tráfego agregado composto de seqüências de vídeo codificadas via MPEG-2 [5]. Se analisarmos a função densidade de probabilidade da taxa de bits de uma seqüência de quadros do tipo "B", notamos que ela pode ser aproximada por uma função densidade de probabilidade do tipo *Gama* [18], como mostra a Figura 2.4(a). Considerando a hipótese de superposição independente de fontes de tais seqüências, a função densidade de probabilidade do tráfego agregado será a convolução das f.d.p.'s individuais, como mostra a Figura 2.4(b), onde foram multiplexadas 16 fontes. Podemos notar, através desta figura, que a taxa de pico do tráfego agregado corresponde a aproximadamente 60

[4]Ver seção ??.

[5]O MPEG-2 ("Moving Picture Experts Group") é um padrão de codificação de vídeo digital largamente utilizado, baseado em codificação preditiva e na Transformada Discreta de Cosseno (DCT).

Mbit/s, o que significa um ganho em largura de banda bem maior, se comparado à alocação exclusiva de 16 canais de 10 Mbit/s, como é o caso da multiplexagem determinística.

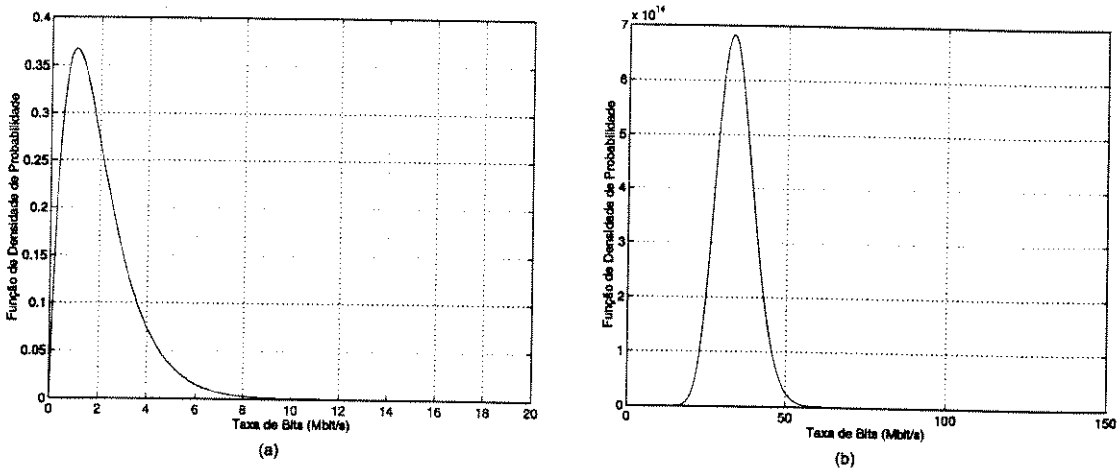

FIGURA 2.4: (a) - Aproximação da função densidade de probabilidade da taxa de bits dos quadros do tipo "B" de uma seqüência de vídeo digital codificada via MPEG-2. (b) - Convolução de 16 fontes de seqüências de quadros do tipo "B".

No entanto, é importante enfatizarmos que a utilização da multiplexagem estatística implica na possibilidade de perda, atraso e variação de atraso de células, conceitos explicados a seguir.

Qualidade de Serviço

A qualidade de serviço de uma conexão está relacionada à *perda de células*, ao *atraso de transferência de células* e à *variação do atraso das células* pertencentes a uma dada conexão. Para o ATM, a qualidade de serviço de uma conexão está também intimamente relacionada à largura de banda que ela utiliza, pois é preciso se garantir a largura de banda mínima exigida pela conexão, ao longo da rede, para se ter o mínimo de qualidade de serviço.

A *perda de células* na rede ATM se dá, basicamente, por dois motivos :

- **Encaminhamento equivocado :** A perda de células por encaminhamento equivocado é causada pela má interpretação dos cabeçalhos das células nos comutadores da rede. Normalmente, este tipo de problema é solucionado através da requisição de retransmissão (ARQ). Em redes ATM, no entanto, nenhuma ação dinâmica está definida contra a perda de células. Apenas ações preventivas serão possíveis, através da alocação de recursos na fase de estabelecimento das conexões, verificando-se a existência de recursos suficientes disponíveis na rede.
- **Transbordamento de células nos buffers da rede :** A perda de células devido ao transbordamento dos *buffers* nos comutadores da rede será, com certeza, um dos principais problemas da rede ATM. Como não está previsto nenhum controle de fluxo nos nós da rede[6], existe a possibilidade de que os *buffers* sejam sobrecarregados pelas seqüências de células das várias fontes, com diferentes taxas de bit. Atualmente, este é o

[6]O ITU-T (International Telecommunications Union - Telecommunications) prevê um controle de fluxo apenas na Interface Usuário-Rede, através de um mecanismo chamado de Controle de Fluxo Genérico (GFC).

problema que mais tem preocupado os projetistas de redes ATM, em função da recente descoberta do caráter *auto-similar* ou *fractal* dos principais tributários que comporão o tráfego da rede ATM, como por exemplo, tráfego com taxa de bit variável (VBR) e tráfego Ethernet de redes locais. As características e implicações deste tipo de tráfego serão vistas mais adiante e consistem em objeto de estudo desta dissertação .

Na rede ATM alguns serviços poderão se beneficiar de uma indicação explícita de Prioridade de Perda de Células (CLP), incluída em um bit específico no cabeçalho, como um meio de gerenciar a perda de células durante períodos de congestionamento na rede. Isto permite ao usuário a escolha entre dois níveis de razões de perda de células em uma única conexão virtual : alta prioridade, para células contendo informações essenciais e baixa prioridade, para células sujeitas a descarte, dependendo das condições da rede. Para classes de serviços[7] como a de Taxa de Bit Variável (VBR) e a de Taxa de Bit Disponível (ABR), será necessário que a rede garanta uma capacidade mínima. Assim, em tempos de congestionamento, a rede pode descartar algumas células ATM sem prejudicar a qualidade mínima requerida.

O *atraso de transferência de células* através da rede é determinado por diferentes partes da rede, cada uma das quais contribuindo individualmente para o atraso total. Estes atrasos estão mostrados na Figura 2.5.

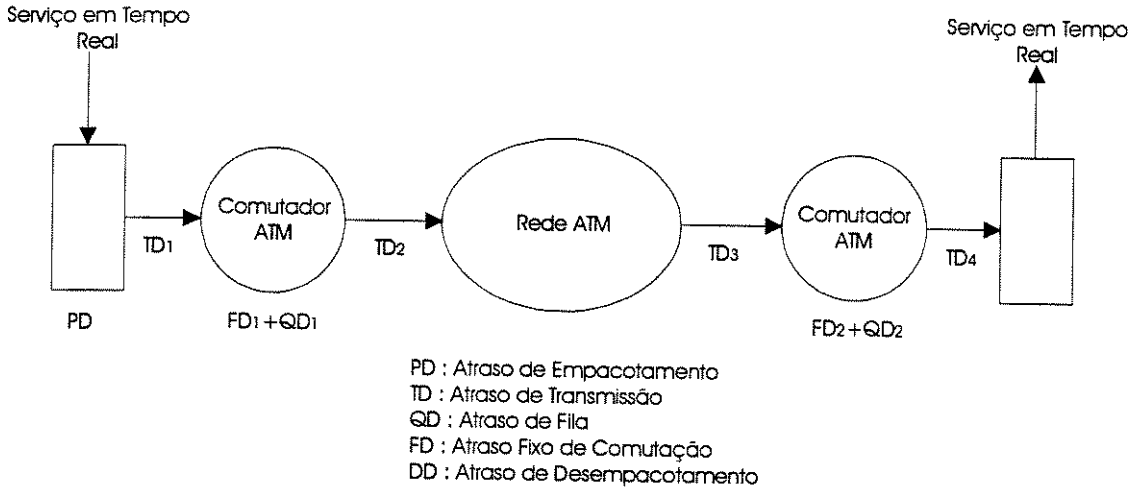

FIGURA 2.5: Fontes de atraso na rede ATM.

De acordo com a Figura 2.5, os seguintes parâmetros contribuem para o atraso na rede :

- **Atraso de Transmissão (TD) :** Este atraso depende das distâncias entre os pontos terminais e do meio de transmissão utilizado. Este é independente do modo de transferência utilizado;
- **Atraso de Empacotamento (PD) :** Este atraso é introduzido toda vez que um serviço é transformado em pacotes (células);
- **Atraso de Comutação :** O atraso de comutação é composto por duas partes :
  - **Atraso de Comutação Fixo (FD)** : Este atraso depende da arquitetura dos comutadores da rede, e é causado por atrasos na transferência dos pacotes através do "hardware" do comutador.

[7] O conceito e descrição de classes de serviços previstas para a rede ATM é analisado na seção 2.4

- **Atraso de Fila (QD)** : Como o ATM realiza a multiplexagem estatística dos pacotes nos comutadores, as filas são necessárias para agregar e absorver o volume de células de diversas fontes direcionadas a um dado enlace. Essas filas são decorrentes dos *multiplexadores estatísticos*, que são as peças-chave para garantir o alto desempenho da rede. O atraso varia com a carga da rede e é determinado pelo comportamento estatístico das filas.

- **Atraso de Desempacotamento (DD) :** Este atraso é adicionado nos pontos terminais da rede a fim de garantir a reconstrução sincronizada da seqüência de bits. Este atraso de desempacotamento tende a reduzir o efeito da *variação do atraso das células* ("jitter") introduzido pela rede.

A *variação do atraso de células* ("Cell Delay Variation" – CDV) refere-se a todos os fenômenos responsáveis pela variação no atraso de transferência de células de uma certa conexão. Durante a sua transferência, a célula pode encontrar diferentes condições de tráfego nos *buffers* compartilhados e, portanto, experimentar diferentes atrasos de transferência, como por exemplo [3] :

- Algumas células sofrem um atraso mais curto do que as células que as precederam. Durante um curto período de tempo, a taxa de pico de células observada pode ser maior do que a taxa de células da fonte. Este fenômeno é denominado de *Efeito de Agrupamento* ou "clumping effect" ;
- Algumas células sofrem um atraso maior do que as células que as precederam. Este é o *Efeito de dispersão.*

2.2.4 A Comutação em Redes ATM

Um dos principais componentes da Rede ATM é o **comutador de células ATM**. É ele o responsável pela realização, na prática, da **multiplexagem estatística** de tráfegos com diferentes taxas de bit e exigências de QoS em enlaces de transmissão com taxas de bit constantes [15, 9].

A operação de multiplexagem estatística é realizada em cada nó de comutação da rede e é composta de três funções básicas, a saber :

- **Decomposição ou Desmultiplexação** : Esta função trata de coletar as células à entrada do comutador, ler os seus cabeçalhos e separá-las de acordo com os seus destinos.
- **Comutação** : Esta função trata de encaminhar as células desmultiplexadas para as portas de saída correspondentes aos canais virtuais às quais elas pertencem. Os canais virtuais estão caracterizados pelos números das portas *físicas* de entrada e saída e os valores do Identificador de Canal Virtual (VCI) e do Identificador de Caminho Virtual (VPI) à entrada e à saída do comutador em questão.
- **Multiplexagem** : Esta função é responsável pela agregação das células destinadas à mesma porta de saída em uma única fila ("buffer"), a fim de aguardarem a vez de serem transmitidas.

A Figura 2.6 mostra um comutador ATM típico [15] com n enlaces de entrada e q enlaces de saída. Cada enlace E_i de entrada representa o fluxo das células ATM que foram agregadas no nó anterior da rede. Nesta Figura, podemos perceber que as células ATM são comutadas de uma entrada E_i para uma saída S_j e que, ao mesmo tempo, os valores de seus cabeçalhos são

alterados, deixando de indicar um valor para indicar um valor β. Em cada enlace de entrada e saída individual, os valores dos cabeçalhos são únicos. No entanto, cabeçalhos idênticos podem ser encontrados em enlaces distintos (por exemplo, o cabeçalho com valor x nos enlaces E_1 e E_n). A comutação de células de E_i para S_j $(i = 1, 2, \ldots, n; j = 1, 2, \ldots, q)$ é realizada por uma *tabela ou matriz de comutação* , como mostra a Tabela 2.1. Neste exemplo, vemos que todas as células que têm um cabeçalho igual a x no enlace E_1 são comutadas para a saída S_1, e seus cabeçalhos são "comutados" para o valor k. Todas as células com um cabeçalho x no enlace E_n são também comutadas para a saída S_1, mas seus cabeçalhos assumem o valor n.

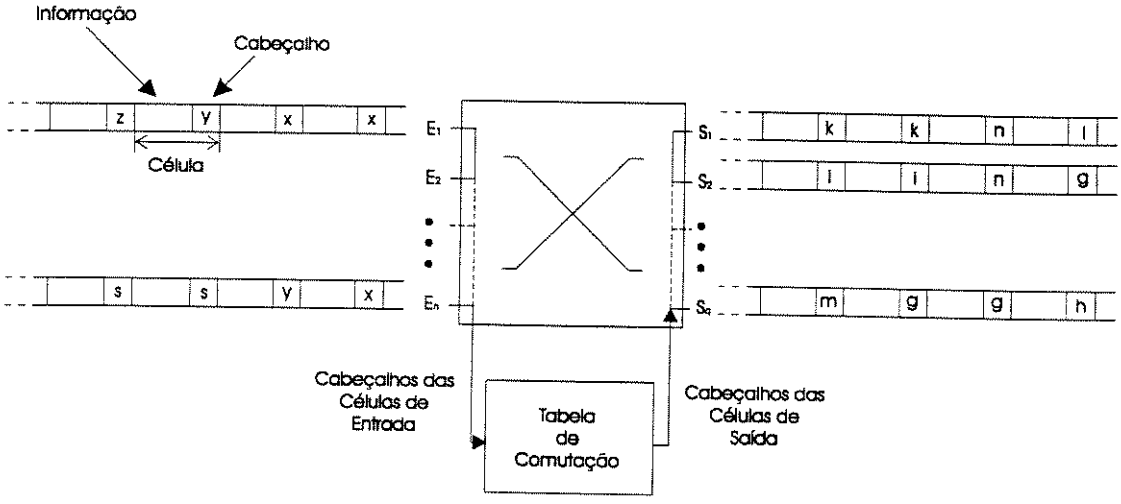

FIGURA 2.6: Comutador de células ATM.

Enlace de Entrada	Cabeçalho	Enlace de Saída	Cabeçalho
E_1	x	S_1	k
	y	S_q	m
	z	S_2	l
.			
.			
.			
E_n	x	S_1	n
	y	S_2	i
	s	S_q	g

TABELA 2.1: Tabela de Comutação

Deve-se observar que é possível que duas células de diferentes entradas (por exemplo, E_1 e E_n) cheguem simultaneamente à entrada do comutador e sejam destinadas para a mesma saída (S_1 por exemplo). Naturalmente, elas não podem ser postas na mesma saída, ao mesmo tempo. Portanto, em algum lugar no comutador, deve ser colocado um *buffer* para armazenar a(s) célula(s) que não podem ser transmitidas naquele instante de tempo. A localização desse *buffer* vai depender da arquitetura escolhida para a fabricação do comutador.

2.2.5 Controle de Tráfego

Uma característica essencial exigida de uma rede de faixa larga é o compartilhamento eficiente dos recursos da rede entre os diversos tipos de tráfego, ao mesmo tempo em que

se garante um certo grau de controle sobre a qualidade de serviço oferecida ao usuário. A natureza das garantias de desempenho que podem ser feitas ao usuário dependem do **controle de tráfego** aplicado.

De acordo com a Recomendação I.371 do ITU-T, o principal propósito do controle de tráfego em uma RDSI-FL é proteger o usuário e a rede de modo a alcançarem objetivos de desempenho pré-definidos, expressos em termos de *razão de perda de células, atraso de transferência de células* e *variação do atraso de células.*

Basicamente, podemos afirmar que o controle de tráfego refere-se ao conjunto de ações tomadas pela rede a fim de evitar o congestionamento. Congestionamento este, que pode ser causado por flutuações estatísticas *não previstas* de fluxos de tráfego, ou por falhas dentro da rede, ocasionando uma perda excessiva de células e/ou atrasos inaceitáveis.

Os objetivos do controle de tráfego na RDSI-FL podem ser assim resumidos :

- **Flexibilidade** : o controle de tráfego deve atender a todo o conjunto de *classes de serviços* definidas para a rede ATM ;
- **Simplicidade** : o controle de tráfego a ser empregado deve ser simples o bastante de forma a minimizar a complexidade do equipamento da rede, ao mesmo tempo em que maximiza a sua utilização ;
- **Robustez** : o controle de tráfego deve garantir uma alta eficiência no uso dos recursos da rede sob qualquer circunstância de tráfego.

A fim de alcançar os objetivos mencionados acima, estão definidas duas funções para o gerenciamento e o controle de tráfego em redes ATM : O *Controle de Admissão de Conexão (CAC)* e o *Controle de Parâmetros do Usuário (UPC).*

Controle de Admissão de Conexão

O *Controle de Admissão de Conexão* representa o conjunto de ações tomadas pela rede com o objetivo de aceitar ou rejeitar uma nova conexão. A aceitação de uma nova conexão é feita apenas quando recursos suficientes estão disponíveis para suportarem a qualidade de serviço exigida pela nova conexão, sem comprometer, ao mesmo tempo, a qualidade de serviço das conexões já estabelecidas dentro da rede.

Durante a fase de estabelecimento da conexão, as seguintes informações, contidas nas especificações de um contrato de tráfego, devem ser negociadas e acordadas entre o usuário e a rede para que o CAC realize decisões confiáveis de aceitação/negação de uma conexão :

- Limites específicos do volume de tráfego que a rede deve transportar, em termos de descritores de tráfego bem definidos ;
- A classe de serviço requerida, expressa em termos de atraso de transferência de células, variação do atraso de células e razão de perda de células ;
- Uma tolerância para a variação do atraso de células introduzida. Por exemplo, variação do atraso de células introduzida pelo Equipamento Terminal (TE) ou Equipamento do Usuário (CPE).

Estas informações podem ser renegociadas pelo usuário durante todo o tempo em que a conexão estiver ativa. Todavia, a rede pode limitar a freqüência dessas renegociações. Os esquemas de CAC não estão, até o momento, padronizados.

Controle de Parâmetros de Usuário e de Rede

O *Controle de Parâmetros de Usuário* e o *Controle de Parâmetros de Rede* (UPC/NPC) são realizados na Interface Usuário-Rede (UNI) e na Interface Rede-Nó (NNI), respectivamente. Eles representam o conjunto de ações tomadas pela rede para monitorar e controlar o tráfego, em uma conexão ATM, em termos de volume de tráfego de células e validade do encaminhamento das células. Esta função é chamada também de "Função de Policiamento". Seu principal propósito é forçar o cumprimento do contrato de tráfego negociado de modo que a QoS das outras conexões sejam garantidas.

Um algoritmo de UPC/NPC eficiente deve exibir as seguintes características :

- Capacidade de detectar qualquer situação de tráfego ilegal;
- Tempo de resposta rápido às violações dos contratos;
- Simplicidade de Implementação.

Especificação dos Parâmetros de Tráfego

Com o objetivo de garantir uma operação apropriada das funções de CAC e de UPC/NPC, as características intrínsecas dos tráfegos das conexões devem ser descritas adequadamente por um conjunto padronizado de parâmetros de tráfego.

Um parâmetro de tráfego é uma especificação de um aspecto particular de tráfego. Os parâmetros de tráfego podem, por exemplo, descrever aspectos quantitativos (tais como o tempo médio de duração da conexão, a taxa de pico de células, a taxa média de células, a duração média de um surto, etc.) ou um aspecto qualitativo (tal como o tipo de fonte : telefone, vídeo-fone, etc.). A lista de todos os parâmetros especificados de tráfego é denominada de *descritor de tráfego ATM.*

Os parâmetros de tráfego usados em um descritor de tráfego ATM deveriam, idealmente, ter as seguintes propriedades :

- Serem facilmente entendidos pelo usuário;
- Serem significativos, para fins de alocação de recursos;
- Serem facilmente verificáveis pela rede.

De fato, para conexões com taxas variáveis, é difícil se atender a todas as três exigências simultaneamente. Por exemplo, a taxa média de uma conexão é, certamente, muito significativa para alocação de recursos, mas pode ser difícil para o usuário predizê-la inicialmente. Sendo assim, devido à importância da capacidade de se prevenir que os usuários enviem tráfegos acima dos declarados, provocando congestionamentos, os parâmetros de tráfego padronizados são, atualmente, **baseados em uma regra**. A regra adotada atualmente para a RDSI-FL é o **Algoritmo Genérico de Taxa de Células (GCRA)**. Este algoritmo foi definido inicialmente pelo Forum ATM ao generalizar a definição de Taxa de Pico de Células feita pelo ITU-T [15].

O GCRA(T, τ) determina se uma seqüência de células está a uma taxa nominal $1/T$ sujeita à tolerância τ. Mais precisamente, a célula k está "conformante" com o GCRA(T, τ), em uma dada interface, se, e somente se,

$$y_k = c_k - a_k < \tau, \tag{2.1}$$

onde a_k é o instante no qual a célula k é observada na interface e c_k é o "instante teórico", definido recursivamente por :

$$\begin{aligned} c_{k+1} &= c_k + T \quad \text{se } \tau \geq y_k \geq 0, \\ &= a_k + T \quad \text{se } y_k < 0, \\ &= c_k \quad \text{se } y_k > \tau. \end{aligned} \tag{2.2}$$

No caso do parâmetro Taxa de Pico de Células, T é o Intervalo de Emissão de Pico ($T = 1/PCR$) e τ é tolerância da Variação de Atraso de Célula (CDV).

Enquanto que os descritores de tráfego baseados em regras são bastante atrativos de um ponto de vista prático (notadamente para testes de conformação e controle de tráfego), a seleção ótima desses parâmetros permanece um problema em aberto.

2.2.6 Sinalização

A negociação entre o usuário e a rede com relação aos recursos (VCI/VPI, vazão, QoS) é realizada através de um canal separado (virtual) de sinalização. O protocolo de sinalização a ser usado neste canal consiste de um aprimoramento do protocolo utilizado na sinalização da RDSI de faixa estreita.

Para configurações ponto-a-ponto na Interface Usuário-Rede (UNI) existe um canal de sinalização pré-definido. No caso de configurações ponto-multiponto, onde múltiplos terminais compartilham uma única interface, múltiplos canais virtuais de sinalização podem ser estabelecidos (pelo menos um por terminal) via o canal de *meta-sinalização*. Este canal de *meta-sinalização* é transportado por um VPI/VCI pré-definido na Interface Usuário-Rede.

2.2.7 Controle de Fluxo

Controle de Fluxo Genérico

A princípio, o ATM não deveria suportar controle de fluxo para seqüências de informações. Acreditava-se que um controle de fluxo iria requerer funções adicionais nos cabeçalhos da células ATM, além de uma maior quantidade de *buffers* dentro da rede, o que implicaria em equipamentos mais complexos, maiores atrasos, maior variação de atraso, vazão reduzida, etc..

Porém, reconheceu-se que, em alguns casos, o fluxo de tráfego nas conexões de um terminal para a rede deveria ser controlado. A fim de tornar isso possível, o ITU-T propôs um mecanismo chamado de GFC (Controle de Fluxo Genérico) para ser utilizado na Interface Usuário-Rede (UNI). Esta função é desempenhada por um campo específico no cabeçalho da célula ATM.

Dois conjuntos de procedimentos estão previstos para uso dentro do campo GFC : "Transmissão Não-Controlada" e "Transmissão Controlada". O procedimento de "Transmissão Não-Controlada" é para uso em configurações ponto-a-ponto. O procedimento de "Transmissão Controlada" pode ser usado em ambas as configurações (ponto-a-ponto e configurações de meios compartilhados). Este procedimento ainda se encontra sob estudos. No entanto, ele deve garantir uma utilização eficiente e justa da capacidade disponível entre o terminal e a rede, além de ser independente da topologia da rede (anel, estrela, etc.).

Células de Gerenciamento de Recursos

Um segundo mecanismo de controle de fluxo foi proposto pelo ITU-T especialmente para a classe de serviço ABR ("Available Bit Rate"). O objetivo desta classe de serviço é utilizar toda a capacidade de vazão da rede que não estiver sendo usada pelas conexões "regulares". As conexões ABR podem conviver com vazões variáveis (taxas elásticas) e maiores atrasos. Em contrapartida, não permitem grandes perdas de células. Por isso, o sistema supervisor da rede necessita de um meio eficiente e rápido de indicar a disponibilidade dos recursos da rede para os usuários dessa classe de serviço.

Este mecanismo é, seguramente, mais complexo e sofisticado do que o GFC. Seu princípio é baseado no controle da taxa de células de um terminal através da inserção de células ATM especiais, chamadas de *Células de Gerenciamento de Recursos* (RM). As células de RM são diferenciadas através de um código especial no sub-campo PT do cabeçalho da célula ATM. Elas têm a função especial de informar à fonte quando aumentar ou diminuir sua taxa de células. A rede permite o aumento gradual da taxa de células até que ocorra algum congestionamento, em algum lugar da rede, que inclua a conexão em questão. O congestionamento em qualquer parte da rede ATM é sinalizado através do bit EFCI ("Explicit Forward Congestion Indication") no cabeçalho das células ATM de todas as conexões afetadas. Ao receber as células dos usuários com o bit de EFCI presente, o ponto de destino ajusta as indicações do controle de largura de banda nas células de RM direcionadas para a fonte.

2.3 O Modelo de Referência do Protocolo ATM

Em 1983 um modelo geral de referência de protocolo para Interconexão de Sistemas Abertos ("Open System Interconnection" – OSI) foi aprovado como padrão para transmissão de dados em redes de comunicação [15, 16]. A mesma arquitetura hierárquica lógica foi adotada para a rede ATM na Recomendação I.321. O modelo adotado utiliza o conceito de planos separados para fins de segregação das funções de usuário, controle e gerenciamento. No entanto, apenas as camadas mais baixas estão definidas.

O modelo de protocolo RDSI-FL para ATM é mostrado na Figura 2.7. Este modelo contém três planos : um plano no nível de usuário, para transportar informação do usuário; um plano de controle, composto de informação de sinalização (principalmente), e um plano de gerenciamento, usado para desempenhar funções operacionais e de manutenção da rede. Além disso, uma terceira dimensão é adicionada ao modelo, chamada de plano de gerenciamento, o qual é responsável pelo gerenciamento dos diferentes planos.

Para cada plano, é utilizada uma aproximação em camadas como no modelo OSI, as quais apresentam uma interdependência. De acordo com o ITU-T, as camadas podem ser divididas mais ainda, como descrito na Tabela 2.2. Três camadas estão definidas : a camada física (PHY), que transporta informação (bits/células); a camada ATM, que desempenha encaminhamento, comutação e multiplexagem; e a camada de adaptação ATM (AAL), que é responsável pela adaptação dos serviços de informação às seqüências de células ATM. Estas camadas podem ainda ser divididas em subcamadas. Cada subcamada realiza uma série de funções. A seguir, descreveremos, brevemente, cada camada.

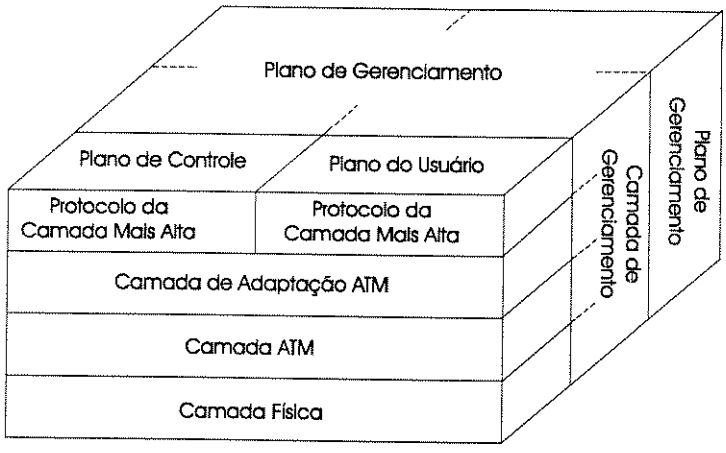

FIGURA 2.7: Modelo de Referência do Protocolo ATM.

Convergência Segmentação e Reagrupamento	Sub-Camada de Convergência Sub-Camada de Segmentação e Reagrupamento (SAR)	AAL
Controle de Fluxo Genérico Leitura de VPI/VCI Multiplexagem/Desmultiplexagem de Células Desacoplamento da Taxa de Células		ATM
Desacoplamento da Taxa de Células Geração /Verificação do HEC Embaralhamento/Desembaralhamento de Células Delineação de células Identificação do Caminho do Sinal Justificação da Freqüência Embaralhamento/Desembaralhamento de Quadros Geração e Recuperação de Quadros	Convergência de Transmissão (TC)	PHY
Tempo do Bit Codificação de linha Embaralhamento/Desembaralhamento Dependente do Meio Físico	Meio Físico (PM)	

TABELA 2.2: Funções e Subcamadas do Modelo de Referência do Protocolo

2.3.1 A Camada Física

A camada física da RDSI-FL é ainda decomposta em duas subcamadas : a Subcamada do Meio Físico (PM), que suporta funções de bit dependentes do meio, e a Subcamada de Convergência de Transmissão (TC), que converte as seqüências de células ATM em bits a serem transportados através do meio físico.

A SubCamada do Meio Físico

Esta subcamada é responsável pela correta transmissão e recepção de bits no meio físico apropriado. As funções desempenhadas estão mostradas na Tabela 2.2. No nível mais baixo, esta função é totalmente dependente do meio (óptico, elétrico, etc.) e é chamada de meio físico. Além disso, esta subcamada deve garantir uma reconstrução apropriada do tempo do bit no receptor. As subcamadas do meio físico foram especificadas pelo ITU-T nas Recomendações G.703, G.957 e pelo Forum ATM.

A Subcamada de Convergência de Transmissão

Esta subcamada realiza, basicamente, 5 funções como mostra a Tabela 2.2. A primeira função depois da reconstrução do bit é a adaptação ao sistema de transmissão utilizado. Os sistemas de transmissão possíveis estão especificados nas Recomendações G.708 e G.709 para Hierarquia Digital Síncrona (SDH), e Recomendações G.703 e G.804 para Hierarquia Digital

Plesiócrona (PDH). As células são adaptadas ao sistema de transmissão de acordo com o mapeamento padronizado. O Forum ATM adicionou ainda o FDDI ("Fiber Distributed Data Interface") como uma opção para a Interface Usuário-Rede (UNI).

Esta subcamada é ainda responsável pela geração, no transmissor, do byte HEC (Verificação de Erro no Cabeçalho) de cada célula, além de sua verificação no receptor. Isto permite o reconhecimento dos limites da célula, isto é, a delineação apropriada da célula no receptor. Uma vez determinada a delineação da célula, um mecanismo adaptativo utiliza o HEC para correção e detecção de erros no cabeçalho. Erros em bits isolados são corrigidos. Mas, se ocorrer uma seqüência de várias células com erros no cabeçalho, a correção é abandonada e é feita uma detecção de alta precisão das células com erros, para subseqüente eliminação. Dessa forma, evita-se a passagem de células durante períodos de erros em surtos.

Esta subcamada deve garantir também a inserção e supressão de células vazias para adaptar a taxa útil do sistema de transmissão à carga disponível a ser transmitida. Além disso, informações de Operação e Manutenção (OAM) devem ser trocadas com o plano de Gerenciamento.

2.3.2 A Camada ATM

A camada ATM é totalmente independente do meio físico utilizado para transportar as células ATM e, portanto, é independente da camada física. Esta camada desempenha as seguintes funções :

- Multiplexagem de células de diferentes conexões (identificadas por valores de VCI e VPI diferentes) em uma única seqüência de células na camada física;
- Leitura do identificador de células, a qual é requerida na maioria dos casos, quando se comuta uma célula de um enlace físico para outro. Pode ser o VCI, o VPI ou ambos;
- Fornecer a um usuário de um certo VPC ou VPP a qualidade de serviço requerida. Alguns serviços podem exigir uma certa qualidade de serviço para uma parte do fluxo de células de uma conexão, e uma qualidade de serviço mais baixa para o restante do fluxo. Essa distinção dentro da conexão é feita por meio do bit CLP no cabeçalho da célula;
- Funções de gerenciamento;
- Extração(adição) do cabeçalho das células antes(depois) da célula ser entregue para (pela) camada de adaptação;
- Implementação de um mecanismo de controle de fluxo na Interface Usuário-Rede.

2.3.3 A Camada de Adaptação ATM (AAL)

A Camada de Adaptação ATM (AAL) melhora o serviço executado pela Camada ATM para um nível requerido pela próxima camada mais alta. Ela desempenha funções para os Planos de Usuário, Controle e Gerenciamento, além de suportar o mapeamento entre a Camada ATM e a próxima camada mais alta. As funções executadas na AAL dependem, basicamente, das exigências das camadas mais altas. A camada AAL está subdividida em duas subcamadas : a Subcamada de Segmentação e Reagrupamento (SAR) e a Subcamada de Convergência (CS).

O principal propósito da SAR é a segmentação da informação da camada superior em um tamanho apropriado para a carga útil das células ATM de uma conexão virtual. Inversamente,

ela executa também o reagrupamento dos conteúdos das células de uma conexão virtual em unidades de dados a serem entregues para a camada mais alta.

A Subcamada de Convergência desempenha funções como identificação de mensagens, recuperação do relógio (sincronização), etc. Para alguns tipos de AAL, que suportam transporte de dados via ATM, a Subcamada de Convergência é ainda dividida em uma Subcamada de Convergência de Partes Comuns (CPCS) e uma Subcamada de Convergência Específica de Serviços (SSCS).

Unidades de Dados de Serviços AAL (SDU) são transportados de um Ponto de Acesso de Serviço AAL (SAP) para um ou mais outros AAL-SAP, através da rede ATM. Os usuários da AAL terão a capacidade de selecionar um dado AAL-SAP associado com a QoS exigida para transportar as AAL-SDU.

Até agora, quatro Camadas de Adaptação ATM foram definidas pelo ITU-T, uma para cada classe de serviço. As camadas AAL 3 e a AAL 4, que fornecem adaptação para serviços orientados para conexões e serviços sem conexão, incluem a CPCS. O Forum ATM definiu uma AAL diferente para transferência de dados de alta velocidade, a chamada AAL 5. Esta AAL foi também adotada pelo ITU-T (para sinalização, por exemplo).

2.4 Classes de Serviços ATM

Define-se *modelo de serviço* de uma rede [3] como sendo o conjunto de protocolos e mecanismos de controle de tráfego que permitem suporte a diferentes serviços de telecomunicações, diferenciados por suas características particulares de tráfego e exigências de desempenho.

O modelo de serviço da RDSI-FL é baseado em um conjunto de classes de serviços ATM (ATM transfer capabilities - ATC), padronizadas pelo ITU-T. Este modelo é bastante parecido com o definido pelo Forum ATM, onde diferentes categorias de serviços (similares às ATC) são fornecidas ao usuário.

A definição de classes de serviços é baseada no princípio de que a rede deve satisfazer um certo número de critérios de desempenho relacionados à perda de informação ou atrasos de transmissão, sob a condição de que o tráfego oferecido pelos usuários esteja de acordo com um conjunto de regras. Uma classe de serviço ATM tem como função traduzir um modelo de serviço em um conjunto de caracterizações de tráfego e procedimentos de gerenciamento de tráfego úteis para alguns tipos de aplicações, permitindo uma alocação de recursos eficiente.

Enquanto que as regras e a natureza das garantias de desempenho estão especificadas em detalhes, os padrões definidos não especificam como a rede deve ser operada a fim de se cumprir os compromissos de qualidade de serviço. Além disso, não existe nenhum "guia" sobre quais classes de serviços são mais apropriadas para determinadas aplicações. Tais preocupações são, naturalmente, de grande importância para operadores e, portanto, tem havido um enorme esforço nos últimos anos na busca de soluções satisfatórias. Apesar de toda essa intensa atividade, muitos assuntos relacionados a controle de tráfego continuam sem solução.

Descreveremos a seguir, brevemente, as classes de serviços até agora definidas, indicando algumas diferenças que ainda existem entre o ITU e o Forum ATM.

2.4.1 Deterministic Bit Rate (DBR) ou Constant Bit Rate (CBR)

Um usuário pode ser capaz de negociar e controlar apenas sua taxa de pico. Neste caso, a classe de serviço DBR/CBR é negociada no estabelecimento da conexão. DBR é a denomina-

ção utilizada pelo ITU-T e CBR é a utilizadada pelo Forum ATM. Esta é a classe de serviço ATM mais simples, onde a caracterização do tráfego reduz-se à especificação da Taxa de Pico de Células (PCR) e sua associada tolerância à Variação de Atraso de Células (CDV).

Neste tipo de classe de serviço as células são consideradas independentemente do valor do bit CLP em seus cabeçalhos. As exigências de QoS são expressas em termos de valores desejados para o parâmetro de Taxa de Perda de Células (PCR) e, possivelmente, para parâmetros relacionados a atrasos, como o Atraso de Transferência de Células (CTD) e o Variação de Atraso de Células (CDV).

É importante destacar que a classe de serviço DBR não é destinada apenas para conexões transmitindo aplicações com taxa contínua de células (CBR), uma vez que um usuário pode ou não transmitir continuamente sob a taxa de pico de células negociada; no entanto, a rede deveria ser capaz de oferecer ao usuário uma QoS compatível com o uso de uma conexão DBR transmitindo uma aplicação com taxa constante. O parâmetro PCR pode ser alterado, usando-se a sinalização, enquanto uma conexão está em progresso.

2.4.2 Statistical Bit Rate (SBR) ou Variable Bit Rate (VBR)

O ITU-T utiliza o nome SBR, enquanto que o Forum ATM utiliza o termo VBR para a mesma classe de serviço. As conexões SBR/VBR são descritas por três parâmetros de tráfego : PCR, SCR ("Sustainable Cell Rate") e o IBT ("Intrinsic Burst Tolerance"). O SCR e o IBT são parâmetros baseados em regras que fornecem alguma informação sobre as características das conexões com taxa de bit variável ao mesmo tempo em que se espera que esses parâmetros levem a uma operação mais eficiente da rede, permitindo uma multiplexagem estatística mais controlada.

O ITU-T distingue, atualmente, três classes SBR distintas, as quais se diferenciam pelo uso do bit CLP. Na classe SBR1, as células não são discriminadas com base no bit CLP, e as exigências de CLR são expressas para o fluxo agregado de células. Pode haver ou não a fixação de objetivos para o CTD e o CDV.

Nas classes SBR2 e SBR3 as células são discriminadas com base no bit CLP. Os parâmetros de tráfego negociados para estas classes são o PCR, para o fluxo agregado de células (CLP= 0 ou CLP= 1), e o parâmetro SCR, fixado para o fluxo com CLP= 0. Exigências de QoS são expressas em termos de um objetivo de CLR para o fluxo de células com CLP= 0. Não há nenhuma exigência de QoS prevista relacionada aos parâmetros de atraso ou ao CLR para o fluxo agregado de células. A classe SBR3 diferencia-se da classe SBR2 apenas pelo fato de permitir a mudança de prioridade de células que não estão de acordo com as exigências de QoS, ou seja, é permitido se alterar o bit CLP para o valor 1 (baixa prioridade). A classe SBR2 não permite esta "renomeação ".

O Forum ATM também define três classes, a saber, VBR1, VBR2 e VBR3 à semelhança das definições do ITU-T. Além disso, existe uma distinção entre serviços VBR de tempo real e serviços VBR que não são de tempo real[8]. Serviços VBR de tempo real são bastante exigentes com relação à incidência de atrasos, uma vez que serão apropriados para uma grande variedade de aplicações interativas e de multimídia.

[8]Para o ITU-T esta distinção corresponde a diferentes escolhas de QoS.

2.4.3 ATM Block Transfer (ABT)

A classe de serviço ABT está definida apenas pelo ITU-T. Nesta classe, o usuário é capaz de definir e controlar um estrutura em bloco na sua seqüência de dados. O PCR para cada bloco é negociado entre o usuário e a rede por meio de células de Gerenciamento de Recursos (RM). Duas variantes estão sendo padronizadas atualmente :

- ABT com Transmissão Atrasada (ABT/DT), na qual o usuário espera que a rede aceite explicitamente ou implicitamente um novo bloco ;
- ABT com Transmissão Imediata (ABT/IT), na qual o usuário não espera que a rede aceite um novo bloco. Neste caso, os blocos podem ser descartados pela rede.

Qualquer uma das variantes pode ser *rígida* ou *elástica*. Na versão rígida, a taxa exigida pelo usuário é aceita ou não pela rede. Neste caso, são necessários pedidos de reserva toda vez que um bloco for negado. Na versão elástica, a rede pode oferecer uma taxa menor do que a requerida pelo usuário.

As garantias de QoS para as classes ABT são feitas tanto no nível de células quanto no nível de blocos. A garantia de QoS oferecida para as células nos blocos é equivalente às garantias de QoS oferecidas para as classes DBR e SBR1. A QoS no nível de bloco depende da classe ABT específica :

- Na classe ABT/DT, os blocos podem ser atrasados dentro do equipamento do usuário ; a QoS é caracterizada por um limitante superior no atraso sofrido por um bloco antes de ser aceito pela rede ;
- Na classe ABT/IT os blocos são descartados como um todo, no caso de congestionamento dentro da rede ; a QoS é então caracterizada por um objetivo fixado para a Razão de Perda de Blocos.

Uma característica essencial da classe ABT é o uso de células de RM para realizarem um rápido ajuste da taxa designada para uma conexão, sem precisar da clássica sinalização para renegociar os parâmetros de conexões DBR ou SBR.

2.4.4 Available Bit Rate

As classes de serviços descritas acima tentam evitar o congestionamento através do controle preventivo (com exceção da versão elástica da classe ABT). Em contraste, a classe de serviço ABR utiliza um controle reativo, através do qual as taxas que os usuários podem transmitir são definidas dinamicamente pela rede. O objetivo principal é utilizar toda a capacidade disponível da rede e compartilhá-la igualitariamente entre todos os usuários que possuem exigências de taxas elásticas, isto é, a taxa da conexão não é uma característica intrínseca, mas pode ser fixada livremente entre valores máximo e mínimo declarados. O mecanismo ABR está especificado detalhadamente pelo Forum ATM, e permite várias opções distintas. Em particular, duas possibilidades existem para se determinar a taxa alocada para uma dada conexão :

- A taxa é determinada pelo usuário a partir de um bit de indicação de congestionamento, instruindo o usuário a aumentar ou diminuir sua taxa de acordo com um algoritmo definido ;
- A taxa é calculada pela rede e comunicada explicitamente para o usuário.

A classe ABR tem como objetivo evitar a perda de células através do ajuste das taxas de bit das conexões dentro da capacidade disponível na rede. Atrasos de transmissão não são claramente garantidos além daqueles correspondendo a uma taxa mínima de células (MCR) que pode ser especificada como parte do contrato de tráfego.

Da mesma forma que a classe ABT, a classe ABR baseia-se no uso de células de Gerenciamento de Recursos (RM) para comunicar taxas alocadas ou congestionamento.

As exigências de QoS são muito pouco especificadas, uma vez que, se o usuário adapta sua taxa de transmissão de acordo com a realimentação indicada pelas células de RM, a seqüência de dados do usuário deve experimentar uma *pequena* Razão de Perda de Células.

2.4.5 Unspecified Bit Rate (UBR)

Uma outra classe de serviço foi definida pelo Forum ATM e será, provavelmente, adotada pelo ITU-T. É a classe de serviços com Taxa de Bits Não-Especificada (Unspecified Bit Rate). Esta classe aplica-se a conexões sem parâmetros de tráfego declarados e sem garantias de QoS. Nesta classe, os próprios usuários devem implementar um controle reativo por meio de protocolos em camadas mais altas, tais como o TCP, a fim de alcançar o melhor desempenho possível por parte da rede. O bit EFCI do cabeçalho das células pode ser usado para informar aos usuários a situação de congestionamento da rede.

Capítulo 3

Processos Estocásticos Auto-Similares

3.1 Introdução

Os estudiosos de processos estocásticos têm se dedicado, na maioria das vezes, ao estudo de seqüências de variáveis aleatórias *independentes*, a processos de Markov e a outros tipos de processos que possuem a propriedade básica de que amostras, suficientemente espaçadas, são independentes ou praticamente independentes. De fato, sabe-se que a hipótese de independência é, na maioria dos casos, apenas uma aproximação para a estrutura de dependência real do processo [19, 20].

Na prática, a correlação entre amostras não muito distantes pode ser detectada de uma maneira relativamente fácil, em conjuntos de dados relativamente grandes. Modelos com memória de **curto prazo**, tais como os modelos AR (auto-regressivo), MA (média móvel), ARMA (auto-regressivo e de média móvel) e os processos de Markov, são bem conhecidos e freqüentemente utilizados na prática. Todos eles, sem exceção, apresentam a característica básica de possuírem funções de autocorrelação que decaem *exponencialmente.*

No entanto, diversos estudos em diferentes áreas da ciência, têm relatado a ocorrência de processos estocásticos caracterizados pela presença de *persistência de longo prazo* ou *dependência de longo prazo.* Essa *persistência de longo prazo* é expressa pela evolução dos valores assumidos pelas componentes das séries temporais, nas quais valores consecutivos das amostras (positivos ou negativos) são seguidos por valores de mesma ordem de grandeza e de mesmo sinal, de tal forma que as séries temporais parecem apresentar uma sucessão de ciclos. Em particular, elas apresentam ciclos cujos períodos são da ordem de grandeza do comprimento total T da série. Além disso, estes ciclos, vistos em diferentes amostras da *mesma* série, possuem períodos distintos (mesmo do ponto de vista do comportamento cíclico dominante).

Processos com dependência de longo prazo constituem, de fato, uma importante classe de processos com uma característica bastante peculiar : a *auto-similaridade estatística.* Por um *processo estocástico auto-similar* entende-se um processo cujas características estatísticas independem de escalonamentos no tempo.

Hoje, é possível identificar uma grande variedade de fenômenos naturais e processos em geral que exibem este tipo de comportamento. Como exemplos, podemos citar [21] :

- Séries temporais geofísicas, tais como variação de temperatura, variações pluviométricas, medições de fluxos oceânicos, variação do nível dos rios, variação na freqüência de rotação da terra e variação das manchas solares na superfície do Sol ;
- Séries temporais econômicas, tais como a série temporal da variação do índice Dow

Jones da Bolsa de Valores de Nova York;

- Séries temporais fisiológicas, tais como a taxa instantânea de batimentos cardíacos, variações nos registros de Eletro-encefalogramas de pacientes submetidos a estímulos e taxas de insulina em pacientes com diabete;
- Séries temporais biológicas, tais como tensões através de nervos e membranas;
- Flutuações magnéticas, tais como ruídos de radiação cósmica, intensidade de fontes de luz e o fluxo eletromagnético em supercondutores;
- Ruídos em dispositivos eletrônicos, tais como em transistores bipolares, tubos a vácuo, diodos Zener, Schottky e túnel;
- Variação de freqüência em osciladores de cristais de quartzo e relógios atômicos;
- Fenômenos induzidos pelo homem, tais como variações no fluxo de tráfego de automóveis e a música;
- Padrões de erros em surtos em certos canais de comunicações;
- Variação de textura em terrenos naturais como nuvens, montanhas, etc;
- Tráfego de pacotes em redes de comunicações de alta velocidade;
- Vídeo digital codificado com taxa de bit variável (VBR).

A finalidade deste capítulo é apresentar os principais conceitos e definições relacionados a processos com dependência de longo prazo e a processos auto-similares. Em particular, apresentaremos três dos principais modelos utilizados hoje para se modelar este tipo de processo : o movimento Browniano fracionário (fBm), o processo ARIMA fracionário e o processo de recompensas renovadas.

3.2 Dependência de Longo Prazo e Auto-Similaridade

Um dos primeiros trabalhos[1] a caracterizar o fenômeno da dependência de longo prazo foi o trabalho do hidrólogo Hurst, em 1951 [22]. Hurst estudou os registros de fluxos de rios, como o rio Nilo. Em particular, Hurst estudou o *intervalo seqüencial* de fluxos acumulados de água.

Por *intervalo seqüencial* de um processo estocástico $\{X(t), t \in \mathbb{R}\}$ define-se a medida

$$M(t,T) = \sup_{t \le s \le t+T} [X(s) - X(t)] - \inf_{t \le s \le t+T} [X(s) - X(t)] \; . \tag{3.1}$$

Se $X(t)$ possuir trajetórias amostrais contínuas e os parâmetros t e T forem finitos, podemos trocar sup por max e inf por min .

Os resultados empíricos de Hurst fizeram-no concluir que o *intervalo seqüencial* de fluxos acumulados de água é proporcional a T^H, onde $\frac{1}{2} < H < 1$. Este resultado causou grande surpresa para os estatísticos da época, pois modelos como

$$X(T) = \int_0^T Y(s)\, ds, \tag{3.2}$$

[1] Uma das primeiras observações do fenômeno de dependência de longo prazo remonta ao ano 1895, quando o astrônomo Newcomb discutiu o fenômeno de dependência de longo prazo em conjuntos de dados astronômicos e denominou-o de "erros sistemáticos".

onde $Y(s)$ é um processo estocástico estacionário cujo somatório da função de covariância é convergente, possui um intervalo seqüencial assintoticamente proporcional a $T^{1/2}$. Isto levou estudiosos a concluírem, precipitadamente, que os fluxos dos rios não poderiam ser representados por processos estocásticos estacionários. O parâmetro H passou então a ser denominado de *parâmetro de Hurst.*

Anos mais tarde, Adelman [23] investigou também a presença de dependência de longo prazo em séries temporais econômicas. Os ciclos longos das séries foram investigados através de análise espectral, numa tentativa de se determinar se tais ciclos correspondiam a máximos espectrais em freqüências bem definidas. As experiências mostraram que o máximo espectral mais claramente identificado era, na verdade, localizado (tipicamente) muito próximo da menor freqüência observada, a saber, o inverso do tamanho T da série temporal em questão.

Em 1969, Benoit B. Mandelbrot [12] sugeriu então que a melhor maneira de se explicar a universalidade da forma do espectro dessas séries seria supondo-se que a densidade espectral $S(f)$ do processo gerador tende rapidamente ao infinito quando $f \to 0$. A exigência de que $S(f)$ tenda rapidamente para o infinito quando $f \to 0$ sugere que uma das formas possíveis para as funções $S(f)$ seja dada por

$$S(f) \sim f^{1-2H}, \qquad \text{para } \frac{1}{2} < H < 1. \tag{3.3}$$

Os processos para os quais $S(f) = f^{1-2H}$, com $\frac{1}{2} < H < 1$, foram denominados de **ruídos fracionários** ou **ruídos** $1/f$. Como estas funções possuem uma forma hiperbólica, Mandelbrot resolveu denominar a explicação dessa forma de espectro por "hipótese do espectro hiperbólico" ou "hipótese do espectro H". Mandelbrot denominou também esse fenômeno de "síndrome do espectro hiperbólico" ou "Joseph Effect", numa referência a uma passagem bíblica [12].

A hipótese do espectro H, em essência, é uma forma de se expressar o fato de que a interdependência do processo em questão é, teoricamente, **infinita** e que a sua função de autocorrelação exibe, tipicamente, um decaimento do tipo *hiperbólico*, ou seja, da forma

$$r(k) \sim k^{-\beta}, \; 0 < \beta < 1, \tag{3.4}$$

o que implica na não-somabilidade da função de autocorrelação.

Atualmente, considera-se a seguinte definição para um processo com dependência de longo prazo [3] :

Definição 3.1 : *Seja* $\{X(t), t \in \mathbb{R}\}$ *um processo estocástico estacionário no sentido amplo. Seja também a seqüência de variáveis aleatórias* X_n *tais que* $X(t_n) = X_n$. *A seqüência* $X_1, X_2, \ldots$ *de variáveis aleatórias é chamada de* ***dependente a longo prazo*** *se*

$$\sum_{n=1}^{\infty} Cov\{X_1, X_n\} = \infty. \tag{3.5}$$

Além da função de autocorrelação do tipo hiperbólica, uma outra característica de processos com dependência de longo prazo é o fato de que a estrutura de correlação do processo *acumulativo* é, assintoticamente, independente de escalonamentos no tempo [3]. Este fato está intimamente relacionado a um outro conceito matemático : o da *auto-similaridade.*

Antes, porém, de introduzirmos o conceito de processo estocástico auto-similar, é interessante expormos inicialmente o conceito de uma *função auto-similar* :

Definição 3.2 *Uma função $f(t)$ é dita* ***auto-similar,*** *com parâmetro de auto-similaridade H, se, para qualquer $\alpha > 0$,*

$$f(\alpha t) = \alpha^H f(t). \tag{3.6}$$

Assim, podemos definir um processo estocástico auto-similar.

Definição 3.3 *Um processo estocástico $X(t)$ é dito* ***estritamente auto-similar*** *com parâmetro de auto-similaridade H (parâmetro de Hurst) se, para qualquer $\alpha > 0$,*

$$\{X(\alpha t)\} \stackrel{P}{=} \{\alpha^H X(t)\} \tag{3.7}$$

onde $\stackrel{P}{=}$ indica que os processos $X(\alpha t)$ e $\alpha^H X(t)$ possuem as mesmas funções de distribuição de probabilidade conjuntas (de dimensão finita).

Em se tratando de características de segunda ordem, podemos estabelecer a seguinte definição :

Definição 3.4 *Um processo $X(t)$ é* ***auto-similar no sentido amplo*** *com parâmetro de auto-similaridade H se, para qualquer $\alpha > 0$, o processo $X(\alpha t)$ e $\alpha^H X(t)$ possuem as mesmas características de segunda ordem. Por outro lado, será dito* ***assintoticamente auto-similar de segunda ordem*** *se as características de segunda ordem de $X(\alpha t)$, normalizadas apropriadamente, convergem para as características de um processo auto-similar de segunda ordem quando $\alpha \to \infty$.*

Assim, dado que α é equivalente a um efeito de agregação, um processo que é (estritamente ou assintoticamente) auto-similar de segunda ordem, com $1/2 < H < 1$ é, essencialmente, equivalente a um processo com dependência de longo prazo. Ou seja, embora os conceitos de dependência de longo prazo e auto-similaridade sejam, em geral, dois conceitos distintos, eles caracterizarão o mesmo processo caso este seja auto-similar de segunda ordem. Em particular, se estivermos lidando com processos Gaussianos, podemos afirmar que se tratam de processos *estritamente* auto-similares.

Como conseqüência da auto-similaridade estatística, as trajetórias de processos com dependência de longo prazo são, tipicamente, *fractais* [21, 24]. Genericamente falando, fractais são objetos geométricos que possuem uma estrutura não-degenerada em todas as escalas temporais (ou espaciais).

Em geral, as trajetórias dos processos estocásticos são curvas unidimensionais bem comportadas no plano, as quais são bem caracterizadas pela sua "dimensão topológica". No entanto, processos estocásticos auto-similares possuem trajetórias amostrais tão irregulares que sua "dimensão efetiva" excede sua dimensão topológica unitária. Esta dimensão "efetiva" é usualmente denominada de *dimensão fractal* de uma curva[2]. No entanto, não existe um conceito único de dimensão fractal. Pelo contrário, existem na literatura diversas definições. Independente da definição utilizada, a dimensão fractal D de uma curva qualquer varia, tipicamente, entre $D = 1$ e $D = 2$. Para processos auto-similares, existe uma forte correspondência entre a dimensão fractal D e o parâmetro de auto-similaridade H do processo. Em particular, um aumento no parâmetro H corresponde a uma diminuição na dimensão D.

[2]Uma definição de dimensão fractal encontra-se no Apêndice A.

3.3 O Movimento Browniano Fracionário (fBm)

Antes de introduzirmos o *movimento Browniano fracionário* propriamente dito, é interessante definirmos o *movimento Browniano padrão*. Por *movimento Browniano* entende-se um processo estocástico $B(t)$ com incrementos Gaussianos tais que $B(t_2) - B(t_1)$ possui média zero e variância $|t_2 - t_1|$. Além disso, $B(t_2) - B(t_1)$ é independente de $B(t_4) - B(t_3)$ se os intervalos (t_1, t_2) e (t_3, t_4) não se sobrepõem [6].

A família de processos estocásticos conhecida hoje por *movimento Browniano fracionário* (fBm) foi proposta, inicialmente, em 1940 por Kolmogorov [25]. No entanto, a sua atual popularidade deve-se, sem dúvida alguma, aos trabalhos de Benoit B. Mandelbrot e John Van Ness [6], que juntos desenvolveram toda a teoria e a difundiram em diversos trabalhos [26, 27, 5].

Por movimento Browniano fracionário, Mandelbrot e Van Ness denominaram uma família de processos estocásticos *Gaussianos* construídos a partir de uma *média móvel* de incrementos infinitesimais sucessivos do *movimento Browniano puro*, isto é, de $dB(t)$, no qual os incrementos passados de $B(t)$ são ponderados pelo "kernel" $(t-s)^{H-1/2}$, onde $0 < H < 1$.

Uma definição mais conveniente, embora um pouco mais especializada, foi proposta por Barton e Poor [28] e será adotada aqui pelo simples fato de ser a mais utilizada atualmente.

Definição 3.5 *Seja* $\{B(t), t \in \mathbb{R}\}$ *o movimento Browniano puro e seja* H *um parâmetro real satisfazendo* $0 < H < 1$. *Denomina-se de* ***movimento Browniano fracionário de parâmetro*** **H** *o processo estocástico* $\{B_H(t), t \in \mathbb{R}\}$ *definido por :*

$$\begin{aligned} B_H(t) &= \frac{1}{\Gamma(H+1/2)} \left\{ \int_{-\infty}^{0} \left[|t-\tau|^{H-1/2} - |\tau|^{H-1/2} \right] dB(\tau) \right. \\ &\quad \left. + \int_{0}^{t} |t-\tau|^{H-1/2} dB(\tau) \right\}, \qquad t \in \mathbb{R}, \end{aligned} \tag{3.8}$$

onde, para $t < 0$, *a notação* $\int_0^t$ *deve ser interpretada como* $-\int_t^0$.

Da forma como está definido, $B_H(t)$ é um processo Gaussiano de média zero e com $B_H(0) = 0$. Notemos que, se $H = 1/2$, teremos

$$B_{1/2}(t) = B(t), \qquad t \in \mathbb{R},$$

ou seja, o fBm reduz-se ao movimento Browniano puro. Neste sentido, o movimento Browniano fracionário pode ser considerado como uma generalização do movimento Browniano puro. Para outros valores de H, $B_H(t)$ é denominado de uma *integral fracionária* de $B(t)$, no sentido de Weyl [29].

Pode-se mostrar [28, 30] que a função de autocorrelação de $B_H(t)$ é dada por :

$$E\{B_H(s)B_H(t)\} = R_{B_H}(s,t) = \frac{V_H}{2}\left(|s|^{2H} + |t|^{2H} - |t-s|^{2H} \right), \qquad s,t \in \mathbb{R}, \tag{3.9}$$

sendo que

$$V_H \triangleq \mathrm{Var}\left[B_H(1)\right] = \Gamma(1-2H)\frac{\cos(\pi H)}{\pi H}. \tag{3.10}$$

Daí resulta que

$$\mathrm{Var}\left[B_H(t)\right] = V_H |t|^{2H}. \tag{3.11}$$

Uma definição alternativa, baseada nas propriedades do fBm, foi dada por Norros [2],[11]. De acordo com Norros, um *movimento Browniano fracionário normalizado*, com parâmetro $H \in [\frac{1}{2}, 1)$, é um processo estocástico Z_t, $t \in (-\infty, \infty)$, caracterizado pelas seguintes propriedades :

1. Z_t possui incrementos independentes ;
2. $Z_0 = 0$, e $E[Z_t] = 0$ para todo t;
3. $E[Z_t^2] = |t|^{2H}$ para todo t;
4. Z_t possuit trajetórias contínuas ;
5. Z_t é um processo Gaussiano, isto é, todas as suas funções de distribuições conjuntas (finitas) são distribuições Gaussianas.

Das Eqs. (3.9) e (3.11) podemos concluir que o fBm é um processo estocástico *não estacionário*. Sendo um processo não-estacionário, a definição usual de densidade espectral de potência não se aplica ao movimento Browniano fracionário. Ao invés disso, é necessária a aplicação de alguma definição de densidade espectral de potência para processos estocásticos não-estacionários. Mais ainda, mesmo se tivermos uma definição bem estabelecida de espectro *dependente do tempo*, teremos de levar em conta o problema de sua medição, para que seja possível a sua relação com dados obtidos experimentalmente.

Uma abordagem bastante interessante foi dada por Patrick Flandrin [30] e serviu para esclarecer melhor o conceito de espectro do fBm. Flandrin dividiu a sua análise em duas partes distintas mas, ao mesmo tempo, complementares : uma análise de tempo-freqüência e uma análise de tempo-escala. A seguir, descreveremos essas duas análises.

Análise de Tempo-Freqüência do fBm

Embora não exista uma ferramenta única para análise espectral *dependente do tempo* para processos não-estacionários, uma série de trabalhos recentes tem enfatizado o uso da definição de espectro de Wigner-Ville [31]. Por definição, o espectro de Wigner-Ville (WVS) de um processo $X(t)$ com função de autocorrelação $R_X(t,s)$ é dado por

$$W_X(t,\omega) = \int_{-\infty}^{+\infty} R_X\left(t + \frac{\tau}{2}, t - \frac{\tau}{2}\right) e^{-i\omega\tau} d\tau. \tag{3.12}$$

Esta definição destaca-se por possuir várias propriedades interessantes e por generalizar, de um modo natural, o conceito de densidade espectral de potência para processos estacionários, de modo que, se o processo for estacionário, ela reduz-se ao conceito usual.

Aplicando-se a definição (3.12) ao fBm obtém-se, substituindo (3.9) em (3.12) [30],

$$W_{B_H}(t,\omega) = \left(1 - 2^{1-2H}\cos 2\omega t\right) \cdot \frac{1}{|\omega|^{2H+1}}. \tag{3.13}$$

A partir dessa descrição em tempo-freqüência podemos deduzir propriedades bastante interessantes. Se introduzirmos o fBm escalonado

$$B_{H;a}(t) = B_H(at), \quad a > 0, \tag{3.14}$$

e utilizarmos uma das propriedades do WVS, a saber,

$$W_{B_{H;a}}(t,\omega) = \frac{1}{a} W_{B_{H;a}}\left(at, \frac{\omega}{a}\right), \tag{3.15}$$

podemos concluir, a partir de (3.13), que

$$W_{B_{H;a}}(t,\omega) = W_{a^H B_H}(t,\omega), \tag{3.16}$$

ou seja, temos uma manifestação (de segunda-ordem) da auto-similaridade do fBm.

Como o WVS nos fornece um espectro *dependente do tempo*, é possível deduzirmos um espectro *médio* sobre um intervalo de tempo de comprimento T. Esta operação pode ser definida por

$$S_X(\omega;T) = \frac{1}{T}\int_0^T W_X(t,\omega)dt. \tag{3.17}$$

Aplicando esta definição ao fBm, obtemos de (3.13) :

$$S_X(\omega;T) = \left[1 - 2^{1-2H} \cdot \frac{\sin 2\omega T}{2\omega T}\right] \cdot \frac{1}{|\omega|^{2H+1}} \cdot \tag{3.18}$$

Como uma primeira conseqüência disto, temos que se T for escolhido de modo apropriado, as oscilações de (3.18) podem ser anuladas, ou seja,

$$S_{B_H}\left(\omega; k\frac{\pi}{2\omega}\right) = \frac{1}{|\omega|^{2H+1}}, \quad k = \cdots, -1, 0, +1, \cdots. \tag{3.19}$$

Do ponto de vista prático, a definição (3.17) pode ser visualizada como se fosse uma medição [31]. De fato, uma medição em uma dada freqüência exige um intervalo de análise cuja duração está relacionada ao inverso da mesma freqüência. Portanto, a segunda conseqüência de (3.18) é que, para uma dada freqüência ω_0,

$$T \gg \frac{1}{\omega_0} \Rightarrow S_{B_H}(\omega_0;T) \cong \frac{1}{|\omega_0|^{2H+1}}. \tag{3.20}$$

Daí,

$$S_{B_H}(\omega) = \lim_{T\to\infty} S_{B_H}(\omega;T) = \frac{1}{|\omega|^{2H+1}}.$$

Portanto, podemos expressar o comportamento espectral do fBm como o resultado de um processo de medição associado a um espectro dependente do tempo.

Notemos que

$$W_{B_H}(t,\omega) \geq 0 \Longleftrightarrow H \geq 1/2. \tag{3.21}$$

Portanto, para $1/2 \leq H \leq 1$ o fBm é um exemplo de processo não-estacionário com um WVS não-negativo. Isto nos fornece um modo alternativo de se pensar sobre o intervalo de valores do parâmetro H.

Caso exista uma derivada para o fBm, o comportamento espectral desta derivada será da forma (aplicando-se as propriedades da Transformada de Fourier) :

$$S_{B'_H}(\omega) = \frac{1}{|\omega|^{2H-1}}, \tag{3.22}$$

ou seja, para $1/2 < H < 1$ a derivada do fBm terá um comportamento espectral do tipo $1/f$.

Análise de Tempo-Escala do fBm

Uma segunda abordagem para se analisar a densidade espectral de potência do fBm é utilizar um método de tempo-escala para se examinar o comportamento do fBm em diferentes escalas de tempo. Uma maneira apropriada e simples de se realizar esta análise é através da poderosa *transformada wavelet* [32, 33, 21], a qual, para um processo $X(t)$, é definida por

$$T_X(t,a) = \frac{1}{\sqrt{a}} \int_{-\infty}^{+\infty} X(s) g\left(\frac{s-t}{a}\right) ds. \tag{3.23}$$

Em (3.23), $a > 0$ é o parâmetro de escala e $g(t)$ é uma função wavelet arbitrária (localizada), normalizada, de forma que sua transformada de Fourier $G(\omega)$ satisfaz $G(0) = 0$ e

$$\int_0^{+\infty} |G(\omega)|^2 \cdot \frac{d\omega}{\omega} = 1. \tag{3.24}$$

Dada uma wavelet básica $g(t)$, (3.23) proporciona uma análise de multiresolução ao filtrar o processo $X(t)$ com versões dilatadas e comprimidas de um único sistema de observação. No caso de um processo estocástico de média zero, a transformada (3.23) é, ela própria, um processo estocástico de média zero.

Se analisarmos a função de autocorrelação deste processo, dada por

$$R_X(t,s;a) = E\left\{T_X(t,a)\, T_X(s,a)\right\}, \tag{3.25}$$

obteremos (usando o fato de que $G(0) = 0$),

$$R_{B_H}(t,s;a) = -(V_H/2)\, a^{2H+1} \int_{-\infty}^{+\infty} \gamma_g\left(\tau - \frac{t-s}{a}\right) |\tau|^{2H} d\tau, \tag{3.26}$$

onde

$$\gamma_g(\tau) = \int_{-\infty}^{+\infty} g(\theta) g(\theta - \tau)\, d\theta. \tag{3.27}$$

A Eq. (3.26) exibe duas características interessantes. A primeira delas é que, quando analisado em uma dada escala, o fBm torna-se *estacionário*. A segunda característica é que o parâmetro relevante em (3.26) é a razão $(t-s)/a$, a qual é uma nova expressão da auto-similaridade no sentido de que as estatísticas de segunda ordem, relativas a um intervalo de tempo escalonado $b(t-s)$, $b>0$, são idênticas àquelas do intervalo não alterado, mas observado relativo a uma escala modificada a/b.

Como a função de autocorrelação (3.26) é uma função da diferença $(t-s)$, é possível se obter desta expressão uma densidade espectral de potência tomando-se a sua transformada de Fourier. O resultado é a densidade

$$S_{B_H}(\omega;a) = a\left|G\left(a\omega\right)\right|^2 \cdot \frac{1}{\left|\omega\right|^{2H+1}}\ , \tag{3.28}$$

a qual corresponde ao espectro do fBm quando visto através do filtro de escala nominal a.

Como o conteúdo espectral do fBm é caracterizado por (3.28) (relativo a cada escala), é possível se obter um espectro global adicionando-se as contribuições pertencentes a cada escala. Sendo assim, levando-se em conta a medida natural associada com a distribuição de energia da transformada wavelet na direção da escala, obtém-se o resultado

$$S_{B_H}(\omega) = \int_{-\infty}^{+\infty} S_{B_H}(\omega;a)\cdot\frac{da}{a} = \frac{1}{\left|\omega\right|^{2H+1}}. \tag{3.29}$$

É importante observarmos que em ambas as análises o comportamento espectral obtido foi o mesmo.

O teorema que segue motivou a introdução do fBm e apresenta duas importantes propriedades dos *incrementos* do fBm :

Teorema 3.1 *: Os incrementos do movimento Browniano fracionário são estacionários e auto-similares com parâmetro H.*

A prova deste teorema é dada através do seguinte corolário :

Corolário 3.1 (Lei T^H do desvio padrão) *: A variância dos incrementos do movimento Browniano fracionário é dada por :*

$$Var\left\{\left[B_H\left(t+T\right)-B_H\left(t\right)\right]\right\} = E\left\{\left[B_H\left(t+T\right)-B_H\left(t\right)\right]^2\right\} = \left|T\right|^{2H}V_H. \tag{3.30}$$

onde V_H é dado pela Eq. (3.10).

Este resultado é facilmente verificado aplicando-se a Eq. (3.9) e observando-se que o processo de incrementos possui média nula :

$$\begin{aligned}
\mathrm{Var}\left\{\left[B_H\left(t+T\right)-B_H\left(t\right)\right]\right\} &= E\left\{\left[B_H\left(t+T\right)-B_H\left(t\right)\right]^2\right\} \\
&= E\left\{B_H^2(t+T)\right\} - 2E\left\{B_H\left(t+T\right)B_H\left(t\right)\right\} + E\left\{B_H^2(t)\right\} \\
&= V_H\left|t\right|^{2H} + V_H\left|t+T\right|^{2H} - V_H\left(\left|t+T\right|^{2H} + \left|t\right|^{2H} - \left|T\right|^{2H}\right) \\
&= \left|T\right|^{2H}V_H.
\end{aligned} \tag{3.31}$$

Por este corolário, podemos concluir que os incrementos do fBm constituem um processo estocástico Gaussiano *estritamente estacionário,* cuja variância é função apenas do intervalo T entre as amostras. Além disso, percebemos por este corolário que

$$\mathrm{Var}\left\{\left[B_H\left(t+T\right)-B_H\left(t\right)\right]\right\}=\mathrm{Var}\left\{h^{-H}\left[B_H\left(t+hT\right)-B_H\left(t\right)\right]\right\}=\left|T\right|^{2H}V_H, \tag{3.32}$$

o que nos faz concluir que os incrementos do movimento Browniano fracionário $B_H(t)$ são *auto-similares*, ou seja,

$$\left\{\left[B_H\left(t+T\right)-B_H\left(t\right)\right]\right\}\overset{P}{=}\left\{h^{-H}\left[B_H\left(t+hT\right)-B_H\left(t\right)\right]\right\}. \tag{3.33}$$

Em termos das trajetórias, pode-se mostrar também que os movimentos Brownianos fracionários são estruturas **fractais**. Mais especificamente, é possível demonstrar [24, 21, 34] que funções amostrais de movimentos Brownianos fracionários, para $0 < H < 1$, possuem uma dimensão fractal (no sentido de Hausdorff-Besicovitch) dada por

$$D=2-H. \tag{3.34}$$

Como a Eq. (3.30) tende a zero quando $T \to 0$, $B_H(t)$ é contínuo no sentido médio quadrático. Mais ainda, pode-se provar que $B_H(t)$ possui trajetórias quase sempre contínuas. No entanto, $B_H(t)$ não é derivável no sentido médio quadrático e também não possui trajetórias amostrais deriváveis [6].

Por essa razão, o conceito de derivada do fBm, assim como do ruído Gaussiano branco, não está definido rigorosamente. O próprio Mandelbrot, em seu trabalho original, sugeriu uma aproximação para a derivada do fBm, denominada de **ruído Gaussiano fracionário (fGn)**, definido como um "processo estocástico generalizado" no sentido das distribuições de Schwartz [6, 35]. Posteriormente, Barton e Poor [28] formalizaram de maneira rigorosa a relação entre o ruído Gaussiano fracionário (fGn) e o movimento Browniano fracionário (fBm), definindo o fGn como uma derivada "generalizada" do fBm.

Antes porém de apresentarmos a expressão da derivada "generalizada" do fBm, analisaremos inicialmente um processo igualmente importante e que será o mais utilizado na prática. Utilizando-se a terminologia usada por Mandelbrot [6], definiremos o processo $B_H(t;\delta)$ como sendo o processo de *suavização* de $B_H(t)$, da forma :

$$\begin{aligned} B_H(t;\delta) &= \delta^{-1}\int_t^{t+\delta} B_H(s)\,ds \\ &= \int_{-\infty}^{\infty} B_H(s)\varphi_1(t-s)\,ds, \qquad \delta>0, \end{aligned} \tag{3.35}$$

onde

$$\varphi_1(t)=\begin{cases} \delta^{-1} & \text{se } 0\le t\le\delta \\ 0 & \text{caso contrário.} \end{cases}$$

O processo $B_H(t;\delta)$ possui derivada estacionária dada por

$$B'_H(t;\delta)=\frac{\left[B_H\left(t+\delta\right)-B_H\left(t\right)\right]}{\delta}, \tag{3.36}$$

ou seja, o processo $B'_H(t;\delta)$ é formado pelos *incrementos normalizados* de $B_H(t)$.

O processo $B'_H(t;\delta)$, por si só, é um processo estocástico bastante interessante. De (3.33) podemos perceber que, para qualquer $\delta > 0$, o processo $B'_H(t;\delta)$ é um processo Gaussiano *estacionário*, *auto-similar*, de *média nula* e com função de autocorrelação da forma :

$$\begin{aligned} R_{B'_{H,\delta}}(\tau;\delta) &= E\left\{B'_H(t;\delta)B'_H(t+\tau;\delta)\right\} \\ &= \frac{1}{2}V_H\delta^{2H-2}\left[\left(\frac{|\tau|}{\delta}+1\right)^{2H} - 2\left(\frac{|\tau|}{\delta}\right)^{2H} + \left|\frac{|\tau|}{\delta}-1\right|^{2H}\right], \qquad \tau \in \mathbb{R} \end{aligned} \tag{3.37}$$

Se $|\tau| \gg \delta$,

$$R_{B'_{H,\delta}}(\tau;\delta) \cong V_H H(2H-1)\,|\tau|^{2H-2}. \tag{3.38}$$

Da Eq. (3.38) pode-se mostrar que o processo $B'_H(t;\delta)$ é ergódico e que seu comportamento depende fortemente do valor de H. Além disso, podemos notar que $R_{B'_{H,\delta}}(\tau;\delta)$ possui o mesmo sinal que $H - 1/2$. De fato, o processo $B'_H(t;\delta)$ divide-se em três grandes famílias, correspondendo, respectivamente, a $0 < H < 1/2$, $H = 1/2$ e $1/2 < H < 1$.

Se $\frac{1}{2} < H < 1$, é possível se mostrar [6] que

$$\int_0^\infty R_{B'_{H,\delta}}(\tau;\delta)\,ds = \infty\,, \tag{3.39}$$

ou seja, $B'_H(t;\delta)$ é um processo com dependência de longo prazo. Por outro lado, se $0 < H < \frac{1}{2}$, pode-se mostrar que $B'_H(t;\delta)$ é um processo *negativamente correlacionado.*

Finalmente, se $1/2 < H < 1$, o processo $B'_H(t;\delta)$ é *estacionário*, *auto-similar* e com *dependência de longo prazo.*

Barton e Poor [28] estabeleceram de maneira rigorosa a relação entre o fBm e o fGn definindo o fGn como uma derivada "generalizada" do fBm. Fazendo-se então $\delta \to 0$ e definindo-se $H' = H - 1$, o movimento Browniano fracionário possui a **derivada generalizada** [21, 28] :

$$B'_H(t) = \lim_{\delta\to 0} B'_H(t;\delta) = \frac{1}{\Gamma(H'+1/2)}\int_{-\infty}^{t} |t-s|^{H'-1/2}\,dB(s) \tag{3.40}$$

a qual é estacionária e auto-similar com parâmetro H'.

Sendo um processo estacionário, a densidade espectral de potência do processo de incrementos pode ser obtida do modo usual. Como já afirmado anteriormente, se o processo for estacionário a Eq. (3.12) resume-se na definição usual de densidade espectral de potência. Analisando-se então o processo de incrementos

$$\delta B'_H(t;\delta) = B_H(t+\delta) - B_H(t)\,, \tag{3.41}$$

obtemos, de (3.12),

$$W_{\delta B'_H(t;\delta)}(t,\omega) = 4\left(\sin\frac{\omega\delta}{2}\right)^2 \cdot \frac{1}{|\omega|^{2H+1}}. \tag{3.42}$$

Portanto, a densidade espectral de $B'_H(t)$ pode ser obtida como o limite

$$\lim_{\delta\to 0} W_{B'_H(t;\delta)}(t,\omega) = \frac{1}{|\omega|^{2H-1}}, \tag{3.43}$$

o qual nos indica um comportamento do tipo $1/f$ para o fGn quando $1/2 < H < 1$.

3.4 O Modelo ARIMA fracionário

O modelo ARIMA fracionário introduzido por Hosking (1981) [36] teve como principal objetivo modelar uma versão em tempo discreto para o ruído Gaussiano fracionário proposto por Mandelbrot e Van Ness [6].

O próprio Mandelbrot propôs uma versão em tempo discreto para o ruído Gaussiano fracionário, o qual foi definido como um processo cuja função de autocorrelação é a mesma que a do processo de incrementos unitários $\Delta B_H(t) = B_H(t) - B_H(t-1)$ do movimento Browniano fracionário de tempo contínuo. Hosking, no entanto, procurou definir um processo baseado nas principais propriedades do fBm. Neste sentido, ele partiu da versão em tempo discreto do movimento Browniano padrão - o passeio aleatório ("random walk") - para obter o modelo desejado.

O passeio aleatório consiste em um processo de tempo discreto $\{x(n)\}$ definido por :

$$\nabla x(n) = (1-B)x(n) = e(n), \tag{3.44}$$

onde B é o operador de atraso, definido por $B[x(n)] = x(n-1)$, e $e(n)$ é uma seqüência aleatória i.i.d..

A *diferença primeira* do processo $\{x(n)\}$ é justamente o ruído branco de tempo discreto $e(n)$. Por analogia com a definição de ruído branco de tempo contínuo, define-se o ruído branco *diferenciado fracionariamente com parâmetro* H, como sendo a diferença fracionária de ordem $\left(\frac{1}{2} - H\right)$ do ruído branco de tempo discreto.

O operador *diferença fracionária* é definido por uma série binomial :

$$\nabla^d = (1-B)^d = \sum_{k=0}^{\infty} \binom{d}{k} (-B)^k = 1 - dB - \frac{1}{2}d(1-d)B^2 - \frac{1}{6}d(1-d)(2-d)B^3 - \ldots \tag{3.45}$$

Escreve-se então $d = H - \frac{1}{2}$ de forma que o ruído Gaussiano fracionário de tempo contínuo com parâmetro H passa a ter um processo análogo em tempo discreto, ou seja, o processo

$$x(n) = \nabla^{-d} e(n) \tag{3.46}$$

ou $\nabla^d x(n) = e(n)$, onde $\{e(n)\}$ é um ruído branco.

Pode-se mostrar [36] que quando $-\frac{1}{2} < d < \frac{1}{2}$, o processo $\{x(n)\}$ é estacionário e pode ser descrito por uma média-móvel da forma :

$$x(n) = \sum_{k=0}^{\infty} c(k)e(n-k), \tag{3.47}$$

onde

$$c(k) = \frac{\Gamma(k+d)}{\Gamma(k+1)\Gamma(d)}, \quad k \geq 1. \tag{3.48}$$

Quando $k \to \infty$, $c(k) \sim \Gamma(d)^{-1} k^{d-1}$, ou seja, os coeficientes decaem *hiperbolicamente*, ao invés de *exponencialmente,* como é o caso dos coeficientes dos tradicionais modelos ARMA (ou ARIMA $(p, 0, q)$).

A função de covariância do processo $\{x(n)\}$ (para $-\frac{1}{2} < d < \frac{1}{2}$), é dada por

$$\gamma(k) = E\{x(n)x(n-k)\} = \frac{(-1)^k(-2d)!}{(k-d)!(-k-d)!}, \tag{3.49}$$

e a função de autocorrelação é dada por

$$R(k) = \frac{d(1+d)\ldots(k-1+d)}{(1-d)(2-d)\ldots(k-d)}, \quad k = 1, 2, \ldots.$$

Quando $k \to \infty$,

$$R(k) \sim \frac{(-d)!}{(d-1)!}k^{2d-1}. \tag{3.50}$$

Assim, para $0 < d < \frac{1}{2}$ e k grande, o processo $\{x(n)\}$ possui uma função de autocorrelação positiva que decai hiperbolicamente, indicando a presença de *dependência de longo prazo*. Este fato também é comprovado através da sua função de densidade espectral de potência, dada por

$$S(\omega) = \left[2\sin\left(\frac{1}{2}\omega\right)\right]^{-2d} \quad \text{para } 0 < \omega \leq \pi. \tag{3.51}$$

Quando $\omega \to 0$, $S(\omega) \sim \omega^{-2d}$, de modo que próximo à origem o processo $\{x(n)\}$ possui um espectro hiperbólico.

Apesar de ser um modelo bastante poderoso para explicar o comportamento a longo prazo de séries auto-similares, o *ruído Gaussiano fracionário* apresenta algumas inconveniências, como a incapacidade de modelar as estruturas de correlação de curto prazo encontradas na prática.

Portanto, a fim de incorporar uma estrutura de curto prazo a seu modelo, Hosking propôs uma extensão do processo original ARIMA (p, d, q) de Box & Jenkins [37]. Definiu então o processo **ARIMA fracionário** (p, d, q), ou **FARIMA** (p, d, q), como sendo o processo $\{y(n)\}$ dado por

$$\Phi(B)\nabla^d y(n) = \Theta(B)e(n), \tag{3.52}$$

onde ∇^d é operador de diferenciação fracionária, $e(n)$ é um ruído branco, p e q são números inteiros, d é real e

$$\Phi(B) = 1 - \Phi_1 B - \ldots - \Phi_p B^p \tag{3.53}$$

e

$$\Theta(B) = 1 - \Theta_1 B - \ldots - \Theta_q B^q \tag{3.54}$$

são polinômios no operador de atraso B.

Definindo-se o processo dessa forma, podemos observar que o efeito do parâmetro d decai apenas hiperbolicamente à medida que a distância entre as observações cresce, enquanto

que os efeitos de Θ e Φ decaem exponencialmente. Assim, pode-se dizer que o parâmetro d descreve a estrutura de correlação de *longo prazo* e Θ e Φ descrevem a estrutura de *curto prazo* do processo em questão. De fato, o comportamento a longo prazo de um processo ARIMA fracionário (p, d, q) pode ser expresso por um processo ARIMA fracionário $(0, d, 0)$ (ou seja, o processo $x(n)$ descrito anteriormente), uma vez que, para observações distantes, os efeitos de Θ e Φ serão desprezíveis.

3.5 Processos de Recompensas Renovadas ("Renewal Reward Processes")

Um segundo "sintoma" comum em algumas séries temporais é a presença de observações que são extremamente grandes se comparadas com as demais observações da amostra. Elas ocorrem, tipicamente, em um pequeno conjunto da amostra, ou seja, em menor número, e variam muito de amostra para amostra.

Da mesma forma que os efeitos de persistência de longo prazo, a presença de tais "picos" faz com que diferentes amostras da mesma série pareçam distintas umas das outras. Mais ainda, as variâncias amostrais são enormemente influenciadas pelos valores destes "picos". Desta forma, impossibilita-se qualquer tentativa de estimação do valor verdadeiro da variância do processo.

Com o objetivo de considerar a variabilidade irregular das variâncias amostrais de tais processos (sem abandonar, para isso, a hipótese de estacionariedade), Mandelbrot propôs a "hipótese de variância infinita" ou "hipótese IFV" (Infinite Variance hypothesis). Segundo esta hipótese, a variância da população do processo gerador é **infinita** [12].

Dentre as densidades de probabilidade $p(x)$ que satisfazem a hipótese de variância infinita, a mais simples satisfaz, para $|x|$ grande, a relação

$$p(x) \sim |x|^{-\alpha-1}, \quad \text{para } 0 < \alpha < 2. \tag{3.55}$$

As funções de densidade de probabilidade que satisfazem esta relação são denominadas de *hiperbólicas* ou *lei de Pareto*. Na prática, a hipótese de variância infinita é equivalente à hipótese de uma densidade de probabilidade hiperbólica. Dentre as densidades de probabilidade do tipo hiperbólicas, a mais fácil de se lidar possui uma função característica da forma $\exp\left(-|\xi|^{\alpha}\right)$, com $0 < \alpha < 2$. Estas densidades foram consideradas inicialmente por Cauchy e são denominadas de "densidades estáveis simétricas". Densidades de probabilidade estáveis assimétricas foram encontradas por P. Levy. Essas densidades de probabilidade têm sido chamadas de "densidades de Pareto-Levy".

Além do *movimento Browniano fracionário (fBm)*, Mandelbrot propôs ainda um modelo de processo estocástico capaz de apresentar, simultaneamente, as duas "síndromes" teoricamente divergentes : a "síndrome do espectro hiperbólico" e a "síndrome da variância infinita".

Este modelo, proposto inicialmente para explicar os principais aspectos irregulares encontrados em algumas séries temporais econômicas, tem servido como base para uma explicação fenomenológica da contribuição do tráfego gerado por uma única fonte para a estrutura de dependência de longo prazo encontrada no tráfego agregado **total** da rede.

A construção do **processo de recompensas renovadas** foi baseada na construção original de variáveis aleatórias Gaussianas, introduzida originalmente por de Moivre. De acordo com essa construção, o ruído Gaussiano branco pode ser definido como um limite de agregações lineares da forma

$$X(t) = \mathrm{d} \lim_{M \to \infty} M^{-1/2} \sum_{m=1}^{M} W_m(t) \tag{3.56}$$

onde d lim indica o limite *em distribuição,* e $\{W_m(t)\}$ constitui uma seqüência de variáveis aleatórias independentes e identicamente distribuídas segundo uma distribuição binomial (ou qualquer outra distribuição com variância finita) [12].

Mandelbrot utilizou-se desta abordagem para construir um processo estocástico $X(t)$ *localmente Gaussiano*, no qual as propriedades almejadas são incorporadas nos elementos básicos $W_m(t)$. Ao agir dessa forma, ele esperava construir um processo $X(t)$ que preservasse as propriedades desejadas e, ao mesmo tempo, possuísse uma certa suavização proporcionada pela introdução da agregação linear.

Por um processo "localmente Gaussiano" entende-se qualquer processo estocástico estacionário $X(t)$ de instantes de tempo inteiros t, cujas distribuições N-dimensionais, ou seja, as distribuições dos vetores com coordenadas $X(t_1), X(t_2), \cdots, X(t_N)$, são "quase" Gaussianas quando N for pequeno, mas podem se tornar menos próximas da distribuição Gaussiana quando N for maior. Neste caso, subentende-se que uma definição apropriada para o conceito de distribuição "quase Gaussiana " tenha sido adotada [12].

Para a construção do *processo de recompensas renovadas* considera-se, inicialmente, uma seqüência $\{T_k\}$ de tempos discretos formando um *processo de renovação* ("renewal process"). Por um *processo de renovação* entende-se o processo estocástico formado pela seqüência de tempos discretos T_k onde os intervalos entre renovações $U_k = T_k - T_{k-1}$ são variáveis aleatórias independentes e identicamente distribuídas (i.i.d.). Além disso, considera-se que este processo de renovação seja estacionário.

Para se obter um processo de renovação estacionário devemos ter $E\{U\} < \infty$ e instantes de tempo T_k obtidos através do seguinte procedimento :

1. Inicialmente, descarta-se a variável aleatória U_0 e selecionam-se os pontos T_0 e T_{-1} de acordo com as seguintes regras :
   - Escolhe-se T_0 de forma a satisfazer

$$P[T_0 = u] = \frac{P[u+1]}{\sum_{s=1}^{\infty} P[s]} \tag{3.57}$$

     onde $P[u] = P[U \geq u]$.
   - Quando o valor de T_0 for determinado, T_{-1} é escolhido de forma a satisfazer

$$P[T_{-1} = u \mid T_0 = v] = \frac{P[u+v]}{P[v]}. \tag{3.58}$$

2. Na segunda etapa, os pontos restantes T_k são construídos da seguinte forma :

$$T_k = T_0 + \sum_{u=1}^{k} U_k, \qquad \text{para } k > 0;$$

$$T_k = T_{-1} - \sum_{k}^{-1} U_k \qquad \text{para } k < -1.$$

Seja agora uma seqüência de variáveis aleatórias identicamente distribuídas W_k com

$$E\{W_k\} = 0 \quad \text{e} \quad E\{W^2\} < \infty, \tag{3.59}$$

as quais são independentes umas das outras e de $\{T_k\}$. Define-se então o "processo de núcleo" $W(t)$ como sendo o processo estocástico estacionário igual a W_k para todo t satisfazendo $T_{k-1} < t \leq T_k$.

Mandelbrot mostrou que, se W_k possuir uma distribuição Gaussiana, a distribuição marginal de $W(t)$ será também Gaussiana. No entanto, a distribuição de $\{W(t), W(t+s)\}$ não será uma distribuição Gaussiana bidimensional, de modo que $W(t)$ não será um processo Gaussiano.

Pode-se mostrar [27] que a função de covariância de $W(t)$ é dada por

$$\text{Cov}\{W(t)W(t+n)\} = E\{W(t)W(t+n)\} = 1 - \left\{\sum_{s=1}^{n} P[s]\right\} \cdot \left\{\sum_{s=1}^{\infty} P[s]\right\}^{-1}. \tag{3.60}$$

Sabe-se [12] que uma condição suficiente para que um processo satisfaça a *hipótese do espectro H* é que a sua função de covariância seja do tipo hiperbólica, ou seja, da forma

$$C(\tau) \sim K(\tau)\,\tau^{1-\alpha} \quad \text{quando } \tau \to \infty \text{ e } 1 < \alpha < 2, \tag{3.61}$$

onde

$$\lim_{\tau \to \infty} K(h\tau)/K(\tau) = 1 \tag{3.62}$$

A fim de que o processo $\{W(t)\}$ satisfaça esta condição, devemos ter [12] :

$$P[U \geq u] \sim u^{-\alpha} K(u), \quad \text{quando } u \to \infty \tag{3.63}$$

onde $1 < \alpha < 2$. Portanto, se a função distribuição de probabilidade *complementar* dos intervalos entre-renovações for do tipo "heavy tail", como dado pela Eq. (3.61), o processo de núcleo $W(t)$ terá um espectro do tipo

$$S(f) \sim f^{\alpha-2} \tag{3.64}$$

próximo a $f = 0$, com $1 < \alpha < 2$.

Se analisarmos agora a distribuição de probabilidade do processo *acumulativo*

$$W^*(t) = \sum_{k=1}^{t} W(k), \tag{3.65}$$

é possível se mostrar [12] que, no caso em que $P[U \geq u] \sim u^{-\alpha} K(u)$,

$$\text{d} \lim_{T \to \infty} T^{-\alpha} W^*(T) = \Lambda_\alpha, \tag{3.66}$$

onde Λ_α é uma variável aleatória com distribuição de Pareto.

A partir destes resultados, Mandelbrot definiu o **processo de recompensas renovadas** $\Omega_M(t)$ como a soma de M *funções de núcleo* independentes e identicamente distribuídas, ou seja,

$$\Omega_M(t) = M^{-1/2} \sum_{m=1}^{M} W_m(t). \tag{3.67}$$

Construída dessa forma, a distribuição marginal de $\Omega_M(t)$ torna-se, rapidamente, muito próxima da distribuição Gaussiana quando $M \to \infty$. Além disso, a distribuição conjunta de $\Omega_M(t)$ e $\Omega_M(t+n)$ é também assintoticamente Gaussiana, bem como a distribuição conjunta, finita, K-dimensional do vetor $\{\Omega_M(t_1), \ldots, \Omega_M(t_K)\}$. Similarmente,

$$\Omega_M^*(t) = \sum_{s=1}^{t} \Omega_M(s) = M^{-1/2} \sum_{m=1}^{M} W_m^*(t) \tag{3.68}$$

tende a uma distribuição Gaussiana quando $M \to \infty$ e t permanece fixo.

No entanto, para M fixo (grande ou pequeno), é sempre possível tomarmos t suficientemente grande tal que $\Omega_M^*(t)$ pode ser aproximada por uma forma truncada de uma variável aleatória com distribuição de Pareto [12]. Assim, o processo $\Omega_M(t)$ é um processo estocástico *localmente Gaussiano* num sentido não-trivial. Ao se balancear o valor de M e o tamanho da amostra, podemos controlar a extensão sobre a qual o conceito de "local" permanece válido.

Em [12] e [38] é mostrado que, para T e M grandes, com $T \ll M$, o processo $\Omega_M^*(t)$, quando normalizado apropriadamente, comporta-se como o *movimento Browniano fracionário*. Além disso, nas mesmas condições, o processo de incrementos de $\Omega_M^*(t)$ comporta-se como o *ruído Gaussiano fracionário*.

Capítulo 4

A Auto-Similaridade do Tráfego Ethernet

4.1 Introdução

Em 1994, pesquisadores dos laboratórios Bellcore publicaram o que hoje é considerado o principal resultado dos últimos anos na área de redes de computadores. Em seu trabalho, eles demonstraram que o tráfego em redes do tipo LAN Ethernet é *estatisticamente auto-similar* e que nenhum dos modelos de tráfego comumente utilizados serve para caracterizar este comportamento que é *fractal* e com dependência de longo prazo [1].

Para chegarem a esta conclusão, eles realizaram uma análise estatística rigorosa de centenas de milhões de dados coletados entre 1989 e 1992 na rede LAN Ethernet dos laboratórios do Bellcore. Esses dados foram coletados por Leland e Wilson [39] através de medições de alta resolução. Ao realizar essas medições, Leland e Wilson detectaram a presença de surtos ao longo de diversas escalas de tempo ("burstiness"), o que posteriormente veio a ser confirmado (através de análises estatísticas) como uma manifestação de auto-similaridade. O comportamento fractal ou auto-similar do tráfego agregado de redes LAN Ethernet diferencia-se bastante do tráfego de telefonia convencional e de modelos comumente utilizados para se modelar tráfego de pacotes, como por exemplo, o modelo de Poisson (ou modelos relacionados, como os processos de Poisson Modulados por Markov [40]), modelos de trens de pacotes [41], modelos de fluxo de fluidos [42], etc. Nestes modelos, o tráfego agregado torna-se mais "suave" à medida em que o número de fontes de tráfego aumenta. Isto porque, em todos esses modelos, está implícita a hipótese de tráfego com *incrementos independentes*. Ou seja, todos esses modelos caracterizam-se pelo fato de possuírem incrementos independentes, ao contrário dos processos auto-similares, nos quais os incrementos são correlacionados. Sob a hipótese de incrementos independentes e pela Lei dos Grandes Números, é possível se justificar uma *uniformização* do tráfego agregado através da multiplexagem estatística, uma característica bastante desejada e que foi, inicialmente, admitida pelos idealizadores da Rede ATM [9].

Atualmente, diversos trabalhos têm se dedicado a analisar o comportamento estatístico de diversos tipos de tráfego, especialmente aqueles que serão, possivelmente, os principais tributários da rede ATM. Esta característica auto-similar já foi detectada também, por exemplo, em tráfegos com taxa de bit variável (VBR), especialmente o tráfego gerado por vídeo digital codificado via MPEG-2 [43, 8, 9] e tráfego de Internet [44].

Neste Capítulo, faremos uma exposição da auto-similaridade do tráfego nas redes locais Ethernet. Embora já existam estudos baseados em outros tipos de tráfego, como tráfego VBR

e tráfego em redes WAN ("Wide Area Networks") [45, 7], estamos, na verdade, interessados no problema mais básico de compreender o comportamento auto-similar em si, independente de sua origem (embora possa vir a existir diferenças significativas, daí os diversos estudos que estão sendo feitos nesta área).

Neste sentido, será exposta inicialmente uma "demonstração visual" do comportamento auto-similar do tráfego na rede Ethernet. Neste caso, o fenômeno da auto-similaridade pode ser muito facilmente sugerido se analisarmos amostras desse tipo de tráfego em diferentes escalas de tempo e compararmos com amostras, nas mesmas escalas, de tráfego convencional (com dependência de curto prazo).

A exposição da auto-similaridade do tráfego da rede Ethernet pode ser feita de uma forma mais rigorosa. Na seção 4.4 vamos expor os principais métodos de estimação do parâmetro H de uma série temporal qualquer. A escolha de métodos eficientes de estimação desse parâmetro tem recebido uma atenção especial por parte dos projetistas das redes ATM, uma vez que a estimação precisa do grau de auto-similaridade será de grande importância para a determinação do comportamento do tráfego e, conseqüentemente, para uma melhor alocação dos recursos da rede. Para fins de ilustração, apresentamos na seção 4.5 a aplicação de alguns dos métodos de estimação do parâmetro H para duas amostras de tráfego real.

Neste capítulo, apresentamos também um modelo de tráfego auto-similar baseado no movimento Browniano fracionário proposto por Norros [11]. A partir deste modelo, vamos expor o impacto da auto-similaridade no comportamento de filas e na estimação de parâmetros de tráfego, um tópico de muita importância que influirá, diretamente, na garantia da QoS oferecida aos usuários da rede ATM. Além disso, apresentaremos o problema da predição do fBm, estudado por Norros [2] e que tem relação direta com o tema principal desta dissertação.

Por último, na seção 4.7 discutiremos um modelo proposto recentemente para explicação da origem da auto-similaridade do tráfego em redes Ethernet, partindo-se da modelagem das fontes de tráfego *individuais*. Este modelo baseia-se no processo de recompensas renovadas exposto na seção 3.5 do Capítulo 3.

4.2 O Tráfego na Rede Ethernet

A rede local tipo Ethernet utiliza o protocolo CSMA-CD, "Carrier Sense Multiple Acess - Colision Detection", com velocidade de 10Mbps. Neste protocolo os terminais monitoram a rede para saber se há outra transmissão naquele instante. Caso afirmativo, o terminal continua "escutando" até que a rede esteja livre. A partir daí, inicia a sua transmissão. Se outro terminal da rede começa a transmitir simultaneamente ao primeiro, é detectada a colisão dos dados e ambos interrompem as transmissões abruptamente, permitindo assim, a economia de tempo e largura de faixa da rede. Após a colisão, ambos os terminais aguardam um período de tempo aleatório e tentam transmitir novamente, assumindo que nenhum outro terminal tenha iniciado a transmissão nesse intervalo de tempo [9]. As características da rede Ethernet constituem a base do padrão 802.3 do IEEE, o qual engloba também velocidades entre 1 a 10 Mbps em vários meios de transmissão. Por um abuso de linguagem, denominaremos o tráfego nas redes Ethernet por "tráfego Ethernet".

4.3 Uma Demonstração Visual da Auto-Similaridade do Tráfego Ethernet

É possível obtermos uma demonstração bastante simples do fenômeno da auto-similaridade do tráfego Ethernet observando-se o comportamento do tráfego em diferentes escalas de tempo e comparando-o com uma amostra de tráfego sintético produzido a partir de algum modelo tradicional.

A coluna da esquerda da Figura 4.1 mostra uma seqüência de gráficos correspondentes à contagem de pacotes (número de pacotes por unidade de tempo) para diferentes unidades de tempo de amostras de tráfego Ethernet. Em particular, esta figura refere-se às medidas de tráfego Ethernet realizadas em agosto de 1989 nos laboratórios do Bellcore. Começando com uma unidade de tempo de 100 s, cada gráfico subseqüente é obtido do anterior aumentando-se a resolução temporal por um fator de 10 e concentrando-se em um sub-intervalo escolhido aleatoriamente (indicado por uma região mais escura).

Podemos observar que todos os gráficos da coluna esquerda são intuitivamente muito "similares" entre si, isto é, o tráfego Ethernet parece o mesmo tanto em escalas maiores (min, h) quanto em escalas menores (s, ms). Em particular, podemos verificar a ausência de um comprimento natural para os surtos. Em todas as escalas de tempo, variando-se de milisegundos a minutos e horas, os surtos consistem de subperíodos de surtos separados por subperíodos com menos surtos. Esta característica invariante a escalas ou "auto-similar" do tráfego Ethernet é totalmente diferente do tráfego de telefonia convencional ou dos modelos estocásticos para tráfego de pacotes considerados na literatura. Estes últimos produzem gráficos de contagem de pacotes que são indistinguíveis do ruído branco após sofrerem alguma agregação, como mostram os gráficos da coluna direita da Fig. 4.1. Esta seqüência foi obtida do mesmo modo que na primeira seqüência, exceto pelo fato de que ela consiste em um tráfego sintético, gerado a partir de um processo de Poisson com as mesmas características (em termos de tamanho médio de pacotes e taxa de chegada). A Fig. 4.1 sugere um método muito simples para se inferir sobre a auto-similaridade de um processo qualquer.

4.4 Estimação do Parâmetro de Hurst (Parâmetro H)

A dependência de longo prazo, o decaimento lento das variâncias amostrais e o comportamento do tipo $1/f$ para a densidade espectral de potência correspondem a diferentes manifestações do fenômeno da auto-similaridade. Baseado nestes fatos, o problema de se testar e estimar o grau de auto-similaridade de um certo processo pode ser abordado de diferentes maneiras. Existem hoje na literatura diversos métodos para a estimação do parâmetro H (parâmetro de Hurst) de uma série temporal [46].

Este tema, como já abordado anteriormente, será de fundamental importância para o controle, gerenciamento e policiamento de tráfego em redes ATM, bem como para o Controle de Admissão (CAC) da rede. A correta estimação do parâmetro H permitirá uma melhor caracterização do tráfego, permitindo uma alocação mais racional dos recursos da rede e, conseqüentemente, uma maior garantia QoS contratada com os usuários. A seguir, apresentaremos alguns dos principais métodos de estimação do parâmetro H de uma série temporal.

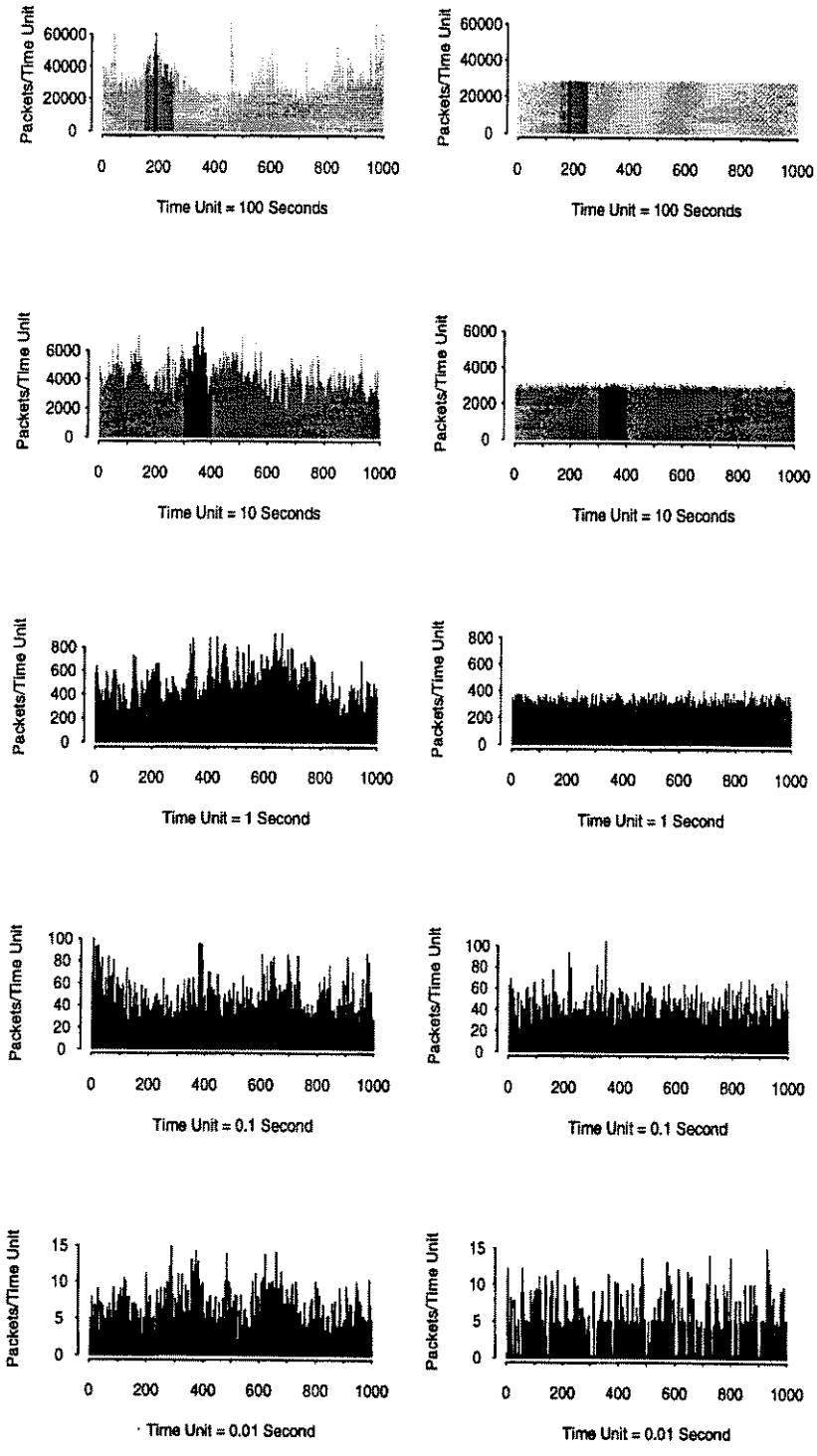

FIGURA 4.1: Coluna esquerda : tráfego Ethernet real (pacotes por unidade de tempo) em cinco escalas de tempo diferentes. Coluna direita : tráfego sintético gerado a partir de um modelo de tráfego Poissoniano nas mesmas escalas de tempo. (Figura extraída de [1]).

4.4.1 Método da Estatística R/S

O método da estatística R/S é um dos métodos mais conhecidos e empregados para estimação do parâmetro H de uma série temporal. Este método foi proposto inicialmente por Mandelbrot e Wallis [47] e está descrito detalhadamente em [48]. Dada uma série temporal $X = \{X_k,\ k = 1, 2, \ldots, n\}$ com soma parcial

$$Y(n) = \sum_{k=1}^{n} X_k \tag{4.1}$$

e variância amostral

$$S^2(n) = \left(\frac{1}{n}\right) \sum_{k=1}^{n} X_k^2 - \left(\frac{1}{n}\right)^2 Y^2(n), \tag{4.2}$$

a estatística R/S, ou *estatística do intervalo ajustado re-escalonado* ("rescaled adjusted range statistic"), é definida por :

$$\frac{R(n)}{S(n)} = \frac{1}{S(n)} \left\{ \max_{0 \le t \le n} \left[Y(t) - \frac{t}{n} Y(n) \right] - \min_{0 \le t \le n} \left[Y(t) - \frac{t}{n} Y(n) \right] \right\}. \tag{4.3}$$

No caso de séries temporais representativas de processos estocásticos auto-similares, esta estatística satisfaz, quando $n \to \infty$, a

$$E\{R(n)/S(n)\} \sim C_H n^H, \tag{4.4}$$

onde C_H é uma constante positiva, finita e independente de n, e onde H é o parâmetro de Hurst da série temporal.

Para se determinar o parâmetro H de uma série temporal de N amostras, subdivide-se esta série em K blocos não sobrepostos de comprimento $n = \lfloor N/K \rfloor$ (onde $\lfloor \cdot \rfloor$ indica o maior inteiro menor ou igual a N/K) e em seguida, para cada valor de n, calcula-se a estatística $R(k_i, n)/S(k_i, n)$ para os K blocos que se iniciam nos pontos

$$k_i = i \left(\frac{N}{K}\right) + 1, \quad i = 0, 1, 2, \ldots$$

tais que $k_i + n \le N$. Portanto, para cada valor de n obtém-se K diferentes estimativas de $R(n)/S(n)$. A análise gráfica da estatística R/S consiste em traçar o gráfico

$$\log\left[R(k_i, n)/S(k_i, n)\right] \times \log(n).$$

Este gráfico é comumente denominado de *diagrama pox*.

O *diagrama pox* apresenta, para pequenos valores de n, uma região de transição representativa da natureza de dependência de *curto prazo* da série temporal. À medida que o valor de n cresce, os pontos do gráfico tendem a flutuar em torno de uma reta de certa inclinação, terminando em uma região de poucos pontos, correspondendo a maiores valores de n. A fim de tornar a estimação mais confiável, descartam-se do gráfico os pontos correspondendo à região de transição e os pontos para os quais n é muito grande. Finalmente, tendo em vista a equação 4.4, a estimativa $\widehat{H}$ do parâmetro H pode ser obtida através da inclinação de uma reta ajustada ao gráfico via mínimos quadrados, por exemplo. Esta inclinação pode assumir valores entre 1/2 e 1. A principal vantagem da análise R/S é sua relativa robustez com relação a alterações da distribuição marginal do processo gerador.

4.4.2 Método da Variância Amostral

O método da variância amostral consiste em analisar o comportamento das variâncias amostrais das séries agregadas $X^{(m)} = \left(X_k^{(m)} : k = 1, 2, \ldots\right)$, obtidas a partir da série original $X = \{X_i,\ i = 1, 2, \ldots\}$, por divisão em blocos não sobrepostos de tamanho m e calcular a média de cada bloco. Ou seja, para cada $m = 1, 2, 3, \ldots$, a série agregada $X^{(m)}$ é dada por

$$X_k^{(m)} = \frac{1}{m} \sum_{i=(k-1)m+1}^{km} X_i, \qquad k = 1, 2, \ldots \tag{4.5}$$

Sabe-se que a variância dos processos agregados $X^{(m)}$ relativos a processos estocásticos auto-similares (de segunda ordem, pelo menos) decrescem lentamente de acordo com a relação

$$\mathrm{Var}\left(X^{(m)}\right) \sim \sigma^2 m^{\beta} \qquad \text{quando } m \to \infty, \tag{4.6}$$

com $\beta = 2H - 2 < 0$.

É possível então se obter uma estimativa $\widehat{\beta}$ de β (e conseqüentemente de H) a partir do gráfico de $\log\left[\widehat{\mathrm{Var}}\left(X^{(m)}\right)\right]$ versus $\log(m)$. Como $\widehat{\mathrm{Var}}\left(X^{(m)}\right)$ é uma estimativa da $\mathrm{Var}\left(X^{(m)}\right)$, os pontos resultantes no gráfico devem formar uma reta aproximada de inclinação β, onde $-1 < \beta < 0$. Na prática, esta inclinação é obtida ajustando-se uma reta por mínimos quadrados aos pontos do gráfico. Neste processo, descartam-se os pontos para os quais m é pequeno. Se a inclinação da reta pertencer ao intervalo $(-1, 0]$, o processo é auto-similar com parâmetro de Hurst (estimado) $\widehat{H} = 1 + \widehat{\beta}/2$. O caso $\widehat{\beta} = -1$ $(H = 1/2)$ corresponde ao ruído branco descorrelacionado.

4.4.3 Método dos Valores Absolutos das Séries Agregadas

Este método é muito similar ao método da variância amostral. Neste método os dados são repartidos da mesma maneira e as séries agregadas calculadas. Ao invés de se calcular a variância amostral, determina-se a soma dos valores absolutos dos termos das séries agregadas, ou seja, determina-se

$$S = \frac{1}{N/m} \sum_{k=1}^{N/m} \left| X_k^{(m)} \right|. \tag{4.7}$$

Traça-se então o gráfico do logaritmo desta estatística versus o logaritmo de m. Se a série original apresentar dependência de longo prazo, com parâmetro de Hurst H, o resultado será uma reta aproximada com declividade $H - 1$ [46].

4.4.4 Método de Higuchi

Este método, sugerido por Higuchi [49], envolve o cálculo do comprimento de uma trajetória e a determinação de sua dimensão fractal D. O método é muito similar ao método dos valores absolutos agregados das séries agregadas discutidas anteriormente. Segundo este processo, utiliza-se as somas parciais $Y(n) = \sum_{k=1}^{n} X_k$ da série original $\{X_k,\ k = 1, \ldots, N\}$ para se calcular o comprimento normalizado da curva, dado por :

$$L(m) = \frac{N-1}{m^3} \sum_{i=1}^{m} \left\lceil \frac{N-i}{m} \right\rceil^{-1} \sum_{k=1}^{\lceil (N-i)/m \rceil} \left| Y\left(i + km\right) - Y\left(i + (k-1)\, m\right) \right|, \tag{4.8}$$

onde N é o comprimento da série temporal, m é o tamanho dos blocos e $\lceil \cdot \rceil$ denota a função "menor inteiro maior ou igual a". Como $E\{L(m)\}$ aproxima-se, assintoticamente, a

$$E\{L(m)\} \sim C_H m^{-D},$$

onde $D = 2 - H$, um gráfico de $\log(L(m))$ versus $\log(m)$ deve produzir uma reta aproximada cuja inclinação $\widehat{D}$ é uma estimativa da dimensão fractal D da série original.

4.4.5 Método do Periodograma

Na análise do periodograma calcula-se inicialmente a densidade espectral de potência da série $X = \{X_i,\ i = 1, 2, \ldots\}$, dada por

$$S(\omega) = \frac{1}{2\pi N}\left|\sum_{j=1}^{N} X_j e^{ij\omega}\right|^2, \tag{4.9}$$

onde ω é a freqüência e N é o número de termos na série. Como $S(\omega)$ é um estimador da densidade espectral, uma série com dependência de longo prazo deve ter um periodograma proporcional a $|\omega|^{1-2H}$ próximo à origem. Portanto, se traçarmos o gráfico de $\log[S(\omega)]$ versus $\log(\omega)$, deveremos obter uma reta com declividade próxima a $1 - 2H$. Na prática, utilizam-se apenas as freqüências mais baixas, correspondendo a 10% do número total N de freqüências.

4.4.6 Método do Estimador de Whittle

O estimador de Whittle é baseado no método do periodograma. Ele envolve a função

$$Q(\eta) = \int_{-\pi}^{\pi} \frac{S(\omega)}{f(\omega;\eta)} d\omega, \tag{4.10}$$

onde $S(\omega)$ é o periodograma da série X e $f(\omega;\eta)$ é a densidade espectral de potência do processo X, *parametrizada* pela freqüência ω e pelo vetor η de parâmetros desconhecidos.

O estimador de Whittle nada mais é do que o valor de η que minimiza a função Q. Ao se lidar com um processo auto-similar, η corresponde exatamente ao parâmetro H. Este estimador possui a vantagem de fornecer intervalos de confiança para os valores do parâmetro H. No entanto, este método consome um tempo de processamento muito maior que os demais. Diferentemente dos métodos discutidos anteriormente, o estimador de Whittle é obtido através de um método não gráfico, assumindo-se o conhecimento *a priori* da forma parametrizada da densidade espectral de potência do processo X.

4.5 Análise do Tráfego Ethernet

Nesta seção ilustraremos alguns dos métodos estatísticos descritos anteriormente para a estimação do parâmetro H de séries temporais representativas de tráfego real. Para isto, utilizamos duas amostras de tráfego LAN Ethernet disponibilizadas pelo Bellcore, via ftp, no endereço `flash.bellcore.com` no diretório `pub/lan_traffic`. Estas amostras pertencem ao conjunto de medições realizadas por Leland e Wilson nos laboratórios Bellcore [39] e são as mesmas utilizadas por eles na elaboração da análise estatística que resultou na descoberta

do caráter auto-similar do tráfego Ethernet [1]. Os arquivos obtidos consistem nos dados coletados em Outubro de 1989 e referem-se aos tráfegos interno (circulado nos laboratórios do Bellcore) e externo (tráfego trocado entre o Bellcore e o ambiente externo, via Internet).

O arquivo pOct.TL (tráfego interno) contém o registro de um milhão de chegadas de pacotes no cabo principal da rede, monitoradas a partir das 11 :00 horas do dia 05/10/89, e correspondem a 1.759,62 segundos, aproximadamente. O arquivo OctExt.TL (tráfego externo) contém o registro de um milhão de chegadas de pacotes monitoradas a partir das 23 :46 horas do dia 03/10/89, correspondendo a 122.797,83 segundos, aproximadamente.

Ambos os arquivos estão em formato ASCII, sendo a chegada de cada pacote registrada em uma linha de 20 bytes. Cada linha possui dois registros : o primeiro registro refere-se ao instante, em segundos, no qual ocorre a chegada do pacote; e o segundo refere-se ao comprimento do pacote em bytes, excluindo-se o preâmbulo e o cabeçalho. Os bytes monitorados correspondem a pacotes completos, não incluindo fragmentos de colisões. A seguir, apresentamos os resultados da estimação do parâmetro H a partir de amostras de tráfego obtidas dos arquivos pOct.TL e OctExt.TL. Os métodos utilizados foram o do diagrama pox, o da variância amostral e o do periodograma. Os gráficos que serão apresentados foram todos obtidos neste trabalho a partir da implementação dos algoritmos correspondentes a cada método. O método do diagrama pox foi implementado utilizando-se a linguagem de programação disponível no Mathematica 3.0. No caso dos métodos da variância amostral e do periodograma, utilizamos o Matlab 5.0.

4.5.1 Diagrama "Pox" da Estatística R/S

As Figuras 4.2 e 4.3 exibem o diagrama pox calculado para os primeiros 10.000 pacotes correspondentes aos tráfegos externo e interno (arquivos OctExt.TL e pOct.TL). Descartando-se os pontos correspondendo à região de transição e os pontos para os quais n é muito grande, ajustamos uma reta via mínimos quadrados. A inclinação desta reta para o caso do tráfego externo foi de 0,8322, ou seja, o valor estimado de H é de $\widehat{H} = 0,8322$. No caso do tráfego interno, a inclinação calculada da reta foi de 0,6605, o que nos dá um parâmetro H estimado de $\widehat{H} = 0,6605$.

4.5.2 Método da Variância Amostral

As Figuras 4.4 e 4.5 mostram os gráficos do método das variâncias aplicado a 10.000 amostras de tráfego Ethernet. Para o tráfego externo (arquivo OctExt.TL), a reta aproximada possui uma inclinação de $-0,3297$, o que nos fornece uma estimação do parâmetro H de $0,8351$. No caso do tráfego interno, a reta aproximada apresenta uma inclinação de $-0,5552$, o que significa um parâmetro H de 0,7224.

4.5.3 Método do Periodograma

As Figuras 4.6 e 4.7 mostram os gráficos de $\log[S(\omega)]$ versus $\log(\omega)$. Para o tráfego externo, a reta aproximada possui uma inclinação de -0,6853, o que significa um parâmetro H estimado de 0,8426. Para o caso do tráfego interno, a inclinação da reta aproximada foi de -0,2959, de modo que o parâmetro de Hurst estimado foi de 0,6480.

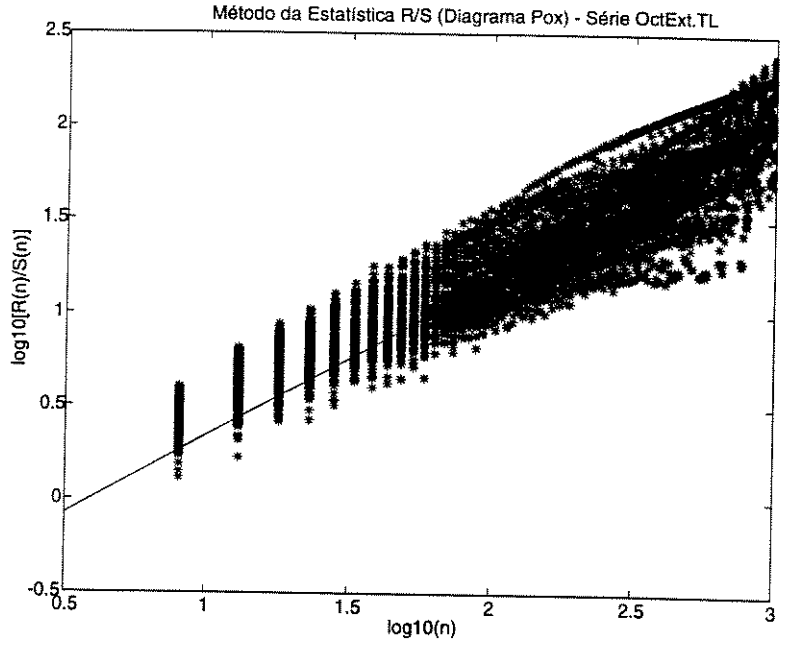

FIGURA 4.2: Diagrama Pox de 10.000 amostras de tráfego Ethernet (série OctExt.TL).

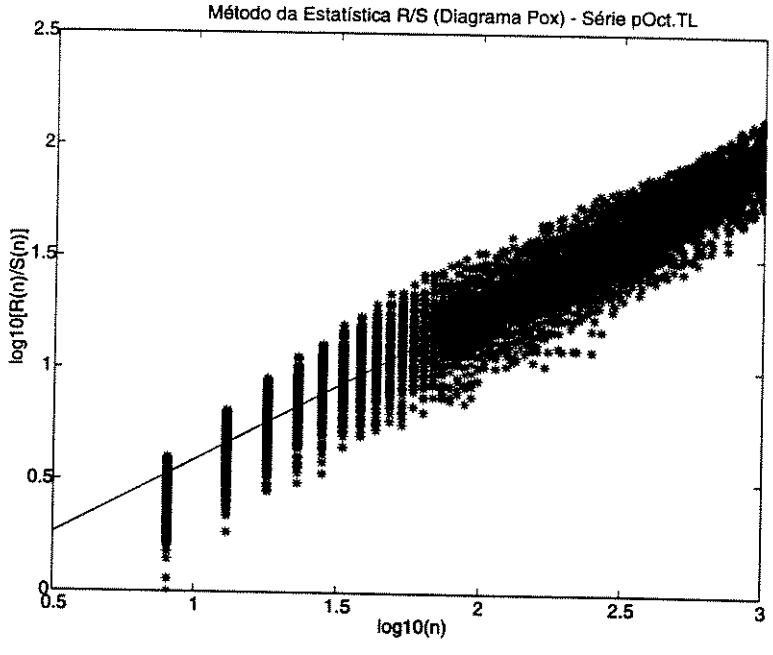

FIGURA 4.3: Diagrama Pox de 10.000 amostras de tráfego Ethernet (série pOct.TL).

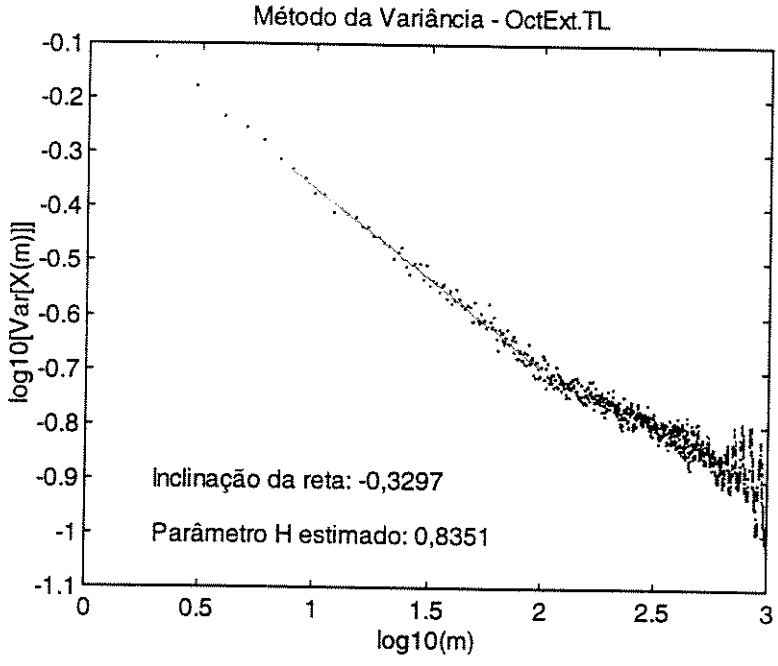

FIGURA 4.4: Método da variância aplicado a 10.000 amostras de tráfego Ethernet (arquivo OctExt.TL).

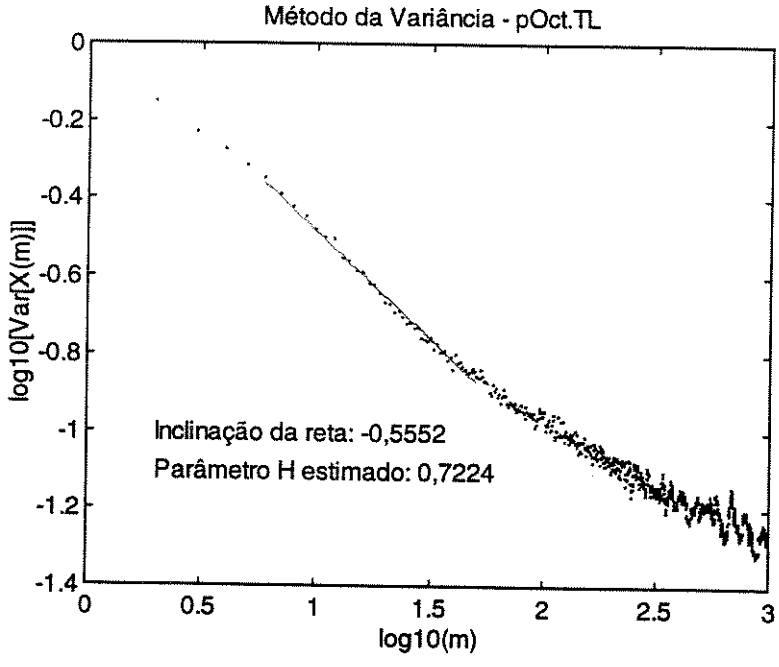

FIGURA 4.5: Método da variância aplicado a 10.000 amostras de tráfego Ethernet (arquivo pOct.TL).

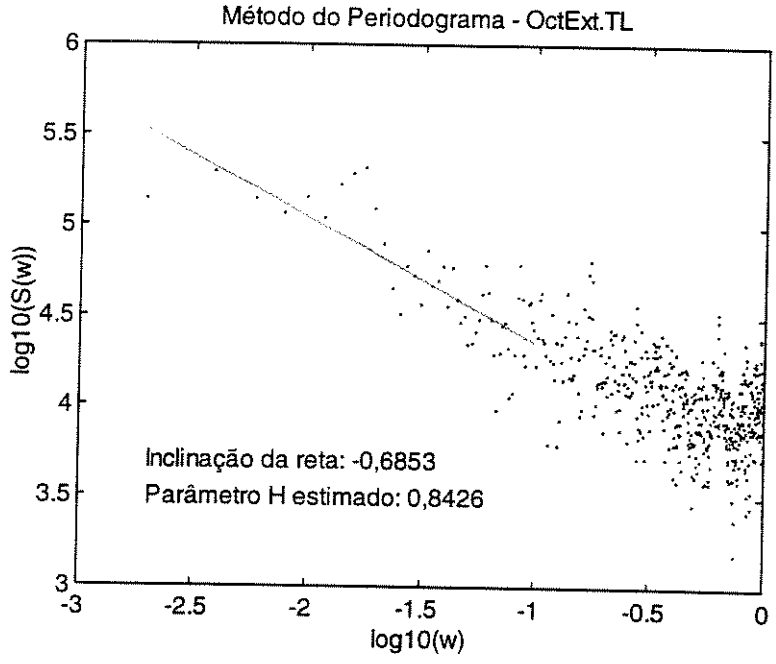

FIGURA 4.6: Método do periodograma aplicado a 10.000 amostras de tráfego Ethernet (arquivo OctExt.TL).

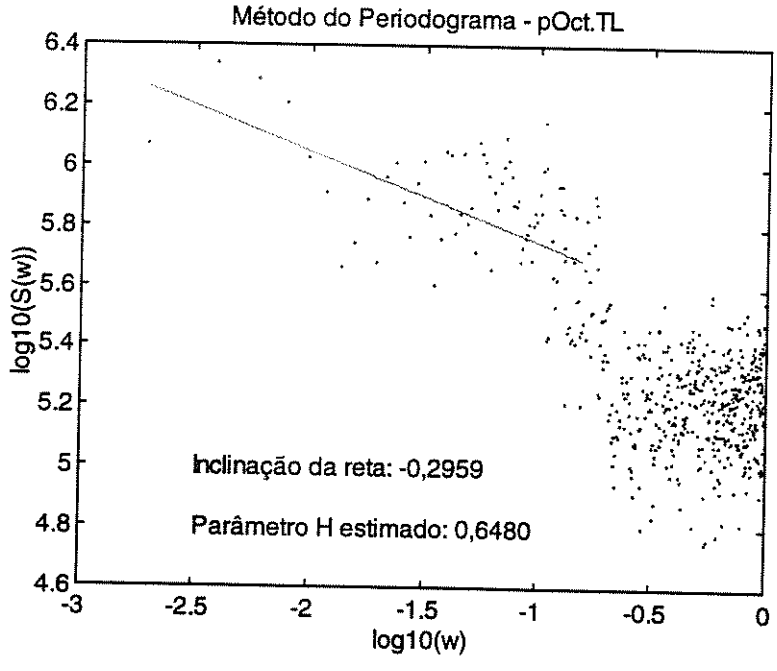

FIGURA 4.7: Método do periodograma aplicado a 10.000 amostras de tráfego Ethernet (arquivo pOct.TL).

4.6 O Tráfego Browniano Fracionário

A divulgação dos resultados dos laboratórios Bellcore tem trazido mudanças significativas nas áreas de projeto, controle e análise de redes de alta velocidade. Em particular, tem-se despendido um esforço muito grande na modelagem de tráfego auto-similar, uma vez que a escolha de um modelo apropriado é de fundamental importância para um melhor entendimento do comportamento do tráfego e, conseqüentemente, uma maior eficiência no uso dos recursos da rede. Uma escolha natural para modelo de tráfego auto-similar é o movimento Browniano fracionário (fBm) exposto no capítulo anterior. No entanto, apesar de ser um modelo matemático bastante apropriado para a representação de processos estocásticos auto-similares, a sua aplicação na modelagem de tráfego real precisa levar em consideração o fato essencial de que o tráfego de redes constitui um processo estocástico de média *não nula*. Uma alternativa bastante interessante, e que contorna este problema, foi proposta por Norros [11], o qual introduziu o *tráfego Browniano fracionário*, descrito a seguir.

No desenvolvimento que segue o tráfego é considerado um "fluido" contínuo, ao invés de um processo de tempo discreto. Denomina-se *processo de contagem* (ou de *acumulação*) o processo estocástico A_t que representa a quantidade de tráfego (bits, bytes, células, etc.) que chega a um certo elemento da rede (um "buffer", por exemplo) no intervalo de tempo $[0, t)$. Em particular, teremos $A_0 = 0$. Considera-se também o *processo de incrementos* do processo A_t, definido por

$$A(s,t) = A_t - A_s, \tag{4.11}$$

o qual representa o tráfego ocorrido no intervalo de tempo $[s, t)$. Norros definiu o *tráfego Browniano fracionário* como sendo o processo estocástico A_t, dado por

$$A_t = mt + \sqrt{am}Z_t\ , \qquad t \in (-\infty, \infty) \tag{4.12}$$

onde Z_t é um fBm com parâmetro H $(1/2 \leq H < 1)$, $m > 0$ é a taxa média de geração de tráfego e $a > 0$ é um coeficiente de variância.

Notemos que, neste modelo, Norros permitiu a possibilidade de incrementos *negativos*, contradizendo o fato de que um processo acumulativo real é sempre crescente. No entanto, este modelo permite a obtenção de importantes resultados, os quais iremos reproduzir mais adiante. Outra observação digna de nota é o fato de que Z_t é um objeto matemático *adimensional* da mesma forma que seu parâmetro t. Portanto, de uma maneira rigorosa, deveríamos escrever $Z(t/t_u)$, onde t é o tempo físico e t_u é a unidade de tempo físico utilizada. Isto ajuda, em particular, a se evitar confusão no momento em que for alterada a unidade de tempo. No entanto, manteremos a notação original para efeito de simplificação. Ainda com relação às dimensões, se o tráfego for medido em bits e o tempo em segundos, o parâmetro a possui dimensão bit·s.

A superposição $A_t = \sum_{i=1}^{K} A_t^i$ de K tráfegos Browniano fracionários *independentes* com mesmo parâmetro H, mas m_i's e a_i's próprios, é também um tráfego Browniano fracionário com mesmo parâmetro H e parâmetros m e a dados por [11] :

$$m = \sum_{i=1}^{K} m_i, \qquad a = \sum_{i=1}^{K} \frac{m_i}{m} a_i.$$

Se os parâmetros a_i forem idênticos, a superposição terá o mesmo parâmetro a dos tráfegos originais. Isto motiva a introdução do fator $\sqrt{m}$ em (4.12) : os papéis dos três parâmetros

podem ser separados tais que H e a caracterizam a "qualidade" do tráfego enquanto que a taxa média m caracteriza a sua "quantidade". Uma interpretação heurística do parâmetro a pode ser obtida se relacionarmos o tráfego Browniano fracionário com algum outro modelo de tráfego com dependência de longo prazo, por exemplo, o modelo de *fluido de surtos* proposto por Kosten [50]. Segundo este modelo, o tráfego é produzido por uma superposição infinita de fontes do tipo "on/off", as quais produzem surtos que começam segundo um processo de Poisson com parâmetro λ, chegam cada um com uma taxa r e possuem comprimentos independentes B_n com distribuição conjunta $F(x) = P(B \leq x)$. Se denotarmos o número de surtos que chegam no instante t por K_t, a taxa de chegada de fluidos no instante t será $R_t = rK_t$ (também denominado de "processo de taxa"). O processo de chegada acumulativo $A_t = \int_0^t R_s ds$ será dependente de longo prazo se o comprimento do surto B_n possuir variância infinita [11, 3]. Comparando o modelo de fluido de surtos com o tráfego Browniano fracionário, Norros obteve a seguinte expressão para o parâmetro a :

$$a = r^{2H-1} x_0^{-2H} \left(2x_0 E(U \wedge x_0) - E(U \wedge x_0)^2\right), \tag{4.13}$$

onde $x_0 = rt$ e U é uma variável aleatória cuja distribuição de probabilidade corresponde à distribuição de "tempo de vida residual" da variável aleatória B,ou seja,

$$P(U \leq u) = \frac{1}{b} \int_0^u (1 - F(t))\, \mathrm{d}t,$$

onde $b = E[B]$. A partir dessa expressão Norros sugeriu a seguinte interpretação do parâmetro a (à luz do modelo de fluido de surtos) :

- O parâmetro a é independente de λ;
- Quando $H = 1/2$, a é o índice de dispersão (variância sobre a média) de A_t para t grande, e é independente da taxa r. Notemos que, neste caso, $H = 1/2$ significa que todos os surtos são pequenos e portanto, em uma escala de tempo maior, suas chegadas podem ser consideradas instantâneas;
- Quando $H > 1/2$, a é o produto de três fatores; o primeiro fator r^{2H-1} fornece uma expressão para a dependência de a em relação à taxa r de transmissão do surto; os dois primeiros fatores juntos mostram que a.depende de H através de um fator da forma $const^{2H}$. O último fator depende, principalmente, da distribuição truncada de B para "surtos pequenos".

Baseados neste modelo, importantes resultados teóricos têm sido obtidos. Descreveremos a seguir o impacto do tráfego Browniano fracionário em problemas relacionados ao controle de tráfego em redes de faixa larga. Em particular, serão expostos resultados recentes sobre o comportamento de filas na presença de tráfego Browniano fracionário, a estimação de parâmetros de tráfego e a predição do tráfego Browniano fracionário.

4.6.1 Análise de Filas com Tráfego Browniano Fracionário

Um importante aspecto a se considerar na análise de desempenho de redes de computadores é o comportamento do preenchimento das filas (ou "buffers") da rede. Para isto, é necessário, inicialmente, a modelagem apropriada do tráfego em questão. O comportamento das filas quando submetidas a tráfego Browniano fracionário ainda não está completamente

entendido. Existe apenas uma fórmula aproximada para a distribuição do comprimento da fila, obtida por Norros [11] e que reproduziremos a seguir.

Assumiremos então que um tráfego Browniano fracionário com parâmetros m, a e H é oferecido em um enlace com capacidade $C > m$, no qual existe um "buffer" com capacidade de armazenamento ilimitada. Podemos definir então o processo estocástico estacionário denominado de *armazenamento Browniano fracionário* com parâmetros m, a e H e capacidade de saída $C > m$, como o processo X_t definido por

$$X_t = \sup_{s \leq t} \left(A(s,t) - (t-s) \cdot C \right), \quad t \in (-\infty, \infty) \tag{4.14}$$

onde

$$A(s,t) = A_t - A_s = (t-s) \cdot m + \sqrt{ma}\,(Z_t - Z_s) \tag{4.15}$$

e Z_t é o movimento Browniano fracionário com parâmetro H.

A estacionariedade de X_t segue da estacionariedade dos incrementos de A_t. O processo X_t é finito e não-negativo, embora o processo de chegada possa possuir incrementos negativos. A distribuição estacionária do processo X_t obedece a uma lei de escala que é facilmente deduzida da auto-similaridade de Z_t [11].

Em princípio, desejamos que a probabilidade de que o tráfego exceda um certo nível x seja menor do que um valor ϵ pequeno. (O valor x neste caso é o substituto para o tamanho do "buffer" no modelo de armazenagem infinita). Se a maior probabilidade de saturação permitida for ϵ, então a equação

$$\epsilon = P\left(X > x\right) \tag{4.16}$$

valerá na carga máxima permitida (onde carga é definida por $\rho = \frac{m}{C}$). A Eq. (4.16) pode ser escrita como uma relação entre os parâmetros de projeto x (espaço no buffer) e C (capacidade do enlace) e os parâmetros de tráfego m, a e H no limite crítico [11] :

$$(C-m)\,(ma)^{-1/(2H)}\, x^{(1-H)/H} = f^{-1}(\epsilon), \tag{4.17}$$

onde a função

$$f(y) = P\left(\sup_{t \geq 0} \left(Z_t - yt\right) > 1 \right) \tag{4.18}$$

depende de H mas não de m, a, C ou de x. No caso de $H = 1/2$, a Eq. (4.17) reduz-se a

$$\frac{1-\rho}{\rho a} \cdot x = \text{const.} \tag{4.19}$$

sendo $\rho = m/C$. Neste caso, para $\rho \simeq 1$, podemos afirmar que para diminuirmos a capacidade livre relativa $(1-\rho)$ pela metade, teremos que aproximadamente dobrar o tamanho da capacidade de armazenamento x.

No caso de $H > 1/2$ a situação é diferente. Escrevendo-se a Eq. (4.17) na forma apropriada para dimensionamento de buffer, teremos [11] :

$$x = f^{-1}\left(\epsilon\right) a^{1/(2(1-H))} C^{(2H-1)/(2(1-H))} \frac{\rho^{1/(2(1-H))}}{(1-\rho)^{H/(1-H)}} \cdot \tag{4.20}$$

Podemos perceber que, neste caso, quando H é alto, um aumento substancial na utilização da capacidade livre (digamos metade novamente) requer um aumento muito maior no espaço de armazenamento. A Eq. (4.17) também pode ser escrita como uma fórmula para alocação de banda [11] :

$$C = m + f^{-1}(\epsilon)\, a^{1/(2H)} x^{-(1-H)/H} m^{1/(2H)}. \tag{4.21}$$

Até agora, nenhuma fórmula explícita para a distribuição do *armazenamento Browniano fracionário* parece ser conhecida. Norros [11] tratou o problema da distribuição de X_t através da determinação do limitante inferior

$$P\{X_t > x\} \geq \max_{t \geq 0} P\{A_t > Ct + x\}, \tag{4.22}$$

onde o valor máximo do segundo membro é obtido em $t = Hx/((1-H)(C-m))$ [11]. Explicitamente, tem-se que

$$P\{X_t > x\} \geq \overline{\Phi}\left(\frac{(C-m)^H x^{1-H}}{\kappa(H)\sqrt{am}}\right), \tag{4.23}$$

onde $\kappa(H) = H^H (1-H)^{1-H}$ e $\overline{\Phi}(y) = P\{Z_1 > y\}$ é a função distribuição residual da distribuição Gaussiana padrão. Utilizando-se a aproximação

$$\overline{\Phi}(y) \sim \exp\left(-y^2/2\right), \tag{4.24}$$

obtém-se a expressão

$$P\{X_t > x\} \sim \exp\left(-\frac{(C-m)^{2H}}{2\kappa^2(H)\, am} x^{2-2H}\right). \tag{4.25}$$

Assim, a distribuição de X_t pode ser aproximada por uma distribuição de Weibull. No caso de $H = 1/2$, a distribuição de Weibull reduz-se a uma distribuição exponencial da forma

$$\max_{t \geq 0} P\{A_t > Ct + x\} \sim \exp\left(-2\frac{1-\rho}{\rho a} \cdot x\right). \tag{4.26}$$

Expressando-se (4.25) em função de C, temos que $P\{X_t > x\} = \epsilon$ é alcançada aproximadamente quando

$$C = m + \left(\kappa(H)\sqrt{-2\ln\epsilon}\right)^{1/H} a^{1/(2H)} x^{-(1-H)/H} m^{1/(2H)}. \tag{4.27}$$

Norros estudou o uso da Eq. (4.27) como uma fórmula prática para dimensionamento da capacidade do enlace considerando a sua sensibilidade com relação aos parâmetros a e H. Utilizando os valores $m = 2$ Mb/s e $\epsilon = 10^{-3}$, ele chegou à conclusão de que quando o "buffer" é pequeno (no caso, 100 kbytes) a capacidade de canal depende muito menos de H do que quando o "buffer" é grande (no caso, 1 Mbyte). Além disso, para os mesmos valores dos parâmetros m e ϵ, ele chegou também à conclusão de que é muito difícil o preenchimento completo do "buffer" grande (1 Mbyte) quando o tráfego possui dependência de curto prazo.

4.6.2 Predição do Tráfego Browniano Fracionário

Uma conseqüência das correlações positivas no tráfego com dependência de longo prazo é, em princípio, a possibilidade de predição a curto prazo do tráfego em várias escalas de tempo. Neste sentido, é interessante estudarmos como tal predição deve ser feita e o que pode ser obtido a partir dela. No entanto, Norros [11, 2] propôs uma solução bastante interessante para o problema teórico da predição do movimento Browniano fracionário[1]. Neste estudo, assume-se que os parâmetros m, a e H tenham sido estimados confiavelmente durante um longo período de observação. O problema de predição a curto prazo de tráfego, digamos, predizer o valor de $A(t, t+h)$ baseado na observação de $A(t-T, s)$ para $s \in [t-T, t]$, reduz-se ao problema de se encontrar expressões mais ou menos explícitas para os preditores (esperanças condicionais)

$$\widehat{Z}_{h,T} = E\left\{Z_h \mid Z_s,\ s \in (-T, 0]\right\}, \qquad h > 0, T \in (0, \infty], \tag{4.28}$$

que são soluções ótimas sob o critério do erro médio quadrático.

Como o movimento Browniano fracionário é um processo Gaussiano, é natural que representemos os preditores como integrais sobre a parte observada do processo, na forma

$$\widehat{Z}_{h,T} = \int_{-T}^{0} g_T(h, t)\, dZ_t, \tag{4.29}$$

onde $g_T(h,t)$ é uma função de peso apropriada. Quando o integrando é uma função determinística suave, uma integral com relação a Z_t pode ser definida simplesmente como um limite de somas de Riemann em L^2 [2]. Foi mostrado em [2] que para cada $h > 0$ e $T \in (0, \infty]$, a representação (4.29) é válida para

$$g_T(h, -t) = \frac{\sin\left(\pi\left(H - \frac{1}{2}\right)\right)}{\pi} t^{-H+\frac{1}{2}} (T-t)^{-H+\frac{1}{2}} \int_0^h \frac{\sigma^{H-\frac{1}{2}} (\sigma + T)^{H-\frac{1}{2}}}{\sigma + t} d\sigma, \tag{4.30}$$

para $T < \infty$, e

$$g_\infty(h, -t) = \frac{\sin\left(\pi\left(H - \frac{1}{2}\right)\right)}{\pi} t^{-H+\frac{1}{2}} \int_0^h \frac{\sigma^{H-\frac{1}{2}}}{\sigma + t} d\sigma. \tag{4.31}$$

Rogers [51] obteve, no caso $T = \infty$, uma outra fórmula onde a integral é uma integral usual e Z_t aparece como um fator no integrando :

$$E\left\{Z_h \mid Z_s,\ s \le 0\right\} = -\frac{\sin\left(\pi\left(H - \frac{1}{2}\right)\right)}{\pi} \int_{-\infty}^{0} \frac{\left|\frac{t}{h}\right|^{-H-\frac{1}{2}}}{h + |t|} Z_t dt. \tag{4.32}$$

É interessante notarmos que a função de peso em (4.30) tende ao infinito tanto na origem quanto em $-T$, quando T for finito. Intuitivamente, a não-monoticidade pode ser entendida como se as "testemunhas mais próximas" para o passado não observado possuíssem peso especial. A Fig. 4.8 mostra o comportamento da função de peso (4.30).

[1] Nesta dissertação estudamos a predição do processo de *incrementos* de tráfego Ethernet real, o qual poderia ser modelado por um ruído Gaussiano fracionário (fGn).

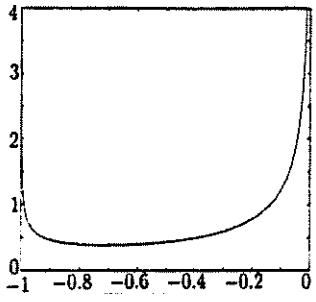

FIGURA 4.8: Função de peso $g_1(1,t)$ para $H = 0.9$. (Figura extraída de [2]).

A variância do preditor $E\{Z_a \mid Z_s,\ s \in (-T,0]\}$ é uma medida concreta da previsibilidade estatística do tráfego Browniano fracionário. Foi mostrado em [2] que

$$\mathrm{Var}\{E[Z_h \mid Z_s,\ s \in (-T,0]]\} = \mathrm{Var}\{Z_h\}\, H \int_0^{T/h} g_{T/h}(1,-s)\left((1+s)^{2H-1} - s^{2H-1}\right) ds. \tag{4.33}$$

Mais ainda, para $T = \infty$ existe uma expressão em forma fechada em termos da função gama :

$$\mathrm{Var}\{E[Z_h \mid Z_s,\ s \le 0]\} = \mathrm{Var}\{Z_h\}\left(1 - \frac{\sin\left(\pi\left(H-\frac{1}{2}\right)\right)}{\pi\left(H-\frac{1}{2}\right)} \frac{\Gamma\left(\frac{3}{2}-H\right)^2}{\Gamma(2-2H)}\right). \tag{4.34}$$

A variância relativa do erro $\mathrm{Var}\left(Z_h - \widehat{Z}_{h,\infty}\right)/\mathrm{Var}(Z_h)$ está mostrada na Figura 4.9 em função de H. Notemos que, como uma conseqüência da auto-similaridade do tráfego, esta quantidade é independente de h. Pode-se notar também que a força preditiva do passado não é muito alta, a menos que H seja muito grande (próximo a 1). O passado antes de 0 resolve metade da variância de Z_h quando H está próximo de 0.85, o qual é um valor típico para tráfego Ethernet, como mostrado no início deste capítulo.

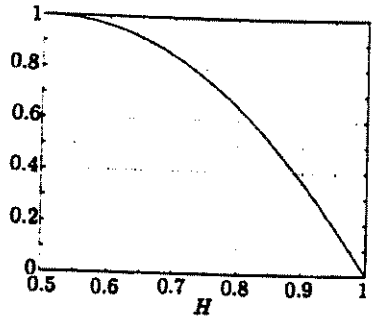

FIGURA 4.9: Variância relativa do erro como função do parâmetro de auto-similaridade H (Figura extraída de [2]).

A Figura 4.10 mostra a variância relativa do erro $\mathrm{Var}\left(Z_1 - \widehat{Z}_{1,T}\right)/\mathrm{Var}(Z_1)$ como uma função de T com $H = 0.9$.

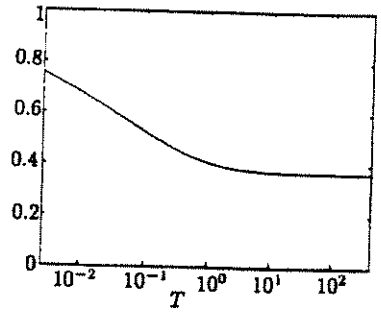

FIGURA 4.10: Variância relativa do erro como função de T, para $H = 0.9$ (Figura extraída de [2]).

Podemos notar desta figura que, para a predição de Z_h, existe pouca diferença se conhecemos Z em $(-h, 0)$ ou em $(-\infty, 0)$. Desta forma, é possível se estabelecer duas regras básicas para a previsibilidade a curto prazo do tráfego Browniano fracionário :

- O passado antes de t resolve aproximadamente metade da variância de qualquer valor futuro A_u, $u > t$;
- Deve-se predizer o próximo segundo baseado no último segundo, o próximo minuto com o último minuto, etc.

4.6.3 Estimação de Parâmetros de Tráfego

Como a garantia da qualidade de serviço oferecida pela rede ATM dependerá, essencialmente, de uma estimação eficiente das características do tráfego, o estudo e caracterização de estimadores de tráfego com dependência de longo prazo é de fundamental importância. Como será mostrado nesta seção, os estimadores para tráfego com dependência de longo prazo são polarizados e convergem mais lentamente do que no caso tradicional de dependência de curto prazo.

Estimação da Taxa Média

A taxa média de uma seqüência de tráfego estacionário sobre um intervalo $[0, T]$ é usualmente estimada através da média amostral

$$\widehat{m}_T = \frac{A_T}{T},$$

onde A_T é o processo de contagem do tráfego de chegada. No entanto, na presença de dependência de longo prazo, a estimação da taxa média pode se tornar tão imprecisa que a noção de taxa média deve ser interpretada de um modo totalmente diferente, por exemplo, daquele utilizado para a taxa média de um processo de Poisson. Para ilustrarmos isso, vamos assumir que o tráfego em questão obedece ao modelo de tráfego proposto por Norros, a saber,

$$A_T = mt + \sqrt{ma}Z_T. \tag{4.35}$$

A variância de $\widehat{m}_T$ é dada então por

$$\text{Var}\{\widehat{m}_T\} = maT^{2H-2}, \tag{4.36}$$

de modo que para diminuirmos o desvio padrão relativo para um valor abaixo de um dado ϵ, devemos ter [3] :

$$T > \left(\frac{1}{\epsilon}\sqrt{\frac{a}{m}}\right)^{1/(1-H)} . \tag{4.37}$$

O segundo membro de (4.37) cresce rapidamente com o valor de H. Por exemplo, se tivermos $m = 2$ Mbit/s e $a = 200$ kbit·s (valores típicos em redes LAN atuais), o tempo de observação exigido para uma precisão modesta de um desvio padrão de 10% seria de $10^{1/(2-2H)}$ segundos, o que significa aproximadamente cinco minutos para um tráfego com $H = 0.8$ e mais de 27 horas para um tráfego com $H = 0.9$. Neste último caso, até o significado de taxa média torna-se problemático, uma vez que as variações não-estacionárias ao longo do dia devem ser levadas em consideração.

A Imprecisão das Variâncias Amostrais

Novamente, vamos nos restringir ao caso do modelo de tráfego de Norros e assumir que a taxa média m é estimada pela média amostral, do modo usual. Em princípio, os dois parâmetros restantes (a e H) podem ser determinados pela Eq. (4.36). No entanto, a estimação destes parâmetros torna-se cada vez mais difícil à medida que H cresce, pois as variâncias amostrais tornam-se fortemente polarizadas e de convergência muito lenta. Para entendermos isso melhor, consideremos inicialmente a polarização. Subtraindo o "drift" estimado de A_t, obtemos o processo

$$B_t = \sqrt{ma}\left(Z_t - \frac{t}{T}Z_T\right), \qquad t \in [0, T] .$$

O fato dos incrementos de B_t possuírem variâncias menores do que as do processo A torna-se mais importante à medida que H cresce. Para simplificar a notação, vamos escolher $T = 1$ e considerar as médias dos quadrados dos incrementos

$$S_n^2 = \frac{1}{n}\sum_{k=1}^{n}\left(B_{k/n} - B_{(k-1)/n}\right)^2 . \tag{4.38}$$

O valor esperado $E\{S_n^2\}$ será então

$$\begin{aligned} E\{S_n^2\} &= \frac{1}{n}\sum_{k=1}^{n}\mathrm{Var}\{B_{k/n} - B_{(k-1)/n}\} = \frac{ma}{n}\sum_{k=1}^{n}\left(n^{-2H} - c(n,k)n^{-2}\right) \\ &= \mathrm{Var}\{A_{1/n}\} - man^{-2}, \end{aligned}$$

onde a soma dos números

$$c(n,k) = 2n\mathrm{Cov}\{Z_{k/n} - Z_{(k-1)/n}, Z_1\} - 1, \quad k = 1, \ldots, n,$$

é igual a um. Para $H \in (1/2, 1)$, as variâncias dos intervalos subseqüentes diferem umas das outras. Os números $c(n,k)$ dependem, essencialmente, apenas de k/n e apresentam a maior variação quando H está em torno de 0.72. O valor de H representa um impacto muito grande na polarização de S_n^2 como um estimador de $\mathrm{Var}\{A_{1/n}\}$. Vamos então considerar a convergência da variância amostral em direção à variância real. Podemos simplificar o problema

assumindo que a taxa média é conhecida exatamente e considerando, assim, os estimadores não-polarizados da variância

$$\widehat{S}_n^2 = \frac{1}{n}\sum_{k=1}^{n}\left(Z(i) - Z(i-1)\right)^2. \tag{4.39}$$

Utilizando o fato de que para um vetor Gaussiano bi-dimensional (X, Y), centrado, nós temos

$$E\left[Y \mid X\right] = \frac{\mathrm{Cov}\left\{X, Y\right\}}{\mathrm{Var}\left\{X\right\}} X$$

e

$$\mathrm{Var}\left[Y \mid X\right] = \left(1 - \frac{\mathrm{Cov}\left\{X, Y\right\}^2}{\mathrm{Var}\left\{X\right\}\mathrm{Var}\left\{Y\right\}}\right)\mathrm{Var}\left\{Y\right\},$$

teremos

$$\begin{aligned}\mathrm{Var}\left\{\widehat{S}_n^2\right\} &= \frac{2}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n}\mathrm{Cov}\left\{Z(i) - Z(i-1), Z(j) - Z(j-1)\right\}^2 \\ &= \frac{2}{n} + \frac{4}{n^2}\sum_{i=2}^{n}\sum_{j=1}^{i-1}\mathrm{Cov}\left\{Z(i) - Z(i-1), Z(j) - Z(j-1)\right\}^2.\end{aligned} \tag{4.40}$$

Para n grande, temos [3]

$$\mathrm{Cov}\left\{Z(i) - Z(i-1), Z(j) - Z(j-1)\right\} \approx H\left(2H-1\right)\left|i-j\right|^{2H-2}. \tag{4.41}$$

Aproximando o somatório duplo em (4.40) pela integral

$$H(2H-1)\int_1^n dt \int_1^t ds\left|t-s\right|^{4H-4},$$

obtém-se a aproximação

$$\mathrm{Var}\left\{\widehat{S}_n^2\right\} \approx \frac{2}{n} + \frac{4}{n^2}\frac{\left(H\left(2H-1\right)\right)^2}{\left|4H-3\right|}\left|\frac{n^{4H-2}-1}{4H-2} - n + 1\right|, \quad H \neq 3/4. \tag{4.42}$$

(para o valor limite $H = 3/4$ a fórmula correspondente é um pouco mais complicada, contendo uma expressão logaritmica). Assim, o valor $H = 3/4$ separa a região onde o desvio padrão relativo decresce segundo $n^{-1/2}$ de uma região onde a convergência é ainda mais lenta e depende de H :

$$\sqrt{\mathrm{Var}\left\{\widehat{S}_n^2\right\}} \sim \begin{cases} n^{-1/2}, & H < 3/4 \\ n^{-2(1-H)}, & H > 3/4. \end{cases} \tag{4.43}$$

Novamente, existe uma grande diferença entre os casos $H = 0.8$ e $H = 0.9$: o tamanho requerido da amostra para um desvio padrão de 10% é menos do que 10^4 para $H = 0.8$, mas maior do que 10^6 para $H = 0.9$.

4.7 A Natureza do Tráfego Gerado por Fontes Isoladas

O trabalho desenvolvido nos laboratórios do Bellcore mostrou que, independente do lugar e do momento em que foram realizadas as medições de tráfego Ethernet, o tráfego coletado **agregado** era intrinsicamente auto-similar, sendo seu grau de auto-similaridade dependente da carga total da rede.

Recentemente, os mesmos autores do trabalho que relatou esta descoberta proporcionaram uma resposta para esta questão. Através de novos resultados matemáticos e a sua subseqüente validação através de novas análises estatísticas, expuseram uma explicação simples e plausível para a auto-similaridade observada no tráfego de pacotes LAN Ethernet em termos da natureza do tráfego gerado pelas fontes **individuais** ou pares fonte-destino que formam a seqüência de tráfego [52].

Desenvolvendo uma abordagem baseada no modelo de recompensas renovadas proposto por Mandelbrot [12], eles mostraram que a superposição de várias fontes ON/OFF independentes e identicamente distribuídas (i.i.d.) e estritamente alternantes, cada uma das quais exibindo o chamado "Efeito de Noah", resulta em um tráfego agregado auto-similar.

Ao apresentar esse resultado no contexto de fontes ON/OFF, eles identificaram o Efeito de Noah como o principal "divisor de águas" entre a modelagem tradicional de tráfego e a modelagem de tráfego auto-similar. A modelagem de tráfego tradicional, quando abordada no contexto de modelos de fontes ON/OFF, sem exceção, assume funções de distribuição com variância **finita** para os períodos ON e OFF (por exemplo, a distribuição exponencial e a distribuição geométrica). Estas hipóteses limitam drasticamente as atividades ON/OFF de uma fonte individual e, como resultado, a superposição de muitas fontes comporta-se como um ruído branco, no sentido que a seqüência de tráfego agregado está isenta de qualquer correlação significativa, exceto, possivelmente, alguma correlação de curto prazo. A seguir, descreveremos os resultados obtidos por Taqqu et al. [52].

Consideremos inicialmente uma única fonte e concentremo-nos na série temporal binária estacionária $\{W(t),\ t \geq 0\}$ gerada por ela. Se $W(t) = 1$, significa que existe um pacote no instante t; se $W(t) = 0$ significa que não há pacote no instante t . Se interpretarmos $W(t)$ como a "recompensa" no instante t, teremos uma recompensa de valor igual a "um" ao longo do período ON, e uma recompensa de valor igual a "zero" ao longo do período OFF. Os comprimentos dos períodos ON são independentes e identicamente distribuídos, o mesmo acontecendo para os períodos OFF. Além disso, os comprimentos dos períodos ON e OFF também são independentes entre si e podem ter distribuições diferentes. Um período OFF sempre segue um período ON, e é o par de períodos ON e OFF que define um período *entre-recompensas*. Dito de outra forma, os períodos ON e OFF são *estritamente alternantes*[2].

Vamos supor agora que existam M fontes i.i.d.. Como cada fonte envia sua própria seqüência de trem de pacotes, ela possui sua própria seqüência de recompensas $\left\{W^{(m)}(t),\ t \geq 0\right\}$. A superposição (ou contagem acumulativa de pacotes) no instante t é $\sum_{m=1}^{M} W^{(m)}(t)$. Reescalonando por um fator de T, consideremos

$$W_M^*(Tt) = \int_0^{Tt} \left(\sum_{m=1}^{M} W^{(m)}(u) \right) du,$$

como sendo o processo de contagem de pacotes no intervalo $[0, Tt]$.

[2]Este conceito é uma particularização do processo de recompensas renovadas de Mandelbrot. Neste modelo, mais genérico, é permitida a ocorrência de "recompensas" consecutivas de mesmo valor.

Estamos interessados no comportamento estatístico do processo estocástico $\{W_M^*(Tt), t \geq 0\}$ para valores grandes de M e T. Este comportamento depende das distribuições dos períodos ON e OFF, os únicos elementos que não foram especificados. Desejamos então escolher estas distribuições de tal forma que, quando $M \to \infty$ e $T \to \infty$, $\{W_M^*(Tt), t \geq 0\}$, normalizado adequadamente, torna-se o processo $\{\sigma_{\lim} B_H(t), t \geq 0\}$, onde $\sigma_{\lim}$ é uma constante positiva finita e $B_H(t)$ é o movimento Browniano fracionário. A fim de especificarmos as distribuições dos períodos ON e OFF, consideremos a notação seguinte.

Assumiremos que, quando $x \to \infty$,

$$\text{ou } F_{1c}(x) \sim l_1 x^{-\alpha_1} L_1(x) \text{ com } 1 < \alpha_1 < 2 \text{ ou } \sigma_1^2 < \infty,$$

e

$$\text{ou } F_{2c}(x) \sim l_2 x^{-\alpha_2} L_2(x) \text{ com } 1 < \alpha_2 < 2 \text{ ou } \sigma_2^2 < \infty,$$

onde $l_j > 0$ é uma constante, $F_{jc}(x)$ é a função distribuição *complementar* do período j, σ_j^2 é a variância do período j e $L_j > 0$ é uma função com variação lenta no infinito, isto é,

$$\lim_{x \to \infty} \frac{L_j(tx)}{L_j(x)} = 1, \tag{4.44}$$

para qualquer $t > 0$. Adotaremos também que $j = 1$ corresponde ao período ON e $j = 0$ será associado ao período OFF. Um ponto a ser observado é o fato de que a média μ_j é sempre finita, mas a variância σ_j^2 será infinita quando $\alpha_j < 2$. Por exemplo, $F_j(x)$ (a função de distribuição) pode ser uma distribuição de Pareto, isto é, $F_{jc}(x) = K^{\alpha_j} x^{-\alpha_j}$ para $x \geq K > 0$, $1 < \alpha_j < 2$ e igual a 0 para $x < K$, ou pode ser exponencial. Como as distribuições F_1 e F_2 podem ser diferentes, podemos ter, por exemplo, uma distribuição com variância finita e outra com variância infinita.

Antes de introduzirmos o resultado principal, precisamos introduzir alguma notação. Quando $1 < \alpha_j < 2$, faremos $a_j = l_j (\Gamma(2 - \alpha_j)) / (\alpha_j - 1)$. Quando $\sigma_j^2 < \infty$, faremos $\alpha_j = 2$, $L_j \equiv 1$ e $a_j = \sigma_j^2$. Os fatores de normalização e as constantes no teorema que segue dependem de que

$$\Lambda = \lim_{t \to \infty} t^{\alpha_2 - \alpha_1} \frac{L_1(t)}{L_2(t)}$$

seja finito, igual a 0 ou infinito. Se $0 < \Lambda < \infty$, fazemos $\alpha_{\min} = \alpha_1 = \alpha_2$,

$$\sigma_{\lim}^2 = \frac{2(\mu_2^2 a_1 \Lambda + \mu_1^2 a_2)}{(\mu_1 + \mu_2)^3 \Gamma(4 - \alpha_{\min})} \text{ , e } L = L_2;$$

Se, por outro lado, $\Lambda = 0$ ou $\Lambda = \infty$, fazemos

$$\sigma_{\lim}^2 = \frac{2\mu_{\max}^2 a_{\min}}{(\mu_1 + \mu_2)^3 \Gamma(4 - \alpha_{\min})} \text{ , e } L = L_{\min},$$

onde min é o índice 1 se $\Lambda = \infty$ (por exemplo, se $\alpha_1 < \alpha_2$) e o índice 2 se $\Lambda = 0$. O mesmo acontecendo para max. Sob as condições estipuladas acima, o seguinte teorema é válido :

Teorema 4.1 *Para M e T grandes, o processo acumulativo agregado $\{W_M^*(Tt),\ t \geq 0\}$ comporta-se estatisticamente da forma*

$$TM\frac{\mu_1}{\mu_1+\mu_2}t + T^H\sqrt{L(T)\,M}\sigma_{\lim}B_H(t)$$

onde $H = (3-\alpha_{\min})/2$ e $\sigma_{\lim}$ é dado como acima. Mais precisamente,

$$\mathfrak{L}\lim_{T\to\infty}\mathfrak{L}\lim_{M\to\infty}T^{-H}L^{-1/2}(T)M^{-1/2}\left(W_M^*(Tt)-TM\frac{\mu_1}{\mu_1+\mu_2}t\right)=\sigma_{\lim}B_H(t),$$

onde $\mathfrak{L}\lim$ significa convergência no sentido das distribuições de dimensões finitas.

Heuristicamente, o teorema acima afirma que o termo $TM\left(\mu_1/\left(\mu_1+\mu_2\right)\right)t$ fornece a principal contribuição quando M e T forem grandes. As flutuações em torno deste valor são dadas pelo movimento Browniano fracionário $\sigma_{\lim}B_H(t)$ escalonado pelo fator $T^HL(T)^{1/2}M^{1/2}$. Notemos que se $1<\alpha_{\min}<2$, teremos $1/2<H<1$, isto é, teremos dependência de longo prazo. Assim, o fator fundamental para se obter um $H>1/2$ é a característica "heavy-tail"

$$F_{jc}(x)\sim l_jx^{-\alpha_j}L_j(x),\ \text{quando } x\to\infty,\ 1<\alpha_j<2,$$

para o período ON ou para o período OFF.

Capítulo 5

Predição de Séries Temporais

5.1 Introdução

Este capítulo tem como finalidade apresentar o problema geral da *predição de séries temporais*. Por uma *série temporal* entende-se qualquer conjunto de observações ordenadas no tempo, como por exemplo, os valores diários de temperatura em uma cidade, as cotações de mercado da Bolsa de Valores, um sinal elétrico amostrado no tempo, o número de pacotes de dados que chegam sucessivamente a um nó de uma rede, etc.

Com esse objetivo, apresentaremos a predição de séries temporais sob dois contextos : a predição *linear* e a predição *não-linear*. No contexto da *predição linear*, apresentaremos a teoria já bem estabelecida da filtragem de Wiener, que servirá como ponto de partida para a compreensão do algoritmo LMS, um algoritmo adaptativo que será utilizado nas nossas predições de tráfego auto-similar. No contexto da *predição não-linear*, enfocaremos o uso das *redes neurais artificiais* como ferramentas poderosas no problema de predição. Particularmente, vamos expor duas arquiteturas de rede : o *perceptron multicamadas FIR* e a *rede de funções de base radiais (RBF)*.

No caso das redes de funções de base radiais (RBF), vamos expor ainda um modelo de preditor adaptativo denominado de "rede RBF dinâmica" proposto em [13]. Além disso, relacionado ao problema da estimação dos parâmetros das funções de base (no caso Gaussianas), vamos expor o algoritmo EM de Dempster [14, 53] aplicado ao problema de estimação de máxima-verossimilhança de parâmetros de misturas de densidades Gaussianas. Como contribuição, propomos uma modificação deste algoritmo através da introdução de uma função de penalidade, com o intuito de evitar a ocorrência de singularidades durante o processo de estimação. A partir deste algoritmo modificado propomos um algoritmo EM "adaptativo", específico para o problema de mistura de densidades Gaussianas, tendo em vista uma possível aplicação em aprendizado em tempo real *não-supervisionado* dos parâmetros das funções de base de uma rede RBF.

5.2 Predição Linear de Séries Temporais

O problema da *predição linear* de um processo estocástico nada mais é do que uma particularização de um problema mais genérico : o da *filtragem linear* de um processo estocástico. Um filtro é dito linear se obedece ao princípio da superposição. De acordo com este princípio, a resposta do filtro à entrada $av_1(t) + bw_1(t)$ é o sinal $av_2(t) + bw_2(t)$, sendo a e b valores

constantes e $v_2(t)$ e $w_2(t)$ as respostas do filtro quando este for excitado, separadamente, pelas entradas $v_1(t)$ e $w_1(t)$. No caso de processos estocásticos de tempo discreto, os filtros projetados serão filtros com resposta ao impulso de tempo discreto.

O problema então da predição linear é projetar um filtro linear de tempo discreto cuja saída $y(n)$ fornece uma estimação de uma resposta desejada $d(n)$, dado um conjunto de amostras de entrada $u(0), u(1), \ldots,$. Uma abordagem bastante utilizada para o problema de otimização do filtro é minimizar o *valor quadrático médio do sinal de erro*, que é definido como a diferença entre a resposta desejada e a saída real do filtro. Para entradas estacionárias, a solução resultante é comumente denominada de *filtro de Wiener*, o qual é dito ótimo no *sentido quadrático médio.*

Consideremos então o filtro linear transversal de tempo discreto da Fig. 5.1.

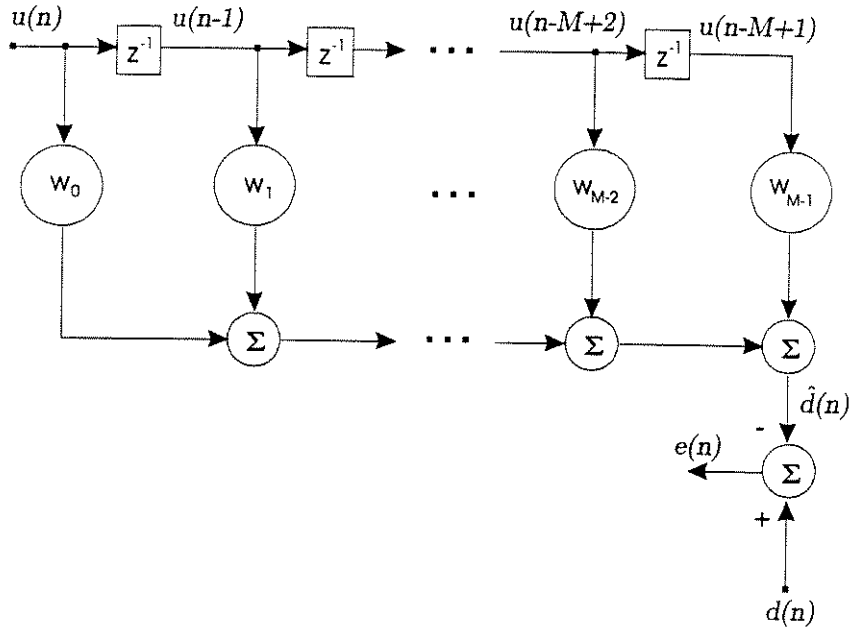

FIGURA 5.1: Filtro linear transversal com resposta ao impulso finita (FIR)

As entradas do filtro consistem da série temporal $\{u(n)\}$ e o filtro é caracterizado pela resposta ao impulso $w_0, w_1, \ldots, w_{M-1}$. Em algum instante de tempo discreto n, o filtro produz a saída $y(n)$ dada pela convolução

$$y(n) = \sum_{k=0}^{M-1} w_k u(n-k)\,, \tag{5.1}$$

ou seja, a saída do filtro consiste do produto interno entre as entradas do filtro

$$\mathbf{u}(n) = [u(n), u(n-1), \ldots, u(n-M+1)] \tag{5.2}$$

e o conjunto de pesos

$$\mathbf{w} = [w_0, w_1, \ldots, w_{M-1}]\,. \tag{5.3}$$

A saída do filtro é utilizada para estimar uma saída desejada $d(n)$. Essa estimação é acompanhada de um erro $e(n)$, definido como a diferença entre a resposta desejada e a saída

do filtro, ou seja,

$$e(n) = d(n) - \sum_{k=0}^{M-1} w_k u(n-k). \tag{5.4}$$

Desejamos então tornar este erro tão pequeno quanto possível segundo algum critério estatístico. O critério aqui utilizado será a minimização do erro quadrático médio. Portanto, podemos definir a função de custo para o filtro transversal da Fig. 5.1 como sendo

$$\begin{aligned} J &= \frac{1}{2}E\left[e^2(n)\right] = \frac{1}{2}E\left[d^2(n)\right] - E\left[\sum_{k=0}^{M-1} w_k u(n-k)d(n)\right] \\ &+\frac{1}{2}E\left[\sum_{k=0}^{M-1}\sum_{i=0}^{M-1} w_k w_i u\left(n-k\right)u\left(n-i\right)\right] \end{aligned} \tag{5.5}$$

Como a esperança é um operador linear, podemos reescrever a Eq. (5.5) na forma equivalente

$$J = \frac{1}{2}E\left[d^2(n)\right] - \sum_{k=0}^{M-1} w_k E\left[u(n-k)d(n)\right] + \frac{1}{2}\sum_{k=0}^{M-1}\sum_{i=0}^{M-1} w_k w_i E\left[u\left(n-k\right)u\left(n-i\right)\right]. \tag{5.6}$$

Podemos identificar as três esperanças na Eq. (5.6) como segue :

1. Para a primeira esperança, temos

$$\sigma_d^2 = E\left[d^2(n)\right], \tag{5.7}$$

onde σ_d^2 é a variância da resposta desejada $d(n)$, assumida aqui como sendo de média zero.

2. Para a segunda,

$$p(-k) = E\left[u(n-k)d(n)\right], \tag{5.8}$$

onde $p(-k)$ é a correlacão cruzada entre a entrada $u(n-k)$ e a resposta desejada $d(n)$. É importante frisarmos aqui que estamos considerando que $u(n-k)$ e $d(n)$ são conjuntamente estacionários.

3. Para a terceira esperança nós temos

$$r_u(i-k) = E\left[u\left(n-k\right)u\left(n-i\right)\right] \tag{5.9}$$

onde $r(i-k)$ é a função de auto-correlação das entradas do filtro para uma distância entre elas de $i-k$.

Sendo assim, podemos reescrever a Eq. (5.6) na forma

$$J = \frac{1}{2}\sigma_d^2 - \sum_{k=0}^{M-1} w_k p(-k) + \frac{1}{2}\sum_{k=0}^{M-1}\sum_{i=0}^{M-1} w_k w_i r_u(i-k). \tag{5.10}$$

A Eq. (5.10) nos indica que quando as entradas do filtro e a resposta desejada forem conjuntamente estacionárias, a função de custo, ou erro quadrático médio J, será precisamente

uma função de segunda ordem dos pesos do filtro. Conseqüentemente, podemos visualizar a dependência da função de custo J em relação aos pesos $w_0, w_1, \ldots, w_{M-1}$ como uma superfície no espaço de $(M+1)$ dimensões com M graus de liberdade, representados pelos pesos do filtro. Esta superfície é comumente denominada de *superfície de desempenho do erro* e é caracterizada por um mínimo **global**. É exatamente neste ponto que o filtro é dito **ótimo**, no sentido que o erro quadrático médio atinge o seu valor mínimo $J_{\min}$. Neste ponto, o vetor gradiente $\boldsymbol{\nabla}(J)$ é identicamente igual a zero. Em outras palavras,

$$\nabla_k(J) = 0, \qquad k = 0, 1, \ldots, M-1, \tag{5.11}$$

onde $\nabla_k(J)$ é a derivada da função de custo J com relação ao k-ésimo peso w_k.

Derivando a Eq. (5.10) em relação ao peso w_k, temos

$$\nabla_k(J) = \frac{\partial J}{\partial w_k} = -p(-k) + \sum_{i=0}^{M-1} w_i r_u(i-k) \tag{5.12}$$

Aplicando então a condição necessária e suficiente da Eq. (5.11) para optimalidade da Eq. (5.12), encontramos que o conjunto ótimo de pesos $w_{o0}, w_{o1}, \ldots, w_{oM-1}$ para o filtro da Fig. 5.1 é determinado pelo seguinte sistema de equações :

$$\sum_{i=0}^{M-1} w_{oi} r_u(i-k) = p(-k), \quad k = 0, 1, \ldots, M-1\,. \tag{5.13}$$

Este sistema é conhecido como as *equações de Wiener-Hopf*, e o filtro cujos pesos satisfazem as equações de Wiener-Hopf é denominado de *filtro de Wiener*. Podemos expressar as equações de Wiener-Hopf na forma matricial

$$\mathbf{R}\mathbf{w}_o = \mathbf{p}\,, \tag{5.14}$$

onde

$$\mathbf{R} = E\left[\mathbf{u}(n)\,\mathbf{u}(n)^T\right] = \begin{bmatrix} r(0) & r(1) & \cdots & r(M-1) \\ r(1) & r(0) & \cdots & r(M-2) \\ \vdots & \vdots & \ddots & \vdots \\ r(M-1) & r(M-2) & \cdots & r(0) \end{bmatrix}$$

é a *matriz de correlação* $M \times M$ do vetor de entradas $\mathbf{u}(n) = [u(n), \ldots, u(n-M+1)]^T$,

$$\mathbf{p} = E\left[\mathbf{u}(n)\,d(n)\right] = [p(0), p(-1), \ldots, p(1-M)] \tag{5.15}$$

é o vetor de correlação cruzada $M \times 1$ entre as entradas do filtro e a resposta desejada $d(n)$, e

$$\mathbf{w}_o = [w_{o0}, w_{o1}, \ldots, w_{oM-1}] \tag{5.16}$$

é o vetor $M \times 1$ de pesos ótimos do filtro transversal da Fig. 5.1.

Para solucionarmos as equações de Wiener-Hopf (5.14) para $\mathbf{w}_o$, assumimos que a matriz de correlação $\mathbf{R}$ é não-singular. Pré-multiplicamos então ambos os membros da Eq. (5.14) por $\mathbf{R}^{-1}$, a inversa da matriz de correlação, obtendo

$$\mathbf{w}_o = \mathbf{R}^{-1}\mathbf{p}\,. \tag{5.17}$$

Portanto, para solucionarmos as equações de Wiener-Hopf para os pesos do filtro, necessitamos, basicamente, calcular a inversa da matriz de correlação. Esta inversão pode se tornar muito complicada se o número de pesos no filtro for muito grande. Sendo assim, uma maneira de evitarmos esta inversão de matrizes é utilizarmos o método do *gradiente descendente*. De acordo com este método, os pesos do filtro assumem uma forma variante no tempo e seus valores são ajustados de uma forma *iterativa*, ao longo da superfície de erro, com o objetivo de movê-los progressivamente em direção à solução ótima. Os ajustes aplicados aos pesos do filtro têm a direção oposta ao vetor gradiente $\boldsymbol{\nabla}(J)$.

Seja então $w_k\,(n)$ o valor do peso w_k calculado na iteração ou tempo discreto n pelo método do gradiente descendente. De uma forma correspondente, o gradiente da superfície de erro do filtro em relação aos pesos, assume uma forma variante no tempo, dada por

$$\nabla_k\,(J\,(n)) = -p(-k) + \sum_{i=0}^{M-1} w_i\,(n)\,r_u(i-k). \tag{5.18}$$

De acordo com o método do gradiente descendente, o ajuste aplicado ao peso $w_k(n)$, no instante n, é definido por :

$$\Delta w_k(n) = -\eta\nabla_k\,(J\,(n))\,, \qquad k = 0, 1, \ldots, M-1\,, \tag{5.19}$$

onde η é uma constante positiva denominada de *taxa de aprendizado*. Dado o valor do k-ésimo peso $w_k(n)$ no instante n, o valor atualizado para este peso no instante $n-1$, será dado por

$$\begin{aligned} w_k\,(n+1) &= w_k\,(n) + \Delta w_k(n) \\ &= w_k\,(n) - \eta\nabla_k\,(J\,(n))\,, \qquad k = 0, 1, \ldots, M-1\,. \end{aligned} \tag{5.20}$$

Na forma matricial,

$$\mathbf{w}\,(n+1) = \mathbf{w}\,(n) - \eta\boldsymbol{\nabla}\,(J\,(n))\,, \tag{5.21}$$

onde

$$\boldsymbol{\nabla}\,(J\,(n)) = -\mathbf{p} + \mathbf{R}\mathbf{w}(n). \tag{5.22}$$

Portanto,

$$\mathbf{w}\,(n+1) = \mathbf{w}\,(n) + \eta\,[\mathbf{p} - \mathbf{R}\mathbf{w}(n)]\,, \qquad n = 0, 1, \ldots \tag{5.23}$$

5.2.1 O Algoritmo de Mínimos Quadrados (LMS)

O algoritmo de *mínimos quadrados* ("LMS - least-mean-square algorithm") [54, 55] é um importante membro da família de algoritmos baseados no gradiente estocástico. Uma característica importante do LMS é sua simplicidade; ele não requer medições das funções de

auto-correlação pertinentes nem requer inversão de matrizes. Na verdade, é a simplicidade do algoritmo LMS que o torna um *padrão* entre os algoritmos de filtragem adaptativa.

A fim de obtermos uma estimação do vetor gradiente $\nabla(J(n))$, a estratégia mais óbvia é introduzirmos estimações da matriz de correlação $\mathbf{R}$ e do vetor de correlação cruzada $\mathbf{p}$ na fórmula

$$\nabla(J(n)) = -\mathbf{p} + \mathbf{R}\mathbf{w}(n). \tag{5.24}$$

A escolha mais simples de estimadores para $\mathbf{R}$ e $\mathbf{p}$ é utilizar estimações *instantâneas* que são baseadas nos valores amostrais do vetor de entradas e na resposta desejada, como definido por

$$\widehat{\mathbf{R}}(n) = \mathbf{u}(n)\mathbf{u}^T(n) \tag{5.25}$$

e

$$\widehat{\mathbf{p}}(n) = \mathbf{u}(n)d(n)\,. \tag{5.26}$$

Portanto, a estimação instantânea do vetor gradiente será dada por

$$\widehat{\nabla}(J(n)) = -\mathbf{u}(n)d(n) + \mathbf{u}(n)\mathbf{u}^T(n)\widehat{\mathbf{w}}(n)\,. \tag{5.27}$$

Substituindo a estimação em (5.27), do vetor gradiente $\nabla(J(n))$, no algoritmo do gradiente descendente dado pela Eq. (5.23), obtemos uma nova relação recursiva para atualização do vetor de pesos :

$$\widehat{\mathbf{w}}(n+1) = \widehat{\mathbf{w}}(n) + \mu\mathbf{u}(n)\left[d(n) - \mathbf{u}^T(n)\widehat{\mathbf{w}}(n)\right]\,. \tag{5.28}$$

O resultado anterior pode ser reescrito na forma de três relações básicas, como segue :

1. Saída do filtro :

$$y(n) = \mathbf{u}^T(n)\widehat{\mathbf{w}}(n)\,; \tag{5.29}$$

2. Estimação do erro :

$$e(n) = d(n) - y(n)\,; \tag{5.30}$$

3. Adaptação do vetor de pesos :

$$\widehat{\mathbf{w}}(n+1) = \widehat{\mathbf{w}}(n) + \mu\mathbf{u}(n)e(n)\,. \tag{5.31}$$

O procedimento iterativo é inicializado com um valor inicial $\widehat{\mathbf{w}}(0)$. Uma escolha conveniente é o vetor nulo $\widehat{\mathbf{w}}(0) = \mathbf{0}$. No método do gradiente descendente, o vetor de pesos $\mathbf{w}(n)$ começa então no valor inicial $\mathbf{w}(0)$ e segue uma trajetória bem definida (ao longo da superfície de erro) que eventualmente termina na solução ótima $\mathbf{w}_0$. Para isto, o parâmetro taxa de aprendizado η deve ser escolhido apropriadamente. No algoritmo LMS, o vetor de pesos $\widehat{\mathbf{w}}(n)$ representa uma "estimação" de $\mathbf{w}(n)$, e segue uma trajetória aleatória em direção à solução ótima $\mathbf{w}_0$. É por esta razão que o LMS é denominado de um "algoritmo de gradiente estocástico". Quando o número de iterações aproxima-se do infinito, $\widehat{\mathbf{w}}(n)$ realiza um "passeio aleatório" (Brownian Motion) em torno da solução ótima $\mathbf{w}_0$.

5.3 Predição Não-Linear de Séries Temporais

Historicamente, um importante passo em direção a uma proposta além dos modelos lineares para predição de séries temporais, foi feita em 1980 por Tong e Lim . Depois de mais de cinco décadas aproximando-se um sistema qualquer através de uma única função linear, eles sugeriram o uso de duas funções lineares : o *modelo auto-regressivo com limiar TAR* ("threshold autoregressive model"), o qual é não-linear num sentido global, ou seja, ele baseia-se no valor do estado do sistema para escolher um entre dois modelos auto-regressivos lineares. A partir deste modelo, uma extensão natural foi a introdução de diversas funções lineares ou a inclusão de fatores quadráticos ou potências de ordens superiores nos modelos lineares originais : os *modelos de Volterra.*

Os modelos TAR, os modelos de Volterra e suas extensões expandem de forma significativa a possibilidade de relações funcionais para modelagem de séries temporais. Em contrapartida, estes abdicam da simplicidade com a qual os modelos lineares podem ser entendidos e ajustados aos dados. Para que modelos não-lineares sejam úteis, deve existir um processo que explore as características dos dados para guiar (e restringir) a construção do modelo. A falta de solução para este problema limitou por muito tempo o uso de modelos não-lineares para séries temporais.

Com o advento das redes neurais artificiais e sua reconhecida capacidade de aproximação de funções não-lineares, diversos trabalhos têm relatado o uso de redes neurais no problema de predição de séries temporais. Um dos trabalhos pioneiros nesta área foi o de Lapedes e Farber [56], os quais treinaram uma rede neural para emular a relação entre a saída (o próximo ponto da série) e entradas (predecessores) para uma série temporal gerada artificialmente. Desde então, as redes neurais tem sido largamente utilizadas em problemas envolvendo a predição *não-linear* de séries temporais.

Na predição não-linear de séries temporais através de redes neurais, a estimação $\widehat{u}(n)$ da amostra real $u(n)$ da série temporal é tomada como a saída $\mathcal{N}$ de uma rede neural alimentada por valores passados da série temporal, ou seja,

$$u(n) = \widehat{u}(n) + e(n) = \mathcal{N}[u(n-1), u(n-2), \ldots, u(n-M)] + e(n), \tag{5.32}$$

onde $e(n)$ é o erro de predição.

Apresentaremos a seguir os principais conceitos envolvendo redes neurais artificiais e a exposição de duas arquiteturas de rede que serão utilizadas para o nosso problema de predição.

5.3.1 As Redes Neurais Artificiais

Uma *rede neural artificial* consiste de um conjunto de unidades de processamento simples, as quais estão interconectadas de alguma forma apropriada e se comunicam através de sinais enviados a um grande número de conexões ponderadas. Além disso, essa estrutura inerentemente paralela e distribuída é capaz, em princípio, de armazenar conhecimento e generalizá-lo através de um processo de treinamento. Para uma completa especificação de uma rede neural, necessitamos de uma série de componentes, descritos a seguir [57, 58] :

- Um conjunto de unidades de processamento ("neurônios" ou "células");
- Um estado de ativação y_j para cada unidade, o qual também representa a saída da unidade;

- Conexões entre as unidades; Geralmente cada conexão é caracterizada por um "peso" w_{ij}, que determina o efeito que o sinal da unidade j produz na unidade i;
- Uma regra de propagação que determina a entrada efetiva v_i destinada a uma unidade a partir de suas entradas externas;
- Uma função de ativação $\varphi_i(\cdot)$ que determina o novo nível de ativação $y_i(n)$ baseado na entrada efetiva (ou potencial de ativação) $v_i(n)$ e o estado corrente $a_i(n)$ no instante n;
- Uma entrada externa ou "offset";
- Um método de *aprendizado*, segundo o qual a rede extrai informação do ambiente e aproveita para melhorar seu desempenho;
- Um ambiente dentro do qual o sistema deve operar, proporcionando sinais de entrada e, se necessário, sinais de erro.

Cada unidade de processamento desempenha uma tarefa relativamente simples : recebe um sinal de unidades vizinhas ou de fontes externas e utiliza-o para produzir um sinal de saída que é propagado para outras unidades. Dentro das redes neurais podemos distinguir três tipos de unidades de processamento : as unidades de *entrada*, que recebem dados ou sinais do ambiente externo; as unidades *escondidas*, cujos sinais de entrada e saída permanecem dentro do próprio sistema; e as unidades de *saída*, que enviam os dados para fora do sistema.

A entrada total para a unidade i é simplesmente a soma ponderada das saídas de cada unidade conectada ao neurônio i, além de um termo de "offset" θ_i :

$$v_i(n) = \sum_j w_{ij}(n) y_j(n) + \theta_i(n). \tag{5.33}$$

Além disso, necessitamos de uma regra que forneça o efeito da entrada total na ativação da unidade. Necessitamos de uma função $\varphi_i(\cdot)$ que tome a entrada total $v_i(n)$ e a ativação atual $y_i(n)$ e produza um novo valor para a ativação da unidade i :

$$y_i(n+1) = \varphi_i(y_i(n), v_i(n)).$$

Freqüentemente, a função de ativação é uma função apenas da entrada total do neurônio :

$$y_i(n+1) = \varphi_i(v_i(n)) = \varphi_i\left(\sum_j w_{ij}(n) y_j(n) + \theta_i(n)\right). \tag{5.34}$$

Existem três tipos básicos de função de ativação :

1. *Função de Limiar :* Para este tipo de função de ativação, temos

$$\varphi(v) = \begin{cases} 1, & \text{se } v \geq 0 \\ 0, & \text{se } v < 0. \end{cases} \tag{5.35}$$

2. *Função Linear por Partes* : Como exemplo de uma função linear por partes, temos

$$\varphi(v) = \begin{cases} 1, & \text{se } v \geq 1/2 \\ 2v, & \text{se } -1/2 < v < 1/2 \\ -1 & \text{se } v \leq -1/2 \end{cases} \tag{5.36}$$

3. *Função Sigmóide :* A função sigmóide é a mais utilizada. É definida como uma função estritamente crescente que exibe suavidade e propriedades assintóticas. Um exemplo de função sigmóide é a *função logística,* definida por

$$\varphi(v) = \frac{1}{1+\exp(-av)} \tag{5.37}$$

onde a é o parâmetro que mede a declividade da função sigmóide. Outro exemplo de função sigmoide é a função tangente hiperbólica, definida por

$$\varphi(v) = \tanh\left(\frac{v}{2}\right) = \frac{1-\exp(-v)}{1+\exp(-v)} \tag{5.38}$$

A maneira na qual os neurônios de uma rede neural estão dispostos e os padrões de conexões por eles estabelecidos determinam a *arquitetura* da rede. Em geral, podemos identificar duas classes distintas de arquiteturas de redes neurais :

- *Redes Multicamadas com Propagação Direta,* onde o fluxo de sinais dos neurônios de entrada para os neurônios de saída se dá de uma forma direta, sem realimentação. Neste caso, o processamento pode estender-se por várias camadas de neurônios. Como exemplo desse tipo de rede, temos os perceptrons multicamadas e o Adaline;
- *Redes Recorrentes,* que contêm conexões estabelecendo realimentações, ou seja, conexões que se estendem das saídas dos neurônios para entradas de neurônios na mesma camada ou em camadas anteriores. Como exemplo de redes recorrentes, temos o mapa de Kohonen e a rede de Hopfield.

Uma rede neural deve ser configurada de tal forma que a aplicação de um conjunto de entradas produza um conjunto de saídas desejadas. Existem vários métodos para se estabelecer os pesos entre as diversas conexões. Um deles é fixar os pesos *explicitamente,* utilizando-se um conhecimento *a priori.* Um outro modo é "treinar" a rede neural a partir de um conjunto de dados de *treinamento* e ajustar os pesos da rede de acordo com alguma *regra de aprendizado.*

Podemos classificar o aprendizado de redes neurais em duas categorias :

- *Aprendizado supervisionado ou aprendizado associativo,* no qual a rede é treinada a partir da apresentação de pares entrada-saída. Estes pares podem ser fornecidos por um "supervisor" externo ou pelo sistema que contém a rede.
- *Aprendizado Não-Supervisionado ou Auto-Organizado,* no qual não existem exemplos específicos da função a ser aprendida pela rede. Neste paradigma, a rede procura descobrir as características estatísticas importantes dos dados de entrada e procura desenvolver representações internas para codificar as características dos dados de entrada.

Em ambos os paradigmas de aprendizado, temos como principal meta ajustar os pesos das conexões entre neurônios de acordo com alguma regra de aprendizado. De uma maneira geral, todas as regras de aprendizado para modelos deste tipo podem ser consideradas como uma variante da regra de aprendizado de Hebb [59]. A idéia básica é que se dois neurônios i e j são ativados simultaneamente, suas interconexões devem ser reforçadas. Se i recebe uma entrada de j, a versão mais simples da regra de aprendizado de Hebb estabelece que o peso w_{ij} deve ser modificado de acordo com a regra

$$\Delta w_{ij} = \gamma a_i a_j, \tag{5.39}$$

onde γ é uma constante positiva de proporcionalidade representando a taxa de aprendizado. Uma outra regra comum utiliza não a saída real do neurônio i, mas a diferença entre as saídas real e desejada para ajuste dos pesos :

$$\Delta w_{ij} = \gamma \left(d_i - a_i\right) a_j \,, \tag{5.40}$$

onde d_i é a resposta desejada fornecida pelo supervisor da rede. Esta regra é comumente denominada de *regra de Widrow-Hoff* ou *regra delta.*

Em nosso estudo de predição de tráfego auto-similar, decidimos utilizar apenas redes multicamadas com propagação direta. Em particular, estudamos o uso do *perceptron multicamadas FIR* e a *rede de funções de base radiais (RBF).*

O perceptron multicamadas FIR pode ser considerado como uma extensão do tradicional modelo de perceptron multicamadas. Em 1991, em uma competição de predição e análise de séries temporais realizada pelo Santa Fe Institute, o *perceptron multicamadas FIR* foi, dentre os modelos de redes neurais, o preditor que apresentou o melhor desempenho para séries caóticas [60].

Por outro lado, devido à sua propriedade de *melhor aproximação* em problemas de aproximação de funções e a sua rapidez de treinamento, as redes neurais RBF têm recebido grande atenção por parte da comunidade científica nos últimos anos. Resolvemos então, estudar a aplicação desses dois tipos de arquiteturas para avaliar a sua aplicabilidade em nosso problema em particular. A seguir, vamos descreve estes dois tipos de redes neurais.

5.3.2 O Perceptron Multicamadas *FIR*

O modelo tradicional do perceptron multi-camadas apresenta como característica básica o fato de que as conexões sinápticas são representadas por coeficientes chamados de *pesos.* Uma variação deste modelo pode ser obtida se trocarmos os pesos sinápticos estáticos por *filtros lineares* com resposta ao impulso finita (FIR).

Por um filtro linear do tipo FIR entende-se que, para uma excitação de entrada finita, a saída do filtro será também de duração finita. Para este filtro, a saída $y(n)$ corresponde a uma convolução dos pesos sinápticos com as entradas do filtro, ou seja,

$$y(n) = \sum_{k=0}^{T} w(k) x(n-k) \,. \tag{5.41}$$

Vamos nos referir a esta estrutura como o *perceptron multicamadas FIR* [60].

Seja $w_{ji}^c\,(l)$ o peso do l-ésimo "tap" do filtro FIR modelando a sinapse que conecta a saída do neurônio i à entrada do neurônio j pertencente à camada c. O índice l varia de 0 a M_{ji}^c, onde M_{ji}^c é o número total de atrasadores no filtro FIR que conecta a saída do neurônio i à entrada do neurônio j pertencente à camada c. Por uma questão de simplificação, adotaremos que todos os filtros, em todas as camadas, possuem o mesmo número de atrasadores. De acordo com este modelo, o sinal $s_{ji}^c(n)$ que aparece na saída da i-ésima sinapse do neurônio j na camada c, é dado pela *soma de convolução*

$$s_{ji}^c(n) = \sum_{l=0}^{M} w_{ji}^c(l) y_i^{c-1}(n-l) \,, \tag{5.42}$$

onde n denota o tempo discreto e $y_i^{c-1}(n-l)$ a saída do neurônio i da camada $c-1$ atrasada de l unidades de tempo e que está ligada ao neurônio j.

Podemos simplificar a expressão acima por uma representação matricial, introduzindo as seguintes notações para o vetor de estados e o vetor de pesos da sinapse i :

$$\mathbf{y}_i^{c-1}(n) = \left[y_i^{c-1}(n), ..., y_i^{c-1}(n-M)\right]^T, \tag{5.43}$$

$$\mathbf{w}_{ji} = [w_{ji}(0), w_{ji}(1), w_{ji}(2), ..., w_{ji}(M)]^T. \tag{5.44}$$

Podemos assim, expressar o sinal $s_{ji}(n)$ como o produto interno dos vetores $\mathbf{w}_{ji}$ e $\mathbf{y}_i^{c-1}(n)$, isto é,

$$s_{ji}^c(n) = \mathbf{w}_{ji}^T \mathbf{y}_i^{c-1}(n). \tag{5.45}$$

Considerando a contribuição de todas as sinapses i, com $i = 1, 2, ..., p$, pertencentes a um dado neurônio, o potencial de ativação do neurônio j, localizado na camada c, será dado por

$$\begin{aligned} v_j^c(n) &= \sum_{i=1}^{p} s_{ji}^c(n) - \theta_j \\ &= \sum_{i=1}^{p} \mathbf{w}_{ji}^T \mathbf{y}_i^{c-1}(n) - \theta_j = \sum_{i=0}^{p} \mathbf{w}_{ji}^T \mathbf{y}_i^{c-1}(n), \end{aligned} \tag{5.46}$$

onde incorporamos o filtro sináptico cujos pesos são todos iguais a θ_j e os elementos do vetor de entradas $\mathbf{y}_0^{c-1}(n)$ são todos iguais a -1.

A saída do neurônio j na camada c será portanto

$$y_j^c(n) = \varphi(v_j^c(n)), \tag{5.47}$$

onde $\varphi(n)$ representa a função de ativação não-linear do neurônio j na camada c.

Notemos que, se os vetores $\mathbf{w}_{ji}$ e $\mathbf{y}_i^{c-1}(n)$ forem substituídos pelos escalares w_{ji} e $y_i^{c-1}(n)$ e se a operação de produto interno for substituída pela multiplicação simples, o modelo dinâmico de um neurônio, descrito em termos matemáticos pelas Eqs. (5.46) e (5.47), reduz-se ao modelo tradicional de neurônio estático. Com efeito, a analogia entre o algoritmo de retropropagação padrão e o de retropropagação temporal será sempre buscada durante o desenvolvimento que se segue.

O Algoritmo de *Retropropagação* Temporal

Para treinarmos a rede, utilizamos um algoritmo de aprendizado supervisionado no qual a resposta atual de cada neurônio na camada de saída é comparada com uma resposta desejada em cada instante de tempo. Assumimos por um momento que o neurônio j se encontra na camada de saída, com a sua resposta real no instante n denotada por $y_j(n)$ e que a resposta

desejada para este neurônio seja $d_j(n)$. Podemos então definir um valor instantâneo para a soma de erros quadráticos produzidos pela rede como segue :

$$E(n) = \frac{1}{2}\sum_j e_j^2(n), \tag{5.48}$$

onde o índice j refere-se apenas aos neurônios da camada de saída e $e_j(n)$ refere-se ao sinal de erro definido por

$$e_j(n) = d_j(n) - y_j(n). \tag{5.49}$$

O objetivo do treinamento é minimizar sobre $\mathbf{W}$ (conjunto de todos os coeficientes dos filtros da rede) a função de custo

$$E_{total} = \sum_n E(n), \tag{5.50}$$

onde o somatório é realizado sobre todo o tempo n.

A maneira mais direta de minimizarmos a função de custo é utilizando o método do *gradiente descendente*, o qual requer o conhecimento da *matriz de gradientes*

$$\begin{aligned} \nabla_{\mathbf{W}} E_{total} &= \frac{\partial E_{total}}{\partial \mathbf{W}} = \sum_n \frac{\partial E(n)}{\partial \mathbf{W}} \\ &= \sum_n \nabla_{\mathbf{W}} E(n) \end{aligned} \tag{5.51}$$

onde $\nabla_{\mathbf{W}} E(n)$ é o gradiente de $E(n)$ com relação à matriz de pesos $\mathbf{W}$. Podemos, se desejado, continuar com o desenvolvimento da Eq. (5.51) e derivar as equações de atualização dos pesos sinápticos da rede sem aproximações. No entanto, a fim de desenvolvermos um algoritmo que possa ser usado para treinar a rede em tempo real, utilizamos uma *estimação* do gradiente, a saber, $\nabla_{\mathbf{W}} E(n)$, que resulta numa aproximação do método do gradiente estocástico. Assim, os filtros sinápticos são atualizados a cada incremento de tempo segundo a dinâmica

$$\mathbf{w}_{ji}^c(n+1) = \mathbf{w}_{ji}^c(n) - \eta \frac{\partial E(n)}{\partial \mathbf{w}_{ji}^c(n)}, \tag{5.52}$$

onde η é a taxa de aprendizado.

O modo mais óbvio de se obter os termos do gradiente envolve a técnica de se "abrir" a estrutura da rede e pô-la na sua forma estática equivalente, e então aplicar o algoritmo de *retropropagação* padrão. No entanto, teremos com isto uma série de inconvenientes. O *algoritmo de retropropagação*, quando aplicado a uma rede estática, determina os termos de gradiente associados a **cada** peso da rede. Como a rede "aberta" contém pesos repetidos, os termos dos gradientes individuais terão de ser posteriormente recombinados cuidadosamente para se encontrar o gradiente total correspondente a cada coeficiente dos filtros. Com isto, o processamento dos pesos que ora era distribuído localmente, passa a requerer um tratamento

global, em que todos os termos precisam ser examinados. Até o momento, nenhuma forma simples de recorrência é conhecida.

Um algoritmo mais atrativo pode ser obtido se aproximarmos o problema sob uma perspectiva um pouco diferente. O gradiente da função de custo com relação a um filtro sináptico pode também ser expandido da forma que se segue :

$$\frac{\partial E_{total}}{\partial \mathbf{w}_{ji}^{c}(n)} = \sum_{n} \frac{\partial E_{total}}{\partial v_{j}^{c}(n)} \frac{\partial v_{j}^{c}(n)}{\partial \mathbf{w}_{ji}^{c}(n)}, \tag{5.53}$$

onde o índice de tempo n transcorre sobre $v_j^c(n)$, e não sobre $E(n)$.

Podemos interpretar a derivada $\partial E_{total}/\partial v_j^c(n)$ como a variação no erro quadrático total ao longo de todo o tempo devido à variação no potencial de ativação de um neurônio em um único instante de tempo n. Notemos que não estamos mais expressando o gradiente total da maneira tradicional, como uma soma de gradientes instantâneos, ou seja,

$$\frac{\partial E_{total}}{\partial v_{j}^{c}(n)} \neq \frac{\partial E\left(n\right)}{\partial \mathbf{w}_{ji}^{c}}. \tag{5.54}$$

Apenas quando somamos sobre todo n é que as expressões se tornam equivalentes. Utilizando esta nova expansão, o seguinte algoritmo estocástico é formado :

$$\mathbf{w}_{ji}^{c}(n+1) = \mathbf{w}_{ji}^{c}(n) - \eta \frac{\partial E_{total}}{\partial v_{j}^{c}(n)} \frac{\partial v_{j}^{c}(n)}{\partial \mathbf{w}_{ji}^{c}(n)} \tag{5.55}$$

Da Eq. (5.46) segue-se imediatamente que

$$\frac{\partial v_{j}^{c}(n)}{\partial \mathbf{w}_{ji}^{c}(n)} = y_{i}^{c-1}(n) \tag{5.56}$$

Podemos definir o gradiente local para o neurônio j na camada c como sendo

$$\delta_{j}^{c}\left(n\right) = -\frac{\partial E_{total}}{\partial v_{j}\left(n\right)} \tag{5.57}$$

Reescrevemos portanto, a Eq. (5.55) na forma familiar

$$\mathbf{w}_{ji}^{c}(n+1) = \mathbf{w}_{ji}^{c}(n) - \eta \delta_{j}^{c}\left(n\right) \mathbf{y}_{i}^{c-1}(n) \tag{5.58}$$

Como no algoritmo de retropropagação padrão, a forma explícita do gradiente local $\delta_j^c(n)$ dependerá da camada em que o neurônio se encontrar. Desenvolvendo-se as expressões para o gradiente local em cada caso, o algoritmo final pode ser descrito como segue [60] :

$$\mathbf{w}_{ji}^{c}(n+1) = \mathbf{w}_{ji}^{c}(n) - \eta \delta_{j}^{c}\left(n\right) \mathbf{y}_{i}^{c-1}(n), \tag{5.59}$$

$$\delta_{j}^{c}\left(n\right) = \begin{cases} e_{j}\left(n\right) \varphi'(v_{j}(n)), & \text{se } c = L \\ \varphi'(v_{j}(n)) \sum_{m \in A} \nabla_{m}\left(n\right) \mathbf{w}_{mi}^{c}, & \text{se } 1 \leq c \leq L-1, \end{cases} \tag{5.60}$$

onde

$$\nabla_m(n) = [\delta_m(n), \delta_m(n+1), ..., \delta_m(n+M)]^T, \tag{5.61}$$

e A é o conjunto de todos os neurônios cujas entradas são alimentadas pelo neurônio j.

Se analisarmos as equações acima, notaremos que o cálculo do termo $\delta_j^c(n)$ é, de fato, não causal. A origem desta filtragem não causal pode ser identificada considerando-se a definição da Eq. (5.57). Como leva algum tempo para que a saída de qualquer neurônio interno propague completamente através da rede, a mudança no erro *total* devido à mudança em um estado interno é uma função dos valores futuros dentro da rede. Como a rede é do tipo FIR, apenas um número finito de valores futuros deve ser considerado e, portanto, uma simples reindexação permite-nos reescrever o algoritmo em uma forma causal :

$$\mathbf{w}_{ji}^{L-k}(n+1) = \mathbf{w}_{ji}^{L-k}(n) - \eta\delta_j^{L-k}(n-kM)\,\mathbf{y}_i^{L-k-1}(n-kM) \tag{5.62}$$

$$\delta_j^{L-k}(n-kM) = \begin{cases} e_j(n)\,\varphi'(v_j(n)), & \text{se } k=0 \\ \varphi'(v_j(n-kM))\sum_{m\in A}\nabla_m(n-kM)\,\mathbf{w}_{mi}^{L-k}, & \text{se } 1\le k\le L-1 \end{cases} \tag{5.63}$$

Embora menos esteticamente agradável, estas equações diferem apenas em termos de uma mudança de índices. Estas equações são implementadas propagando-se continuamente os termos em delta no sentido de volta *sem* atraso. No entanto, por definição, isto força o deslocamento no tempo dos valores internos dos deltas. Assim, precisamos armazenar os estados $\mathbf{y}_i^c(n)$ apropriadamente para formar os termos necessários para adaptação. A propagação no sentido de volta dos termos em delta não requer nenhum atraso adicional e ainda continua simétrica à propagação no sentido direto.

5.3.3 A Rede de Funções de Base Radiais (RBF)

A rede neural do tipo funções de base radiais (RBF) tem suas origens em técnicas de interpolação exata de um conjunto de pontos em um espaço multidimensional. O problema de interpolação exata exige que cada vetor de entrada seja mapeado *exatamente* em um vetor desejado correspondente.

Consideremos então um mapeamento de um espaço de entrada $\mathbf{x}$ de d dimensões em um espaço unidimensional t. O conjunto de pontos consiste de N vetores de entrada $\mathbf{x}^n$ e os respectivos valores desejados t^n. O objetivo da interpolação exata é encontrar uma função $h(\mathbf{x})$ tal que

$$h(\mathbf{x}^n) = t^n, \qquad n = 1, \ldots, N. \tag{5.64}$$

A técnica de funções de base radiais utiliza um conjunto de N funções de base, uma para cada ponto, cada uma das quais assumindo a forma $\phi(\|\mathbf{x}-\mathbf{x}^n\|)$, onde $\phi(\cdot)$ é alguma função não-linear. Definidas dessa forma, a n-ésima função dependerá da distância $\|\mathbf{x}-\mathbf{x}^n\|$, normalmente Euclidiana, entre os vetores $\mathbf{x}$ e $\mathbf{x}^n$. A saída do mapeamento será uma combinação linear das funções de base, ou seja,

$$h(\mathbf{x}) = \sum_n w_n\phi(\|\mathbf{x}-\mathbf{x}^n\|). \tag{5.65}$$

O problema de interpolação estabelecido na Eq. (5.64), pode agora ser escrito na forma matricial

$$\Phi \mathbf{w} = \mathbf{t}, \tag{5.66}$$

onde $\mathbf{t} \equiv (t^n)$, $\mathbf{w} \equiv (w_n)$, e a matriz quadrada Φ possui elementos $\Phi_{nn'} = \phi(\|\mathbf{x} - \mathbf{x}^n\|)$. Se a matriz Φ for não-singular, o vetor de pesos $\mathbf{w} \equiv (w_n)$ pode ser obtido da forma

$$\mathbf{w} = \Phi^{-1}\mathbf{t}. \tag{5.67}$$

Já foi mostrado que, para uma grande classe de funções $\phi(\cdot)$, a matriz Φ é não-singular desde que os dados sejam distintos. Quando os pesos em (5.65) são dados pelos valores em (5.67), a função $h(\mathbf{x})$ representa uma superfície contínua diferenciável que passa exatamente por cada um dos pontos.

Várias formas para as funções de base têm sido consideradas. A mais comum delas é a Gaussiana

$$\phi(x) = \exp\left(-\frac{x^2}{2\sigma^2}\right), \tag{5.68}$$

onde σ é um parâmetro cujo valor controla as propriedades de suavização da função de interpolação. A Gaussiana é uma função de base localizada, com a propriedade de que $\phi \to 0$ quando $|x| \to \infty$. Uma outra escolha de função de base com esta mesma propriedade é a função

$$\phi(x) = \left(x^2 + \sigma^2\right)^{-\alpha}, \quad \alpha > 0. \tag{5.69}$$

No entanto, não é necessário que as funções de base sejam localizadas. Outras escolhas possíveis são a função spline

$$\phi(x) = x^2 \ln(x), \tag{5.70}$$

e a função

$$\phi(x) = \left(x^2 + \sigma^2\right)^{\beta}, \quad 0 < \beta < 1, \tag{5.71}$$

as quais possuem a propriedade de que $\phi \to \infty$ quando $x \to \infty$. Por questões de tratabilidade analítica, focalizaremos nossa atenção nas funções de base Gaussianas.

Introduzindo-se uma série de modificações ao procedimento de interpolação exata, Broomhead e Lowe obtiveram o modelo de redes neurais de funções de base radiais [61]. Este modelo proporciona uma função de interpolação mais suave, na qual o número de funções de base é determinado pela complexidade do mapeamento a ser representado, ao invés do tamanho do conjunto de dados. As seguintes modificações foram introduzidas :

1. O número M de funções de base não precisa ser igual ao número N de pontos, e é, tipicamente, muito menor do que N.
2. Os centros das funções de base não estão mais restritos a serem os vetores de entrada. Ao invés disso, devem ser determinados via um processo de treinamento.
3. Ao invés de possuírem o mesmo parâmetro σ, cada função possui seu próprio σ_j, o qual é também determinado via treinamento.

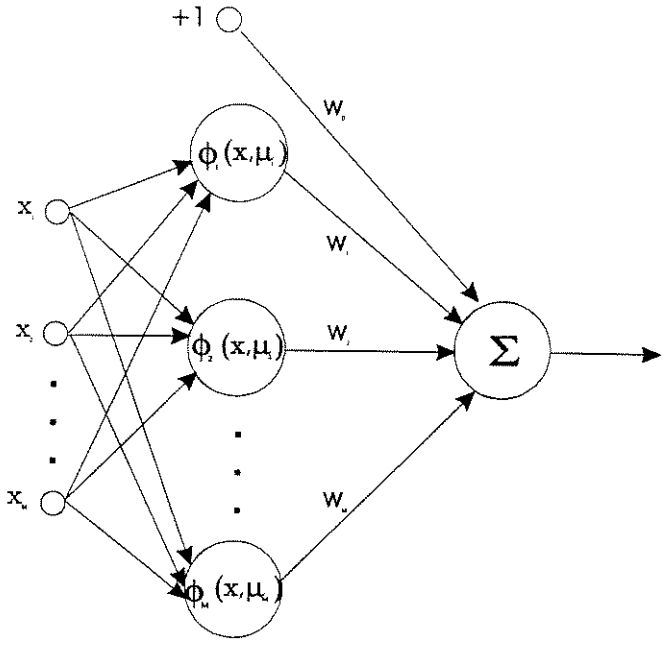

FIGURA 5.2: Diagrama esquemático de uma rede neural RBF

4. Parâmetros de polarização ("bias") são introduzidos na soma linear.

Quando estas modificações são introduzidas na fórmula de interpolação exata (5.65), obtemos a seguinte expressão para a rede neural de funções de base radiais :

$$y_k(\mathbf{x}) = \sum_{j=1}^{M} w_{kj}\phi_j(\mathbf{x}) + w_{k0}\,, \tag{5.72}$$

onde o índice k significa as diferentes saídas da rede.

Para o caso de funções de base Gaussianas, teremos

$$\phi_j(\mathbf{x}) = \exp\left(-\frac{\|\mathbf{x}-\boldsymbol{\mu}_j\|^2}{2\sigma^2}\right), \tag{5.73}$$

onde $\mathbf{x}$ é o vetor de entradas d-dimensional e $\boldsymbol{\mu}_j$ é o vetor que determina os centros das funções de base ϕ_j. Notemos que as funções de base Gaussianas não são normalizadas, uma vez que qualquer fator multiplicativo pode ser absorvido nos pesos da Eq. (5.72), sem perda de generalidade. Na Fig. 5.2 temos um diagrama da rede neural RBF.

Neste diagrama, podemos perceber que a rede RBF, diferentemente do perceptron multicamadas, possui apenas três camadas : a camada de nós-fonte, uma camada escondida formada de unidades de processamento não-linear e uma camada de saída linear.

Girosi e Poggio [62] demonstraram que as redes neurais de funções de base radiais possuem a propriedade de *melhor aproximação*. Um esquema de aproximação possui esta propriedade se, no conjunto de funções de aproximação (isto é, o conjunto de funções correspondendo a todas as possíveis escolhas dos parâmetros ajustáveis), existe uma função que apresenta erro de aproximação mínimo para qualquer função dada a ser aproximada. Eles mostraram também que esta propriedade não é compartilhada pelo perceptron multicamadas.

Treinamento da Rede RBF

As funções de base Gaussianas, consideradas anteriormente, podem ser generalizadas de forma a permitirem matrizes de covariância arbitrárias Σ_j. Assim, as funções de base assumem a forma

$$\phi_j(\mathbf{x}) = \exp\left\{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_j)^T \Sigma_j^{-1}(\mathbf{x}-\boldsymbol{\mu}_j)\right\}. \tag{5.74}$$

Sendo asssim, do ponto de vista de projeto, o nosso problema restringe-se a determinar, de uma forma apropriada, os valores dos parâmetros de cada Gaussiana $(\boldsymbol{\mu}_j, \Sigma_j)$ e os pesos w_{kj} da camada linear, dado que o número M de funções de base radiais esteja fixado[1].

Uma importante e interessante propriedade das redes neurais RBF é o fato delas proporcionarem uma visão unificada entre conceitos aparentemente descorrelacionados, como aproximação de funções, teoria de regularização, estimação de densidades de probabilidade e teoria de classificação ótima [63, 57]. Uma conseqüência deste ponto de vista unificado é a possibilidade de aplicação de diferentes procedimentos para o treinamento da rede RBF, o que não acontece com o perceptron multicamadas. Além disso, alguns desses procedimentos são também muito mais rápidos do que os métodos aplicados ao perceptron multicamadas. Este fato segue da interpretação que pode ser dada às representações internas formadas pelas unidades escondidas, levando-se a diferentes procedimentos de treinamento.

Existem na literatura diversos métodos de treinamento para determinação dos parâmetros de uma rede neural do tipo RBF. Dentre eles, os que mais se destacam são os métodos que utilizam treinamentos híbridos, que combinam aprendizado não-supervisionado com aprendizado supervisionado, os quais são executados em duas etapas totalmente independentes. Tipicamente, na primeira etapa, os parâmetros que governam as funções de base ($\boldsymbol{\mu}_j$ e Σ_j no nosso caso) são determinados utilizando-se métodos *não-supervisionados*. Na segunda etapa, os pesos da camada linear são determinados através de um método *supervisionado*. Existe ainda a possibilidade de treinamentos totalmente supervisionados, a exemplo de algoritmos do tipo gradiente estocástico[55].

No nosso trabalho, utilizamos a rede RBF proposta em [13] denominada de *rede RBF dinâmica*. Esta rede caracteriza-se pelo fato de possuir um aprendizado em *tempo real*, sem a necessidade de treinamentos prévios para determinação dos parâmetros da rede. Além disso, expomos neste trabalho o algoritmo EM de Dempster et al. [14] aplicado ao problema de estimação de parâmetros de mistura de densidades Gaussianas multivariáveis, uma alternativa bastante interessante para estimação dos parâmetros das Gaussianas da rede RBF. Nessa abordagem, supõe-se que os dados (vetores formados por amostras do passado) estão distribuídos no espaço segundo uma densidade de probabilidade parametrizada por uma combinação linear de densidades Gaussianas. Em princípio, qualquer função densidade de probabilidade pode ser representada por uma soma de densidades Gaussianas [63]. Trata-se portanto, de um algoritmo não-supervisionado, a exemplo do algoritmo de k-médias ("k-means"). Este algoritmo precisa ser realizado previamente, em um conjunto de treinamento. Portanto, não é apropriado para aplicações em tempo real.

Devido ao fato de estarmos interessados em preditores capazes de acompanhar possíveis variações estatísticas do sinal em questão, propomos neste trabalho uma versão para o algoritmo

[1] A determinação do número mínimo de funções de base apropriado para um dado problema constitui-se de um problema em aberto. Soluções para este problema têm sido sugeridas através da aplicação do critério de Akaike e algoritmos genéticos.

EM aplicado ao caso de misturas de densidades Gaussianas. Denominamos este algoritmo de "EM adaptativo". Neste algoritmo, as estimações dos parâmetros são feitas em tempo real e é introduzida uma função de penalidade para evitar a ocorrência de singularidades (Gaussianas degeneradas).

A seguir, vamos expor as principais características da rede RBF dinâmica proposta em [13]. Na seção 5.4 introduziremos os principais conceitos do algoritmo EM e a obtenção do algoritmo EM adaptativo para mistura de Gaussianas multivariáveis.

A Rede Neural RBF Dinâmica

Em [55] propõe-se uma implementação de uma rede RBF para ser utilizada como um preditor não-linear, na qual o aprendizado ocorre continuamente, em tempo real. Nesta rede existem três parâmetros a serem determinados : os centros, a variância (comum) das Gaussianas e os pesos da camada linear. Estes parâmetros definem completamente a predição de uma amostra $u(n+1)$, calculada no instante n, como sendo

$$u\left(n+1\right)=F\left[\mathbf{u}\left(n\right)\right]=\sum_{k=1}^{K}w_{k}\left(n\right)\exp\left(-\frac{1}{2\sigma^{2}\left(n\right)}\left\|\mathbf{u}\left(n\right)-\mathbf{t}_{n-k}\right\|^{2}\right), \tag{5.75}$$

onde $\mathbf{u}\left(n\right)=\left[u\left(n\right),u\left(n-1\right),\ldots,u\left(n-M\right)\right]^{T}$ está disponível para processamento no instante n e onde M é a ordem de predição. Os parâmetros da rede são descritos como segue :

1. A variância $\sigma^{2}(n)$ das Gaussianas é calculada através da heurística

$$\sigma^{2}\left(n\right)=\alpha\frac{d_{\max}^{2}\left(n\right)}{K}, \tag{5.76}$$

onde $d_{\max}$ é a distância máxima entre os K centros tomados no instante n e α é uma constante positiva, determinada empiricamente.

2. O vetor de coeficientes $\mathbf{w}\left(n\right)=\left[w_{1}(n),w_{2}(n),\ldots,w_{K}(n)\right]$ satisfaz a *condição de interpolação estrita* :

$$\left[\Phi\left(n\right)+\lambda\left(n\right)\mathbf{I}\right]\mathbf{w}\left(n\right)=\mathbf{d}\left(n\right), \tag{5.77}$$

onde a matriz de interpolação $\Phi\left(n\right)$ é definida por

$$\Phi\left(n\right)=\left\{\exp\left(-\frac{1}{2\sigma^{2}\left(n\right)}\left\|\mathbf{u}\left(n-i\right)-\mathbf{u}\left(n-k\right)\right\|^{2}\right)\right\},\quad\left(i,k\right)=1,2,\ldots,K, \tag{5.78}$$

e $\lambda\left(n\right)$ é o parâmetro de regularização[2] no instante n, estimado, tipicamente, através de um janelamento da série temporal ou fixado *a priori*. A resposta desejada $\mathbf{d}(n)$ é definida por

$$\mathbf{d}(n)=\left[u\left(n\right),u\left(n-1\right),\ldots,u\left(n-K+1\right)\right]^{T}. \tag{5.79}$$

[2] O uso de regularização na solução de problemas de interpolação e aproximação já está bem estabelecido. Em termos gerais, a regularização estabiliza a solução de problemas mal condicionados, no sentido que ela torna a solução menos sensível à presença de erros e ruído no conjunto de dados.

A partir desta formulação, podemos perceber que à medida que novos dados da série temporal se tornam disponíveis, o preditor dinâmico RBF evolui do instante de tempo k para o instante de tempo $k+1$ através de um deslocamento dos centros do preditor e o subseqüente cálculo de $\Phi(n), \sigma(n), \lambda(n)$ e $\mathbf{w}(n)$.

A idéia por trás da rede RBF dinâmica é justamente a realização de uma série de aproximações *locais não-lineares* do mapeamento entrada/saída, através de uma atualização contínua das interpolações *exatas* realizadas sobre o janelamento de dados mais recentes da série temporal em questão.

5.4 Algoritmo EM

Durante muitos anos tem sido estudado o problema da estimação de parâmetros em uma mistura de funções de densidade de probabilidade. Nas últimas duas décadas, o método de máxima-verossimilhança tem sido o mais largamente utilizado, graças, fundamentalmente, ao advento dos computadores digitais de alta velocidade. No entanto, apesar da disponibilidade dos computadores digitais, as estimações de máxima-verossimilhança nem sempre são de fácil obtenção. Em particular, para o problema de mistura de densidades de probabilidade, esta dificuldade surge devido às dependências, muitas vezes complexas, da função de verossimilhança com relação aos parâmetros a serem estimados.

A maneira usual de se encontrar uma estimação de máxima-verossimilhança é através da determinação de um sistema de equações, chamadas de *equações de verossimilhança*, que são satisfeitas pelas estimações de máxima-verossimilhança. As equações de verossimilhança são obtidas, usualmente, derivando-se o logaritmo da função de verossimilhança em relação aos parâmetros desejados e impondo-se que as derivadas sejam todas iguais a zero. Para o problema de mistura de densidades de probabilidade, as equações de verossimilhança são, quase sempre, não-lineares e de difícil solução analítica. Conseqüentemente, a maneira usual encontrada para se resolver este problema é a busca de uma solução aproximada através de algum procedimento iterativo.

Existem hoje muitos procedimentos iterativos para se encontrar uma solução aproximada para as equações de verossimilhança, como por exemplo o método de Newton, o método quasi-Newton e os métodos de gradiente conjugado. No entanto, um método iterativo especial, não relacionado aos anteriores, tem sido utilizado nos problemas de mistura de densidades. Seguindo a terminologia de Dempster, Laird e Rubin [14], este método é chamado de **Algoritmo EM** (**E** de "Expectation" e **M** de "Maximization"). Este método tem se destacado por sua robustez, baixo custo por iteração, economia de armazenamento, facilidade de programação e, de certo modo, um certo apelo heurístico. Infelizmente, em certos casos sua convergência pode ser lenta. Antes de estabelecermos o algoritmo EM para o caso de mistura de densidades, introduziremos algumas definições e nomenclaturas.

Estaremos interessados em uma família paramétrica de *misturas finitas de funções de densidade de probabilidade,* isto é, uma família de funções densidade de probabilidade da forma

$$p(\mathbf{x}\,|\Phi) = \sum_{i=1}^{M} \alpha_i p_i(\mathbf{x} \mid \phi_i), \quad \mathbf{x} = [x_1, x_2, \cdots, x_n]^T \in \mathbb{R}^n, \tag{5.80}$$

onde cada α_i é não-negativo, $\sum_{i=1}^{M} \alpha_i = 1$ e cada $p_i(\cdot)$ é uma função densidade de probabilidade parametrizada por $\phi_i \in \Omega_i \subseteq \mathbb{R}^{n_i}$.

Definimos $\Phi = (\alpha_1, \cdots, \alpha_M, \phi_1, \cdots, \phi_M)$ e fazemos

$$\Omega = \left\{ (\alpha_1, \cdots, \alpha_M, \phi_1, \cdots, \phi_M) : \sum_{i=1}^{M} \alpha_i = 1 \text{ e } \alpha_i \geq 0,\ \phi_i \in \Omega_i \text{ para } i = 1, \ldots, M \right\}. \quad (5.81)$$

Em particular, estaremos interessados no caso em que cada densidade $p_i(\mathbf{x} \mid \phi_i)$ é uma função densidade de probabilidade *normal multivariável*, isto é, $p_i(\mathbf{x} \mid \phi_i)$ e ϕ_i são dados por

$$p_i(\mathbf{x} \mid \phi_i) = \frac{1}{(2\pi)^{n/2} |\Lambda_i|^{1/2}} \exp\left[-\frac{1}{2} (\mathbf{x} - \boldsymbol{\mu}_i)^T \Lambda_i^{-1} (\mathbf{x} - \boldsymbol{\mu}_i) \right], \quad \phi_i = (\boldsymbol{\mu}_i, \Lambda_i), \quad (5.82)$$

onde $\boldsymbol{\mu}_i \in \mathbb{R}^n$ e Λ_i é uma matriz $n \times n$ simétrica positiva definida.

O nosso objetivo será estimar os valores reais $\alpha_1^*, \cdots, \alpha_M^*, \phi_1^*, \cdots, \phi_M^*$ dos parâmetros da mistura $p(\mathbf{x} \mid \Phi)$ a partir de um conjunto de observações $\{\mathbf{x}_k\}_{k=1,\ldots,N}$ representativas desta mistura. Para isto, adotaremos algum critério de otimização, segundo o qual, as estimações serão ditas ótimas. O critério adotado será a *maximização da função de verossimilhança.*

Por *função de verossimilhança* de uma amostra de observações $\mathcal{X} = \{\mathbf{x}_k\}_{k=1,\ldots,N}$, entende-se a função densidade de probabilidade da variável amostral $\mathcal{X}$, calculada sobre os valores das observações obtidas, ou seja,

$$L(\Phi) = p(\mathbf{x}_{1,\ldots,}\mathbf{x}_N \mid \Phi), \quad (5.83)$$

onde Φ é o conjunto de parâmetros que se deseja estimar e $L(\Phi)$ pode ser entendida como uma função de Φ para $\{\mathbf{x}_k\}_{k=1,\ldots,N}$ fixado.

Quando se deseja obter estimativas de máxima-verossimilhança, costuma-se trabalhar com o logaritmo da função de verossimilhança. Se as observações $\{\mathbf{x}_k\}_{k=1,\ldots,N}$ forem variáveis aleatórias i.i.d., teremos então

$$L(\Phi) = \ln p(\mathbf{x}_{1,\ldots,}\mathbf{x}_N \mid \Phi) = \ln \prod_{k=1}^{N} p(\mathbf{x}_k \mid \Phi) = \sum_{k=1}^{N} \ln p(\mathbf{x}_k \mid \Phi). \quad (5.84)$$

A técnica de máxima verossimilhança consiste, portanto, em obter o valor de Φ que maximiza $L(\Phi)$.

A seguir, enunciaremos a formulação geral do **algoritmo EM** introduzida por Dempster, Laird e Rubin [14] para obtenção de estimações de máxima-verossimilhança de *dados incompletos.* A partir deste algoritmo *geral*, vamos expor o caso mais particular do algoritmo EM para o problema de estimação de parâmetros de mistura de densidades.

5.4.1 O Algoritmo EM

Inicialmente, supõe-se a existência de dois espaços amostrais $\mathcal{X}$ e $\mathcal{Y}$ e um mapeamento de $\mathcal{Y}$ para $\mathcal{X}$. Os *dados observados* $\mathbf{x}$ correspondem a uma realização de $\mathcal{X}$. Os dados $\mathbf{y}$ em $\mathcal{Y}$ são

observados apenas *indiretamente* através de $\mathbf{x}$. Mais especificamente, assume-se que existe um mapeamento $\mathbf{y} \rightarrow \mathbf{x}(\mathbf{y})$ de $\mathcal{Y}$ para $\mathcal{X}$, através do qual $\mathbf{y}$ encontra-se em $\mathcal{Y}(\mathbf{x})$, o subconjunto de $\mathcal{Y}$ determinado pela equação $\mathbf{x} = \mathbf{x}(\mathbf{y})$, e onde $\mathbf{x}$ é o dado *observado*. Referimo-nos a $\mathbf{y}$ como o **dado completo**.

Postula-se então uma família de funções de densidade de probabilidade amostrais $f(\mathbf{y} \mid \Phi)$, as quais dependem de um conjunto Φ de parâmetros, e deriva-se a família correspondente de densidades amostrais $g(\mathbf{x} \mid \Phi)$. A especificação dos dados completos $f(\mathbf{y} \mid \Phi)$ está relacionada à especificação dos dados incompletos $g(\mathbf{x} \mid \Phi)$ por :

$$g(\mathbf{x} \mid \Phi) = \int_{Y(\mathbf{x})} f(\mathbf{y} \mid \Phi)\, d\mathbf{y}. \tag{5.85}$$

Para um dado $\mathbf{x} \in \mathcal{X}$, o objetivo do algoritmo EM é maximizar a função de log-verossimilhança dos dados **incompletos** $L(\Phi) = \ln g(\mathbf{x} \mid \Phi)$ sobre $\Phi \in \Omega$, através do uso essencial de $f(\mathbf{y} \mid \Phi)$.

Para $\mathbf{x} \in \mathcal{X}$, seja $\mathcal{Y}(\mathbf{x}) = \{\mathbf{y} \in \mathcal{Y} : \mathbf{x}(\mathbf{y}) = \mathbf{x}\}$. A densidade condicional $k(\mathbf{y} \mid \mathbf{x}, \Phi)$em $\mathcal{Y}(\mathbf{x})$ é dada por

$$k(\mathbf{y} \mid \mathbf{x}, \Phi) = \frac{f(\mathbf{y} \mid \Phi)}{g(\mathbf{x} \mid \Phi)}. \tag{5.86}$$

Aplicando o logaritmo em ambos os membros, teremos

$$\ln f(\mathbf{y} \mid \Phi) = \ln k(\mathbf{y} \mid \mathbf{x}, \Phi) + \ln g(\mathbf{x} \mid \Phi). \tag{5.87}$$

Se aplicarmos o operador *esperança condicional* $E[\cdot \mid \mathbf{x}, \Phi']$, teremos

$$E[\ln f(\mathbf{y} \mid \Phi) \mid \mathbf{x}, \Phi'] = E[\ln k(\mathbf{y} \mid \mathbf{x}, \Phi) \mid \mathbf{x}, \Phi'] + E[\ln g(\mathbf{x} \mid \Phi) \mid \mathbf{x}, \Phi'], \tag{5.88}$$

onde Φ' é uma aproximação para Φ. Rearranjando os termos,

$$E[\ln g(\mathbf{x} \mid \Phi) \mid \mathbf{x}, \Phi'] = E[\ln f(\mathbf{y} \mid \Phi) \mid \mathbf{x}, \Phi'] - E[\ln k(\mathbf{y} \mid \mathbf{x}, \Phi) \mid \mathbf{x}, \Phi']. \tag{5.89}$$

Podemos reescrever a Eq. (5.89) na forma

$$L(\Phi) = Q(\Phi \mid \Phi') - H(\Phi \mid \Phi'), \tag{5.90}$$

onde

$$Q(\Phi \mid \Phi') = E[\ln f(\mathbf{y} \mid \Phi) \mid \mathbf{x}, \Phi'] \tag{5.91}$$

e

$$H(\Phi \mid \Phi') = E[\ln k(\mathbf{y} \mid \mathbf{x}, \Phi) \mid \mathbf{x}, \Phi']. \tag{5.92}$$

Dadas essas definições e nomenclaturas, o algoritmo EM **geral** de Dempster, Laird e Rubin [14] pode ser enunciado da seguinte forma :

Dada uma aproximação atual $\Phi^{(p)}$ de um maximizador Φ^* de $L(\Phi)$, obtemos a próxima aproximação $\Phi^{(p+1)}$ da seguinte forma :

1. **Passo E** ("**Expectation**") : Determine $Q\left(\Phi \mid \Phi^{(p)}\right)$;
2. **Passo M** ("**Maximization**") : Escolher $\Phi^{(p+1)} \in \arg\max_{\Phi\in\Omega} Q\left(\Phi \mid \Phi^{(p)}\right)$, onde a expressão $\arg\max_{\Phi\in\Omega} Q\left(\Phi \mid \Phi^{(p)}\right)$ significa o conjunto de valores $\Phi \in \Omega$ que maximizam $Q\left(\Phi \mid \Phi^{(p)}\right)$ em Ω.

A fim de expormos as condições de convergência do algoritmo EM, introduzimos a notação $\Phi \longrightarrow M(\Phi)$ de Ω em Ω para representarmos o mapeamento definido pelo algoritmo EM ao realizar o passo "M". Assim, cada passo $\Phi^{(p)} \longrightarrow \Phi^{(p+1)}$ é definido por

$$\Phi^{(p+1)} = M\left(\Phi^{(p)}\right).$$

Teorema 5.1 *Para todo algoritmo EM,*

$$L(M(\Phi)) \geq L(\Phi) \quad \textit{para todo } \Phi \in \Omega,$$

ocorrendo a igualdade se, e somente se, $Q(M(\Phi) \mid \Phi) = Q(\Phi \mid \Phi)$ *e* $k(\mathbf{y} \mid \mathbf{x}, M(\Phi)) = k(\mathbf{y} \mid \mathbf{x}, \Phi)$.

Corolário 5.1 *Suponha, para algum* $\Phi^* \in \Omega$, *que* $L(\Phi^*) \geq L(\Phi)$ *para todo* $\Phi \in \Omega$. *Então, para todo algoritmo EM,*

1. $L(M(\Phi^*)) = L(\Phi^*)$,
2. $Q(M(\Phi^*) \mid \Phi^*) = Q(\Phi^* \mid \Phi^*)$,
3. $k(\mathbf{y} \mid \mathbf{x}, M(\Phi^*)) = k(\mathbf{y} \mid \mathbf{x}, \Phi^*)$.

Corolário 5.2 *Se, para algum* $\Phi^* \in \Omega$, $L(\Phi^*) > L(\Phi)$ *para todo* $\Phi \in \Omega$ *tal que* $\Phi \neq \Phi^*$ *então, para todo algoritmo EM,*

$$M(\Phi^*) = \Phi^*.$$

Teorema 5.2 *Suponha que* $\Phi^{(p)}$ *para* $p = 0, 1, 2, \ldots$ *seja uma seqüência de estimativas geradas pelo algoritmo EM tal que :*

1. *a seqüência* $L\left(\Phi^{(p)}\right)$ *seja limitada, e*
2. $Q\left(\Phi^{(p+1)} \mid \Phi^{(p)}\right) - Q\left(\Phi^{(p)} \mid \Phi^{(p)}\right) \geq \lambda\left(\Phi^{(p+1)} - \Phi^{(p)}\right)\left(\Phi^{(p+1)} - \Phi^{(p)}\right)^T$ *para algum escalar* $\lambda > 0$ *e para todo* p.
   Então a seqüência $\Phi^{(p)}$ *converge para algum* Φ^* *em* Ω.

O teorema 5.1 afirma que $L(\Phi)$ é não decrescente em cada iteração do algoritmo EM e estritamente crescente em cada iteração onde $Q\left(\Phi^{(p+1)} \mid \Phi^{(p)}\right) > Q\left(\Phi^{(p)} \mid \Phi^{(p)}\right)$. Os corolários 5.1 e 5.2 afirmam que uma estimativa de máxima-verossimilhança é um ponto fixo de um algoritmo EM, e o teorema 5.2 fornece as condições sob as quais o algoritmo EM converge. No entanto, estes teoremas não implicam em convergência para um estimativa de máxima-verossimilhança. Isto é possível assumindo-se condições bastante gerais, as quais não serão expostas aqui. Para maiores referências, sugerimos consultar [14, 53].

A idéia heurística por trás do algoritmo EM está no fato de que, como não temos o conjunto de dados completo, não podemos calcular $\ln f(\mathbf{y} \mid \Phi)$ diretamente. Ao invés disso, calculamos o valor *esperado* dadas as observações $\{\mathbf{x}_k\}_{k=1,\ldots,N}$ e uma aproximação $\Phi^{(p)}$ do valor verdadeiro Φ^*.

5.4.2 Algoritmo EM para Mistura de Densidades de Probabilidade

Para introduzirmos o algoritmo EM em problemas de estimação de parâmetros de mistura de densidades de probabilidade, temos que assumir que uma família paramétrica de mistura de densidades é especificada previamente e que um particular $\Phi^* = (\alpha_1^*, \cdots, \alpha_M^*, \phi_1^*, \cdots, \phi_M^*)$ é considerado o valor "verdadeiro" a ser estimado. O algoritmo EM para um problema de estimação de mistura de densidades é obtido a partir da interpretação deste problema como um caso típico de ocorrência de *dados incompletos*. A partir daí, o algoritmo é obtido de sua formulação geral dada anteriormente.

No caso de mistura de densidades, considera-se que cada observação da amostra encontra-se "não-rotulada", no sentido de que não se sabe, a priori, a densidade componente de origem de cada observação. Podemos considerar $\{\mathbf{x}_k\}_{k=1,\ldots,N}$ como uma amostra de dados incompletos, assumindo que cada $\mathbf{x}_k$ corresponde à parte "conhecida" de uma observação $\mathbf{y}_k = (\mathbf{x}_k, i_k)$, onde i_k é um inteiro entre 1 e M indicando a densidade componente de origem.

Para $\Phi = (\alpha_1, \cdots, \alpha_M, \phi_1, \cdots, \phi_M) \in \Omega$, as variáveis amostrais $\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N)$ e $\mathbf{y} = (\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_N)$ possuem densidades de probabilidade associadas

$$g(\mathbf{x} \mid \Phi) = \prod_{k=1}^{N} p(\mathbf{x}_k \mid \Phi) \tag{5.93}$$

e

$$f(\mathbf{y} \mid \Phi) = \prod_{k=1}^{N} \alpha_{i_k} p_{i_k}(\mathbf{x}_k \mid \phi_{i_k}), \tag{5.94}$$

onde assumimos a independência das observações $\mathbf{x}_k$ e $\mathbf{y}_k$.

Então, para $\Phi' = (\alpha_1', \cdots, \alpha_M', \phi_1', \cdots, \phi_M') \in \Omega$, a densidade condicional $k(\mathbf{y} \mid \mathbf{x}, \Phi')$ será dada por

$$k(\mathbf{y} \mid \mathbf{x}, \Phi') = \prod_{k=1}^{N} \frac{\alpha_{i_k}' p_{i_k}(\mathbf{x}_k \mid \phi_{i_k}')}{p(\mathbf{x}_k \mid \Phi')}. \tag{5.95}$$

A esperança de $\ln f(\mathbf{y} \mid \Phi)$ é obtida somando-se $\ln f(\mathbf{y} \mid \Phi)$ sobre todos os valores possíveis de $\{i_k\}$, com o fator de ponderação dado pela densidade $k(\mathbf{y} \mid \mathbf{x}, \Phi')$. Logo,

$$E[\ln f(\mathbf{y} \mid \Phi) \mid \mathbf{x}, \Phi'] = \sum_{i_1=1}^{M} \sum_{i_2=1}^{M} \cdots \sum_{i_N=1}^{M} \sum_{k=1}^{N} \ln \alpha_{i_k} p_{i_k}(\mathbf{x}_k \mid \phi_{i_k}) \prod_{k=1}^{N} \frac{\alpha_{i_k}' p_{i_i}(\mathbf{x}_k \mid \phi_{i_k}')}{p(\mathbf{x}_k \mid \Phi')}. \tag{5.96}$$

Se tomarmos

$$\sum_{k=1}^{N} \ln \alpha_{i_k} p_{i_i}(\mathbf{x}_k \mid \phi_{i_k}) = \sum_{k=1}^{N} \sum_{i=1}^{M} \delta_{i i_k} \ln \alpha_i p_i(\mathbf{x}_k \mid \phi_i), \tag{5.97}$$

onde

$$\delta_{i i_k} = \begin{cases} 1, & \text{se } i = i_k \\ 0, & \text{se } i \neq i_k \end{cases} \tag{5.98}$$

e substituirmos na Eq. (5.96), teremos

$$\begin{aligned} E\left[\ln f\left(\mathbf{y} \mid \Phi\right) \mid \mathbf{x}, \Phi'\right] &= \sum_{i_1=1}^{M}\sum_{i_2=1}^{M}\cdots\sum_{i_N=1}^{M}\left[\sum_{k=1}^{N}\sum_{i=1}^{M}\delta_{ii_k}\ln\alpha_i p_i(\mathbf{x}_k \mid \phi_i)\right]\prod_{k=1}^{N}\frac{\alpha'_{i_k}p_{i_k}(\mathbf{x}_k \mid \phi'_{i_k})}{p(\mathbf{x}_k \mid \Phi')} \\ &= \sum_{k=1}^{N}\sum_{i=1}^{M}\left[\sum_{i_1=1}^{M}\sum_{i_2=1}^{M}\cdots\sum_{i_N=1}^{M}\delta_{ii_k}\ln\alpha_i p_i(\mathbf{x}_k \mid \phi_i)\prod_{k=1}^{N}\frac{\alpha'_{i_k}p_{i_k}(\mathbf{x}_k \mid \phi'_{i_k})}{p(\mathbf{x}_k \mid \Phi')}\right] \\ &= \sum_{k=1}^{N}\sum_{i=1}^{M}\ln\alpha_i p_i(\mathbf{x}_k \mid \phi_i)\frac{\alpha'_i p_i(\mathbf{x}_k \mid \phi'_i)}{p(\mathbf{x}_k \mid \Phi')}. \end{aligned} \tag{5.99}$$

Tendo sido obtida uma função apropriada $Q\left(\Phi \mid \Phi'\right)$ para o passo **E**, devemos partir agora para o passo **M**. Podemos observar da expressão acima que o problema de maximização de $Q\left(\Phi \mid \Phi'\right)$ pode ser decomposto em dois problemas separados de maximização : o primeiro envolvendo as proporções α_i e o segundo envolvendo apenas os parâmetros ϕ_i, como pode ser visto na expressão a seguir, obtida da equação anterior :

$$\begin{aligned} E\left[\ln f\left(\mathbf{y} \mid \Phi\right) \mid \mathbf{x}, \Phi'\right] &= \sum_{i=1}^{M}\left[\sum_{k=1}^{N}\frac{\alpha'_i p_i(\mathbf{x}_k \mid \phi'_i)}{p(\mathbf{x}_k \mid \Phi')}\right]\ln\alpha_i + \\ \sum_{k=1}^{N}\sum_{i=1}^{M}\ln p_i(\mathbf{x}_k &\mid \phi_i)\frac{\alpha'_i p_i(\mathbf{x}_k \mid \phi'_i)}{p(\mathbf{x}_k \mid \Phi')} = Q_1 + Q_2. \end{aligned} \tag{5.100}$$

A fim de maximizarmos $Q\left(\Phi \mid \Phi'\right)$ em relação às proporções α_i, devemos tomar as derivadas de Q_1 em relação aos α_i's, ou seja,

$$\begin{aligned} \frac{\partial}{\partial\alpha_i}\left[\sum_{i=1}^{M}\sum_{k=1}^{N}\frac{\alpha'_i p_i(\mathbf{x}_k \mid \phi'_i)}{p(\mathbf{x}_k \mid \Phi')}\ln\alpha_i\right] &= \sum_{k=1}^{N}\frac{\partial}{\partial\alpha_i}\frac{\alpha'_i p_i(\mathbf{x}_k \mid \phi'_i)}{p(\mathbf{x}_k \mid \Phi')}\ln\alpha_i \\ &= \sum_{k=1}^{N}\frac{\alpha'_i p_i(\mathbf{x}_k \mid \phi'_i)}{p(\mathbf{x}_k \mid \Phi')}\frac{1}{\alpha_i}. \end{aligned} \tag{5.101}$$

No entanto, devemos levar em consideração a restrição de que $\sum_{i=1}^{M}\alpha_i = 1$. Introduzindo o multiplicador de lagrange λ, e maximizando a função

$$Q' = \sum_{k=1}^{N}\frac{\alpha'_i p_i(\mathbf{x}_k \mid \phi'_i)}{p(\mathbf{x}_k \mid \Phi')}\ln\alpha_i + \lambda\left(\sum_{i=1}^{M}\alpha_i - 1\right) \tag{5.102}$$

teremos

$$\frac{\partial Q'}{\partial\alpha_i} = \sum_{k=1}^{N}\frac{\alpha'_i p_i(\mathbf{x}_k \mid \phi'_i)}{p(\mathbf{x}_k \mid \Phi')}\frac{1}{\alpha_i} + \lambda. \tag{5.103}$$

Fazendo $\frac{\partial Q'}{\partial \alpha_i} = 0$, teremos

$$\lambda = -\sum_{k=1}^{N} \frac{\alpha_i' p_i(\mathbf{x}_k \mid \phi_i')}{p(\mathbf{x}_k \mid \Phi')} \frac{1}{\alpha_i} \Rightarrow \sum_{i=1}^{M} \lambda \alpha_i = -\sum_{i=1}^{M} \sum_{k=1}^{N} \frac{\alpha_i' p_i(\mathbf{x}_k \mid \phi_i')}{p(\mathbf{x}_k \mid \Phi')}, \tag{5.104}$$

de onde

$$\lambda = -\sum_{i=1}^{M} \sum_{k=1}^{N} \frac{\alpha_i' p_i(\mathbf{x}_k \mid \phi_i')}{p(\mathbf{x}_k \mid \Phi')} \Rightarrow \lambda = -\sum_{k=1}^{N} 1 = -N.$$

Substituindo o valor de λ, na Eq. (5.103) e igualando a zero, teremos

$$\sum_{k=1}^{N} \frac{\alpha_i' p_i(\mathbf{x}_k \mid \phi_i')}{p(\mathbf{x}_k \mid \Phi')} \frac{1}{\alpha_i} - N = 0 \tag{5.105}$$

Portanto,

$$\alpha_i = \frac{1}{N} \sum_{k=1}^{N} \frac{\alpha_i' p_i(\mathbf{x}_k \mid \phi_i')}{p(\mathbf{x}_k \mid \Phi')}. \tag{5.106}$$

Se $\phi_1, \cdots, \phi_M$ forem mutuamente independentes, o segundo problema de maximização reduz-se a M problemas distintos, cada um dos quais envolvendo apenas um dos parâmetros ϕ_i. Neste caso, como ainda não temos a forma funcional das densidades componentes, podemos afirmar que

$$\phi_i \in \arg \max_{\phi_i \in \Omega_i} \sum_{k=1}^{N} \ln p_i(\mathbf{x}_k \mid \phi_i) \frac{\alpha_i' p_i(\mathbf{x}_k \mid \phi_i')}{p(\mathbf{x}_k \mid \Phi')} \tag{5.107}$$

5.4.3 Algoritmo EM para Mistura de Gaussianas Multivariáveis

Consideramos agora o caso em que cada densidade $p_i(\mathbf{x} \mid \phi_i)$ é dada por

$$p_i(\mathbf{x} \mid \phi_i) = \frac{1}{(2\pi)^{n/2} |\Lambda_i|^{1/2}} \exp\left[-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_i)^T \Lambda_i^{-1} (\mathbf{x} - \boldsymbol{\mu}_i)\right], \quad \phi_i = (\boldsymbol{\mu}_i, \Lambda_i), \tag{5.108}$$

onde $\boldsymbol{\mu}_i \in \mathbb{R}^n$ e Λ_i é uma matriz $n \times n$ simétrica positiva definida.

Desejamos encontrar $\phi_i = (\boldsymbol{\mu}_i, \Lambda_i)$ que maximize o segundo termo de (5.100), ou seja,

$$Q_2 = \sum_{k=1}^{N} \ln p_i(\mathbf{x}_k \mid \phi_i) \frac{\alpha_i' p_i(\mathbf{x}_k \mid \phi_i')}{p(\mathbf{x}_k \mid \Phi')}. \tag{5.109}$$

Substituindo pela densidade normal multivariável, teremos

$$Q_2 = \sum_{k=1}^{N} \ln \left\{ \frac{1}{(2\pi)^{n/2} |\Lambda_i|^{1/2}} \exp \left[-\frac{1}{2} (\mathbf{x}_k - \boldsymbol{\mu}_i)^T \Lambda_i^{-1} (\mathbf{x}_k - \boldsymbol{\mu}_i) \right] \right\} \cdot \frac{\alpha_i' p_i(\mathbf{x}_k \mid \phi_i')}{p(\mathbf{x}_k \mid \Phi')}. \quad (5.110)$$

Logo,

$$\begin{aligned} Q_2 \;=\; & -\frac{1}{2} \sum_{k=1}^{N} \ln |\Lambda_i| \cdot \frac{\alpha_i' p_i(\mathbf{x}_k \mid \phi_i')}{p(\mathbf{x}_k \mid \Phi')} - \\ & \frac{1}{2} \sum_{k=1}^{N} (\mathbf{x}_k - \boldsymbol{\mu}_i)^T \Lambda_i^{-1} (\mathbf{x}_k - \boldsymbol{\mu}_i) \cdot \frac{\alpha_i' p_i(\mathbf{x}_k \mid \phi_i')}{p(\mathbf{x}_k \mid \Phi')} + const. \end{aligned} \quad (5.111)$$

A fim de encontrarmos os parâmetros $\boldsymbol{\mu}_i$ e Λ_i^{-1} que maximizam Q_2, fazemos

$$\frac{\partial Q_2}{\partial \boldsymbol{\mu}_i} = \left(\frac{\partial Q_2}{\partial \mu_i^1}, \frac{\partial Q_2}{\partial \mu_i^2}, \cdots, \frac{\partial Q_2}{\partial \mu_i^n} \right) = \mathbf{0} \quad (5.112)$$

e

$$\frac{\partial Q_2}{\partial \Lambda_i^{-1}} = \begin{bmatrix} \frac{\partial Q_2}{\partial \lambda_{11}^i} & \frac{\partial Q_2}{\partial \lambda_{12}^i} & \cdots & \frac{\partial Q_2}{\partial \lambda_{1n}^i} \\ \frac{\partial Q_2}{\partial \lambda_{21}^i} & \frac{\partial Q_2}{\partial \lambda_{22}^i} & \cdots & \frac{\partial Q_2}{\partial \lambda_{2n}^i} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial Q_2}{\partial \lambda_{n1}^i} & \frac{\partial Q_2}{\partial \lambda_{n2}^i} & \cdots & \frac{\partial Q_2}{\partial \lambda_{nn}^i} \end{bmatrix} = \mathbf{0}, \quad (5.113)$$

onde $\Lambda_i^{-1} = \left[\lambda_{rs}^i \right]$.

Encontremos inicialmente os valores de μ_i^j que maximizam Q_2

$$\begin{aligned} \frac{\partial Q_2}{\partial \mu_i^j} \;=\; & \frac{\partial}{\partial \mu_i^j} \left[-\frac{1}{2} \sum_{k=1}^{N} \ln |\Lambda_i| \cdot \frac{\alpha_i' p_i(\mathbf{x}_k \mid \phi_i')}{p(\mathbf{x}_k \mid \Phi')} - \right. \\ & \left. \frac{1}{2} \sum_{k=1}^{N} (\mathbf{x}_k - \boldsymbol{\mu}_i)^T \Lambda_i^{-1} (\mathbf{x}_k - \boldsymbol{\mu}_i) \cdot \frac{\alpha_i' p_i(\mathbf{x}_k \mid \phi_i')}{p(\mathbf{x}_k \mid \Phi')} + const. \right] \end{aligned} \quad (5.114)$$

de onde

$$\frac{\partial Q_2}{\partial \mu_i^j} = \frac{\partial}{\partial \mu_i^j} \left[-\frac{1}{2} \sum_{k=1}^{N} (\mathbf{x}_k - \boldsymbol{\mu}_i)^T \Lambda_i^{-1} (\mathbf{x}_k - \boldsymbol{\mu}_i) \cdot \frac{\alpha_i' p_i(\mathbf{x}_k \mid \phi_i')}{p(\mathbf{x}_k \mid \Phi')} \right]. \quad (5.115)$$

Mas,

$$(\mathbf{x}_k - \boldsymbol{\mu}_i)^T \Lambda_i^{-1} (\mathbf{x}_k - \boldsymbol{\mu}_i) = \sum_{r=1}^{n} \sum_{s=1}^{n} (x_k^r - \mu_i^r) \lambda_{rs}^i (x_k^s - \mu_i^s). \quad (5.116)$$

Substituindo em (5.115),

$$\frac{\partial Q_2}{\partial \mu_i^j} = \frac{\partial}{\partial \mu_i^j}\left[-\frac{1}{2}\sum_{k=1}^{N}\sum_{r=1}^{n}\sum_{s=1}^{n}\left(x_k^r-\mu_i^r\right)\lambda_{rs}^i\left(x_k^s-\mu_i^s\right)\cdot\frac{\alpha_i' p_i(\mathbf{x}_k \mid \phi_i')}{p(\mathbf{x}_k \mid \Phi')}\right] \tag{5.117}$$
$$= -\frac{1}{2}\sum_{k=1}^{N}\frac{\partial}{\partial \mu_i^j}\left[\sum_{r=1}^{n}\sum_{s=1}^{n}\left(x_k^r-\mu_i^r\right)\lambda_{rs}^i\left(x_k^s-\mu_i^s\right)\right]\cdot\frac{\alpha_i' p_i(\mathbf{x}_k \mid \phi_i')}{p(\mathbf{x}_k \mid \Phi')}.$$

Calculando a derivada,

$$\frac{\partial}{\partial \mu_i^j}\left[\sum_{r=1}^{n}\sum_{s=1}^{n}\left(x_k^r-\mu_i^r\right)\lambda_{rs}^i\left(x_k^s-\mu_i^s\right)\right] = -\sum_{r=1}^{n}\left(x_k^r-\mu_i^r\right)\lambda_{rj}^i - \sum_{s=1}^{n}\lambda_{js}^i\left(x_k^s-\mu_i^s\right). \tag{5.118}$$

Se a matriz Λ_i^{-1} for simétrica,

$$\frac{\partial}{\partial \mu_i^j}\left[\left(\mathbf{x}_k-\mu_i\right)^T\Lambda_i^{-1}\left(\mathbf{x}_k-\mu_i\right)\right] = -2\sum_{r=1}^{n}\left(x_k^r-\mu_i^r\right)\lambda_{jr}^i. \tag{5.119}$$

Substituindo de volta na Eq.(5.117),

$$\frac{\partial Q_2}{\partial \mu_i^j} = \sum_{k=1}^{N}\sum_{r=1}^{n}\left(x_k^r-\mu_i^r\right)\lambda_{jr}^i\cdot\frac{\alpha_i' p_i(\mathbf{x}_k \mid \phi_i')}{p(\mathbf{x}_k \mid \Phi')}. \tag{5.120}$$

Q_2 será máximo quando $\frac{\partial Q_2}{\partial \mu_i^j} = 0$. Logo,

$$\frac{\partial Q_2}{\partial \mu_i^j} = 0 \Rightarrow \sum_{k=1}^{N}\sum_{r=1}^{n}x_k^r\lambda_{jr}^i\cdot\frac{\alpha_i' p_i(\mathbf{x}_k \mid \phi_i')}{p(\mathbf{x}_k \mid \Phi')} = \sum_{k=1}^{N}\sum_{r=1}^{n}\mu_i^r\lambda_{jr}^i\cdot\frac{\alpha_i' p_i(\mathbf{x}_k \mid \phi_i')}{p(\mathbf{x}_k \mid \Phi')}. \tag{5.121}$$

Para que haja igualdade devemos ter, para todo r,

$$\sum_{k=1}^{N}x_k^r\lambda_{jr}^i\cdot\frac{\alpha_i' p_i(\mathbf{x}_k \mid \phi_i')}{p(\mathbf{x}_k \mid \Phi')} = \sum_{k=1}^{N}\mu_i^r\lambda_{jr}^i\cdot\frac{\alpha_i' p_i(\mathbf{x}_k \mid \phi_i')}{p(\mathbf{x}_k \mid \Phi')}. \tag{5.122}$$

Portanto,

$$\mu_i^r = \frac{\sum_{k=1}^{N}x_k^r\cdot\frac{\alpha_i' p_i(\mathbf{x}_k|\phi_i')}{p(\mathbf{x}_k|\Phi')}}{\sum_{k=1}^{N}\frac{\alpha_i' p_i(\mathbf{x}_k|\phi_i')}{p(\mathbf{x}_k|\Phi')}}. \tag{5.123}$$

Expressando em termos de todas as componentes,

$$\boldsymbol{\mu}_i = \frac{\sum_{k=1}^{N}\mathbf{x}_k\cdot\frac{\alpha_i' p_i(\mathbf{x}_k|\phi_i')}{p(\mathbf{x}_k|\Phi')}}{\sum_{k=1}^{N}\frac{\alpha_i' p_i(\mathbf{x}_k|\phi_i')}{p(\mathbf{x}_k|\Phi')}}. \tag{5.124}$$

Calcularemos agora a derivada de Q_2 em relação aos elementos da matriz Λ_i^{-1}. Ou seja,

$$\frac{\partial Q_2}{\partial \lambda_{rs}^i} = \frac{\partial}{\partial \lambda_{rs}^i}\left[-\frac{1}{2}\sum_{k=1}^{N}\ln|\Lambda_i|\cdot\frac{\alpha_i' p_i(\mathbf{x}_k \mid \phi_i')}{p(\mathbf{x}_k \mid \Phi')} - \right. \tag{5.125}$$
$$\left. \frac{1}{2}\sum_{k=1}^{N}(\mathbf{x}_k-\boldsymbol{\mu}_i)^T\Lambda_i^{-1}(\mathbf{x}_k-\boldsymbol{\mu}_i)\cdot\frac{\alpha_i' p_i(\mathbf{x}_k \mid \phi_i')}{p(\mathbf{x}_k \mid \Phi')} + const.\right].$$

Tomando o primeiro somatório,

$$\frac{\partial}{\partial \lambda_{rs}^i}\left[-\frac{1}{2}\sum_{k=1}^{N}\ln|\Lambda_i|\cdot\frac{\alpha_i' p_i(\mathbf{x}_k \mid \phi_i')}{p(\mathbf{x}_k \mid \Phi')}\right] = -\frac{1}{2}\sum_{k=1}^{N}\left[\frac{\partial \ln|\Lambda_i|}{\partial \lambda_{rs}^i}\right]\cdot\frac{\alpha_i' p_i(\mathbf{x}_k \mid \phi_i')}{p(\mathbf{x}_k \mid \Phi')} \tag{5.126}$$
$$= -\frac{1}{2}\sum_{k=1}^{N}\left[\frac{1}{|\Lambda_i|}\frac{\partial|\Lambda_i|}{\partial \lambda_{rs}^i}\right]\cdot\frac{\alpha_i' p_i(\mathbf{x}_k \mid \phi_i')}{p(\mathbf{x}_k \mid \Phi')}.$$

Para qualquer matriz quadrada $\mathbf{A}$,

$$\mathbf{A}\,(\text{adj }\mathbf{A}) = (\text{adj }\mathbf{A})\,\mathbf{A} = |\mathbf{A}|\,\mathbf{I}, \tag{5.127}$$

onde $\mathbf{I}$ é a matriz identidade. Assim, se $|\mathbf{A}| \neq 0$,

$$\mathbf{A}^{-1} = \frac{1}{|\mathbf{A}|}\,\text{adj }\mathbf{A}, \tag{5.128}$$

onde adj $\mathbf{A}$ é a transposta da matriz dos cofatores dos elementos a_{ij} de $\mathbf{A}$. Sendo assim, podemos escrever o elemento λ_{rs}^i da matriz inversa Λ_i^{-1} como

$$\lambda_{rs}^i = \frac{\Lambda_{sr}^i}{|\Lambda_i|}, \tag{5.129}$$

onde Λ_{rs}^i é o cofator do elemento ρ_{sr} da matriz Λ_i. Logo, $|\Lambda_i| = \frac{\Lambda_{sr}^i}{\lambda_{rs}^i}$. Derivando em relação a λ_{rs}^i, teremos

$$-\frac{1}{2}\sum_{k=1}^{N}\left[\frac{1}{|\Lambda_i|}\frac{\partial}{\partial \lambda_{rs}^i}\left(\frac{\Lambda_{sr}^i}{\lambda_{rs}^i}\right)\right]\cdot\frac{\alpha_i' p_i(\mathbf{x}_k \mid \phi_i')}{p(\mathbf{x}_k \mid \Phi')} = -\frac{1}{2}\sum_{k=1}^{N}\left[-\frac{1}{|\Lambda_i|}\cdot\frac{\Lambda_{sr}^i}{\left(\lambda_{rs}^i\right)^2}\right]\cdot\frac{\alpha_i' p_i(\mathbf{x}_k \mid \phi_i')}{p(\mathbf{x}_k \mid \Phi')}$$
$$= \frac{1}{2}\sum_{k=1}^{N}\frac{1}{\lambda_{rs}^i}\cdot\frac{\alpha_i' p_i(\mathbf{x}_k \mid \phi_i')}{p(\mathbf{x}_k \mid \Phi')}. \tag{5.130}$$

Tomando o segundo somatório de (5.125),

$$\begin{aligned}
&\frac{\partial}{\partial \lambda_{rs}^{i}}\left[-\frac{1}{2}\sum_{k=1}^{N}\left(\mathbf{x}_k-\boldsymbol{\mu}_i\right)^T\Lambda_i^{-1}\left(\mathbf{x}_k-\boldsymbol{\mu}_i\right)\cdot\frac{\alpha_i' p_i(\mathbf{x}_k\mid\phi_i')}{p(\mathbf{x}_k\mid\Phi')}\right]\\
=\;&-\frac{1}{2}\sum_{k=1}^{N}\frac{\partial}{\partial\lambda_{rs}^{i}}\left[\sum_{l=1}^{n}\sum_{m=1}^{n}\left(x_k^l-\mu_i^l\right)\lambda_{rs}^{i}\left(x_k^m-\mu_i^m\right)\right]\cdot\frac{\alpha_i' p_i(\mathbf{x}_k\mid\phi_i')}{p(\mathbf{x}_k\mid\Phi')} \qquad (5.131)\\
=\;&-\frac{1}{2}\sum_{k=1}^{N}\left(x_k^r-\mu_i^r\right)\left(x_k^s-\mu_i^s\right)\cdot\frac{\alpha_i' p_i(\mathbf{x}_k\mid\phi_i')}{p(\mathbf{x}_k\mid\Phi')}.
\end{aligned}$$

Finalmente,

$$\frac{\partial Q_2}{\partial\lambda_{rs}^{i}}=\frac{1}{2}\sum_{k=1}^{N}\frac{1}{\lambda_{rs}^{i}}\cdot\frac{\alpha_i' p_i(\mathbf{x}_k\mid\phi_i')}{p(\mathbf{x}_k\mid\Phi')}-\frac{1}{2}\sum_{k=1}^{N}\left(x_k^r-\mu_i^r\right)\left(x_k^s-\mu_i^s\right)\cdot\frac{\alpha_i' p_i(\mathbf{x}_k\mid\phi_i')}{p(\mathbf{x}_k\mid\Phi')}. \qquad (5.132)$$

Para obtermos o máximo, devemos ter $\frac{\partial Q_2}{\partial\lambda_{rs}^{i}}=0$. Logo,

$$\lambda_{rs}^{i}=\frac{\sum_{k=1}^{N}\frac{\alpha_i' p_i(\mathbf{x}_k\mid\phi_i')}{p(\mathbf{x}_k\mid\Phi')}}{\sum_{k=1}^{N}\left(x_k^r-\mu_i^r\right)\left(x_k^s-\mu_i^s\right)\cdot\frac{\alpha_i' p_i(\mathbf{x}_k\mid\phi_i')}{p(\mathbf{x}_k\mid\Phi')}}. \qquad (5.133)$$

Observemos que podemos obter a derivada $\frac{\partial Q_2}{\partial\lambda_{rs}^{i}}$ em termos dos elementos da matriz Λ_i, dados por θ_{rs}^{i}. Na Eq.(5.126), substituímos $|\Lambda_i|$ por $\frac{1}{\left|\Lambda_i^{-1}\right|}$, pois

$$|\Lambda_i|\left|\Lambda_i^{-1}\right|=|\mathbf{I}|. \qquad (5.134)$$

Temos então,

$$\begin{aligned}
&-\frac{1}{2}\sum_{k=1}^{N}\left[\frac{1}{|\Lambda_i|}\frac{\partial|\Lambda_i|}{\partial\lambda_{rs}^{i}}\right]\cdot\frac{\alpha_i' p_i(\mathbf{x}_k\mid\phi_i')}{p(\mathbf{x}_k\mid\Phi')}=-\frac{1}{2}\sum_{k=1}^{N}\left[\frac{1}{|\Lambda_i|}\frac{\partial}{\partial\lambda_{rs}^{i}}\frac{1}{\left|\Lambda_i^{-1}\right|}\right]\cdot\frac{\alpha_i' p_i(\mathbf{x}_k\mid\phi_i')}{p(\mathbf{x}_k\mid\Phi')}\\
=\;&-\frac{1}{2}\sum_{k=1}^{N}\left[-\frac{1}{|\Lambda_i|}\cdot\frac{\Theta_{rs}^{i}}{\left|\Lambda_i^{-1}\right|^2}\right]\cdot\frac{\alpha_i' p_i(\mathbf{x}_k\mid\phi_i')}{p(\mathbf{x}_k\mid\Phi')}=\frac{1}{2}\sum_{k=1}^{N}\theta_{sr}^{i}\cdot\frac{\alpha_i' p_i(\mathbf{x}_k\mid\phi_i')}{p(\mathbf{x}_k\mid\Phi')}. \qquad (5.135)
\end{aligned}$$

Comparando esta última expressão com a obtida na Eq. (5.130), concluímos que

$$\theta_{sr}^{i}=\frac{1}{\lambda_{rs}^{i}}. \qquad (5.136)$$

Se a matriz for simétrica,

$$\theta_{sr}^{i}=\frac{1}{\lambda_{sr}^{i}}. \qquad (5.137)$$

Podemos reescrever a Eq. (5.133) na seguinte forma

$$\frac{1}{\lambda_{rs}^{i}} = \theta_{sr}^{i} = \frac{\sum_{k=1}^{N} \left(x_k^r - \mu_i^r\right)\left(x_k^s - \mu_i^s\right) \cdot \frac{\alpha_i' p_i(\mathbf{x}_k|\phi_i')}{p(\mathbf{x}_k|\Phi')}}{\sum_{k=1}^{N} \frac{\alpha_i' p_i(\mathbf{x}_k|\phi_i')}{p(\mathbf{x}_k|\Phi')}}. \tag{5.138}$$

Na forma matricial, teremos :

$$\Lambda_i^{+} = \frac{\sum_{k=1}^{N} \left(\mathbf{x}_k - \mu_i^{+}\right)\left(\mathbf{x}_k - \mu_i^{+}\right)^T \cdot \frac{\alpha_i' p_i(\mathbf{x}_k|\phi_i')}{p(\mathbf{x}_k|\Phi')}}{\sum_{k=1}^{N} \frac{\alpha_i' p_i(\mathbf{x}_k|\phi_i')}{p(\mathbf{x}_k|\Phi')}}. \tag{5.139}$$

5.4.4 Introdução da Função de Penalidade

Sabemos que existem pontos $\Phi \in \Omega$ para os quais a função de log-verossimilhança de uma mistura de densidades Gaussianas assume valores infinitos, ou seja, torna-se degenerada. Estas singularidades da função de log-verossimilhança ocorrerão nas vizinhanças daqueles pontos $\Phi \in \Omega$ para os quais pelo menos uma das matrizes Λ_i é singular. Uma maneira de se evitar este problema é introduzir um limitante inferior às possíveis variâncias dos termos da mistura. Em geral, uma aproximação mais conveniente para solucionar este tipo de problema é adicionar uma função de penalidade à função de log-verossimilhança.

A idéia de penalizar a função de verossimilhança foi introduzida por Good e Gaskins [64, 65] para o problema de estimação não-paramétrica de densidades de probabilidade. Para estimação não-paramétrica, as singularidades da função de verossimilhança foram descritas por Good e Gaskins como a "catástrofe de Dirac". Eles sugeriram adicionar uma função de penalidade da função de log-verossimilhança que garanta a suavidade da densidade em questão. A maximização desta função de log-verossimilhança penalizada é sobre todo o espaço de parâmetros original Ω.

Para o caso multivariável, a função de penalidade sugerida por eles foi [65] :

$$\varphi\left[p(\mathbf{x})\right] = 4\nu \int |\nabla\gamma|^2 \, d\mathbf{x}, \tag{5.140}$$

onde ν é uma constante positiva e $\gamma = (f)^{1/2}$.

Em termos de $p(\mathbf{x})$, a Eq. (5.140) torna-se

$$\begin{aligned} \varphi\left[p(\mathbf{x})\right] &= 4\nu \int \left|\nabla\left[p(\mathbf{x})\right]^{1/2}\right|^2 d\mathbf{x} = 4\nu \int \left|\frac{1}{2}\frac{1}{p(\mathbf{x})^{1/2}}\nabla\left[p(\mathbf{x})\right]\right|^2 d\mathbf{x} \\ &= \nu \int \frac{1}{p(\mathbf{x})} \left|\nabla\left[p(\mathbf{x})\right]\right|^2 d\mathbf{x} = \nu \int \frac{1}{p(\mathbf{x})} \left[\sum_j \left(\frac{\partial p(\mathbf{x})}{\partial x_j}\right)^2\right] d\mathbf{x}. \end{aligned} \tag{5.141}$$

Se $p(\mathbf{x})$ for uma mistura de densidades Gaussianas, a integral (5.141) torna-se muito difícil de ser calculada [65]. Por simplicidade, e sem detrimento à idéia inicial de penalidade, utilizaremos a função

$$\varphi^* [p(\mathbf{x})] = \sum_{i=1}^{M} \varphi [p_i(\mathbf{x})] , \tag{5.142}$$

onde $p_i(\mathbf{x})$ é a i-ésima densidade componente de $p(\mathbf{x})$ e $\varphi [p_i(\mathbf{x})]$ é dada pela Eq. (5.140).

Vamos supor então que $p_i(\mathbf{x})$ é uma densidade Gaussiana de n dimensões, dada pela Eq. (5.82). Derivando em relação às componentes x_j, teremos

$$\begin{aligned}
\frac{\partial p_i(\mathbf{x})}{\partial x_j} &= \frac{\partial}{\partial x_j} \left\{ \frac{1}{(2\pi)^{n/2} |\Lambda_i|^{1/2}} \cdot \exp \left[-\frac{1}{2} \sum_{r=1}^{n} \sum_{s=1}^{n} (x_k^r - \mu_i^r) \lambda_{rs}^i (x_k^s - \mu_i^s) \right] \right\} \qquad (5.143) \\
&= \frac{1}{(2\pi)^{n/2} |\Lambda_i|^{1/2}} \cdot \frac{\partial}{\partial x_j} \exp \left[-\frac{1}{2} \sum_{r=1}^{n} \sum_{s=1}^{n} (x_k^r - \mu_i^r) \lambda_{rs}^i (x_k^s - \mu_i^s) \right] \\
&= p_i(\mathbf{x}) \cdot \frac{\partial}{\partial x_j} \left[-\frac{1}{2} \sum_{r=1}^{n} \sum_{s=1}^{n} (x_k^r - \mu_i^r) \lambda_{rs}^i (x_k^s - \mu_i^s) \right] \\
&= -\frac{1}{2} \cdot p_i(\mathbf{x}) \cdot \sum_{r=1}^{n} \sum_{s=1}^{n} \lambda_{rs}^i \left[\delta_{js} (x_k^r - \mu_i^r) + \delta_{jr} (x_k^s - \mu_i^s) \right] \\
&= -p_i(\mathbf{x}) \cdot \sum_{r=1}^{n} \lambda_{jr}^i (x_k^r - \mu_i^r) .
\end{aligned}$$

Portanto,

$$\begin{aligned}
\varphi [p_i(\mathbf{x})] &= \nu \int \frac{1}{p_i(\mathbf{x})} \left[\sum_j \left(\frac{\partial p_i(\mathbf{x})}{\partial x_j} \right)^2 \right] d\mathbf{x} = \qquad (5.144) \\
&= \nu \sum_j \int \frac{1}{p_i(\mathbf{x})} \cdot p_i(\mathbf{x})^2 \cdot \left[\sum_{r=1}^{n} \lambda_{jr}^i (x_k^r - \mu_i^r) \right]^2 d\mathbf{x} \\
&= \nu \sum_j \int p_i(\mathbf{x}) \cdot \left[\sum_{r=1}^{n} \sum_{s=1}^{n} \lambda_{jr}^i \lambda_{js}^i (x_k^r - \mu_i^r)(x_k^s - \mu_i^s) \right] d\mathbf{x} \\
&= \nu \sum_j \sum_{r=1}^{n} \sum_{s=1}^{n} \lambda_{jr}^i \lambda_{js}^i \int (x_k^r - \mu_i^r)(x_k^s - \mu_i^s) \cdot p_i(\mathbf{x}) \, d\mathbf{x} \\
&= \nu \sum_j \sum_{r=1}^{n} \sum_{s=1}^{n} \lambda_{jr}^i \lambda_{js}^i \theta_{rs}^i .
\end{aligned}$$

Como $\Lambda_i \Lambda_i^{-1} = \mathbf{I}$, e considerando que Λ_i seja simétrica,

$$\begin{aligned}
\varphi [p_i(\mathbf{x})] &= \nu \sum_j \sum_{r=1}^{n} \lambda_{jr}^i \sum_{s=1}^{n} \lambda_{js}^i \theta_{rs}^i = \nu \sum_j \sum_{r=1}^{n} \lambda_{jr}^i \sum_{s=1}^{n} \lambda_{sj}^i \theta_{rs}^i \qquad (5.145) \\
&= \nu \sum_j \sum_{r=1}^{n} \lambda_{jr}^i \delta_{jr} = \nu \sum_j \lambda_{jj}^i .
\end{aligned}$$

Logo,

$$\varphi\left[p_i(\mathbf{x})\right] = \nu \cdot Tr\left[\Lambda_i^{-1}\right], \tag{5.146}$$

e finalmente,

$$\varphi^*\left[p(\mathbf{x})\right] = \nu \sum_{i=1}^{M} Tr\left[\Lambda_i^{-1}\right]. \tag{5.147}$$

5.4.5 Algoritmo EM com Função de Penalidade para Mistura de Gaussianas

Nesta seção, introduziremos uma pequena modificação no algoritmo EM original com o objetivo de introduzir uma função de penalidade para a função de log-verossimilhança no caso de mistura de densidades Gaussianas. Utilizaremos a função de penalidade apresentada na seção anterior. Como definido na seção 5.4, sejam $f(\mathbf{y} \mid \Phi)$, $g(\mathbf{x} \mid \Phi)$ e $k(\mathbf{y} \mid \mathbf{x}, \Phi)$ relacionados por

$$k(\mathbf{y} \mid \mathbf{x}, \Phi) = \frac{f(\mathbf{y} \mid \Phi)}{g(\mathbf{x} \mid \Phi)}. \tag{5.148}$$

Logo,

$$\ln f(\mathbf{y} \mid \Phi) = \ln k(\mathbf{y} \mid \mathbf{x}, \Phi) + \ln g(\mathbf{x} \mid \Phi) \tag{5.149}$$

ou

$$\ln g(\mathbf{x} \mid \Phi) = \ln f(\mathbf{y} \mid \Phi) - \ln k(\mathbf{y} \mid \mathbf{x}, \Phi). \tag{5.150}$$

Se subtrairmos uma função não-negativa $\varphi\left[g(\mathbf{x} \mid \Phi)\right]$ em ambos os membros da Eq. (5.150) anterior, teremos

$$\ln g(\mathbf{x} \mid \Phi) - \varphi\left[g(\mathbf{x} \mid \Phi)\right] = \ln f(\mathbf{y} \mid \Phi) - \varphi\left[g(\mathbf{x} \mid \Phi)\right] - \ln k(\mathbf{y} \mid \mathbf{x}, \Phi). \tag{5.151}$$

Aplicando o operador de esperança condicional $E\left[\cdot \mid \mathbf{x}, \Phi'\right]$, teremos

$$\begin{aligned} & E\left[\ln g(\mathbf{x} \mid \Phi) - \varphi\left[g(\mathbf{x} \mid \Phi)\right] \mid \mathbf{x}, \Phi'\right] = \\ = \; & E\left[\ln f(\mathbf{y} \mid \Phi) - \varphi\left[g(\mathbf{x} \mid \Phi)\right] \mid \mathbf{x}, \Phi'\right] - E\left[\ln k(\mathbf{y} \mid \mathbf{x}, \Phi) \mid \mathbf{x}, \Phi'\right]. \end{aligned} \tag{5.152}$$

Ou seja,

$$L^P(\Phi) = Q^P(\Phi \mid \Phi') - H(\Phi \mid \Phi'), \tag{5.153}$$

onde

$$L^P(\Phi) = \ln g(\mathbf{x} \mid \Phi) - \varphi[g(\mathbf{x} \mid \Phi)] \tag{5.154}$$

e

$$Q^P(\Phi \mid \Phi') = E[\ln f(\mathbf{y} \mid \Phi) \mid \mathbf{x}, \Phi'] - \varphi[g(\mathbf{x} \mid \Phi)]. \tag{5.155}$$

Percebemos que neste caso o valor máximo da função de log-verossimilhança não será mais $L(\Phi^*)$, e sim $L'(\Phi^*) \leq L(\Phi^*)$. Além disso, podemos esperar que as propriedades de convergência do algoritmo EM permaneçam inalteradas, uma vez que a penalidade é aplicada em ambos os membros da Eq. (5.150).

Para o problema específico de mistura de densidades Gaussianas, se tomarmos a função $\varphi[g(\mathbf{x} \mid \Phi)]$ como sendo a Eq. (5.147), nós teremos, a partir da Eq. (5.99),

$$Q^P(\Phi \mid \Phi') = \sum_{k=1}^{N}\sum_{i=1}^{M} \ln \alpha_i p_i(\mathbf{x}_k \mid \phi_i) \frac{\alpha_i' p_i(\mathbf{x}_k \mid \phi_i')}{p(\mathbf{x}_k \mid \Phi')} - \nu \sum_{i=1}^{M} Tr\left[\Lambda_i^{-1}\right]. \tag{5.156}$$

Observando a expressão acima e comparando com os resultados obtidos para o caso em que a função de log-verossimilhança não é penalizada, para maximizarmos a função $Q^P(\Phi \mid \Phi')$ resta-nos obter apenas a expressão para os termos da matriz Λ_i^{-1}, permanecendo inalteradas as expressões para o cálculo dos α_i e μ_i.

Calculando $\frac{\partial Q^P}{\partial \Lambda_i^{-1}}$, obtemos

$$\begin{aligned} \frac{\partial Q^P}{\partial \lambda_{rs}^i} = \frac{\partial}{\partial \lambda_{rs}^i} \Bigg[& -\frac{1}{2}\sum_{k=1}^{N} \ln|\Lambda_i| \cdot \frac{\alpha_i' p_i(\mathbf{x}_k \mid \phi_i')}{p(\mathbf{x}_k \mid \Phi')} - \\ & \frac{1}{2}\sum_{k=1}^{N} (\mathbf{x}_k - \boldsymbol{\mu}_i)^T \Lambda_i^{-1} (\mathbf{x}_k - \boldsymbol{\mu}_i) \cdot \frac{\alpha_i' p_i(\mathbf{x}_k \mid \phi_i')}{p(\mathbf{x}_k \mid \Phi')} - \nu \sum_{i=1}^{M} Tr\left[\Lambda_i^{-1}\right] + const. \Bigg]. \end{aligned} \tag{5.157}$$

As derivadas do primeiro e do segundo somatório já foram obtidas anteriormente, restando calcular apenas a derivada do termo de penalidade, que é dada por

$$\frac{\partial}{\partial \lambda_{rs}^i}\left[\nu \sum_{i=1}^{M}\sum_{j=1}^{n} \lambda_{jj}^i\right] = \nu \delta_{rs}^i \tag{5.158}$$

Tomando o resultado da Eq. (5.132), teremos

$$\frac{\partial Q^P}{\partial \lambda_{rs}^i} = \frac{1}{2}\sum_{k=1}^{N} \frac{1}{\lambda_{rs}^i} \cdot \frac{\alpha_i' p_i(\mathbf{x}_k \mid \phi_i')}{p(\mathbf{x}_k \mid \Phi')} - \frac{1}{2}\sum_{k=1}^{N} (x_k^r - \mu_i^r)(x_k^s - \mu_i^s) \cdot \frac{\alpha_i' p_i(\mathbf{x}_k \mid \phi_i')}{p(\mathbf{x}_k \mid \Phi')} - \nu \delta_{rs}^i. \tag{5.159}$$

Fazendo $\frac{\partial Q^P}{\partial \lambda_{rs}^i} = 0$, obtemos

$$\frac{1}{\lambda_{rs}^{i}} = \theta_{sr}^{i} = \frac{\sum_{k=1}^{N} \left(x_k^r - \mu_i^r\right)\left(x_k^s - \mu_i^s\right) \cdot \frac{\alpha_i' p_i(\mathbf{x}_k|\phi_i')}{p(\mathbf{x}_k|\Phi')} + 2\nu\delta_{rs}^{i}}{\sum_{k=1}^{N} \frac{\alpha_i' p_i(\mathbf{x}_k|\phi_i')}{p(\mathbf{x}_k|\Phi')}}. \tag{5.160}$$

Na forma matricial,

$$\Lambda_i = \frac{\sum_{k=1}^{N} \left(\mathbf{x}_k - \boldsymbol{\mu}_i\right)\left(\mathbf{x}_k - \boldsymbol{\mu}_i\right)^T \cdot \frac{\alpha_i' p_i(\mathbf{x}_k|\phi_i')}{p(\mathbf{x}_k|\Phi')}}{\sum_{k=1}^{N} \frac{\alpha_i' p_i(\mathbf{x}_k|\phi_i')}{p(\mathbf{x}_k|\Phi')}} + \frac{2\nu}{\sum_{k=1}^{N} \frac{\alpha_i' p_i(\mathbf{x}_k|\phi_i')}{p(\mathbf{x}_k|\Phi')}}\mathbf{I} \tag{5.161}$$

Portanto, o *algoritmo EM para mistura de densidades Gaussianas com função de penalidade* será dado por

$$\alpha_i^+ = \frac{1}{N}\sum_{k=1}^{N} \frac{\alpha_i' p_i(\mathbf{x}_k \mid \phi_i')}{p(\mathbf{x}_k \mid \Phi')}, \tag{5.162}$$

$$\boldsymbol{\mu}_i^+ = \frac{\sum_{k=1}^{N} \mathbf{x}_k \cdot \frac{\alpha_i' p_i(\mathbf{x}_k|\phi_i')}{p(\mathbf{x}_k|\Phi')}}{\sum_{k=1}^{N} \frac{\alpha_i' p_i(\mathbf{x}_k|\phi_i')}{p(\mathbf{x}_k|\Phi')}} \tag{5.163}$$

e

$$\Lambda_i^+ = \frac{\sum_{k=1}^{N} \left[\left(\mathbf{x}_k - \boldsymbol{\mu}_i^+\right)\left(\mathbf{x}_k - \boldsymbol{\mu}_i^+\right)^T \cdot \frac{\alpha_i' p_i(\mathbf{x}_k|\phi_i')}{p(\mathbf{x}_k|\Phi')}\right] + 2\nu\mathbf{I}}{\sum_{k=1}^{N} \frac{\alpha_i' p_i(\mathbf{x}_k|\phi_i')}{p(\mathbf{x}_k|\Phi')}}. \tag{5.164}$$

onde o símbolo $^+$ significa a nova estimação do parâmetro em questão. Além disso, utilizamos a estimação mais atual $\boldsymbol{\mu}_i^+$ da média para a estimação Λ_i^+ da matriz de covariância.

No caso particular em que $\Lambda_i = \sigma_i^2\mathbf{I}$, as densidades Gaussianas terão a forma simplificada

$$p_i\left(\mathbf{x} \mid \phi_i\right) = \frac{1}{\left(2\pi\sigma_i^2\right)^{n/2}} \exp\left\{-\frac{\left\|\mathbf{x} - \mu_i\right\|^2}{2\sigma_i^2}\right\}. \tag{5.165}$$

Neste caso, a complexidade do problema reduz-se drasticamente, pois são excluídas as operações de determinante e inversão de matrizes, além de se reduzir o número de parâmetros de $n\left(n+3\right)/2$ (devido à simetria da matriz de covariância) para apenas $\left(n+1\right)$ parâmetros, onde n é a dimensão do espaço em questão.

Repetindo-se um desenvolvimento similar ao anterior, chegamos à seguinte expressão para a atualização da variância σ_i^2 da i-ésima Gaussiana, com a introdução da função de penalidade

$$\sigma_i^{2+} = \frac{\left(\frac{1}{d}\right)\sum_{k=1}^{N} \left\|\mathbf{x} - \mu_i\right\|^2 \frac{\alpha_i' p_i(\mathbf{x}_k|\phi_i')}{p(\mathbf{x}_k|\Phi')} + 2\nu}{\sum_{k=1}^{N} \frac{\alpha_i' p_i(\mathbf{x}_k|\phi_i')}{p(\mathbf{x}_k|\Phi')}}. \tag{5.166}$$

5.4.6 Algoritmo EM Adaptativo para Estimação de Parâmetros de Misturas de Densidades Gaussianas

Do que foi exposto anteriormente, podemos perceber que o algoritmo EM é uma excelente alternativa para a estimação dos parâmetros de uma mistura de densidades de probabilidade. Em particular, mostramos o quão simples se torna quando as densidades em questão são Gaussianas. No entanto, na forma em que foi concebido, este algoritmo exige a utilização de **todo** o conjunto de amostras disponíveis para a realização de uma nova estimação.

Se considerarmos uma situação em que o número de amostras $\mathbf{x}_k$ possa variar, onde a cada instante de tempo uma nova amostra $\mathbf{x}_k$ é adicionada ao conjunto e que as densidades de probabilidade se modificam ao longo do tempo (um ambiente não-estacionário), as estimações deverão ser realizadas dentro de um conjunto cada vez maior de amostras, exigindo um grande volume de armazenamento, o que torna o algoritmo impraticável para qualquer implementação em tempo real.

Pensando nisto, tentamos obter, a partir do algoritmo EM, um algoritmo que pudesse ser implementado em tempo real e que realizasse as estimações dos parâmetros de uma mistura de Gaussianas sem a exigência de armazenamento de todas as amostras. Nesse contexto, as estimações que antes eram realizadas em cada passo do algoritmo EM, agora serão realizadas *instantaneamente*, com a chegada de uma nova amostra $\mathbf{x}_k$.

Embora este algoritmo tenha sido concebido de forma independente, uma abordagem bastante similar foi proposta por Tråvén [66] para o problema de classificação de padrões a partir de estimações semi-paramétricas de funções de densidade de probabilidade. Tråvén partiu diretamente do problema de maximização da função de log-verossimilhança de uma mistura de densidades Gaussianas e utilizou as expressões analíticas da solução de máxima verossimilhança para obtenção de um algoritmo do tipo gradiente estocástico.

Basicamente, o algoritmo aqui concebido diferencia-se do proposto por Tråvén pelo fato de ter sido elaborado no contexto do algoritmo EM e por tratar o problema das singularidades através de uma função de penalidade. Além disso, utiliza um *fator de esquecimento,* ou *fator de ponderação,* no intuito de garantir que dados no passado distante sejam "esquecidos", a fim de garantir a possibilidade de "rastreamento" das variações estatísticas dos dados observados quando em ambientes não-estacionários. Em seu trabalho, Tråvén propõe o truncamento do número de amostras consideradas no passado e sugere a apresentação de amostras distintas para evitar a ocorrência de singularidades [66].

Como principais vantagens vislumbradas para o algoritmo aqui proposto, podemos citar :

- Processamento em tempo real;
- Ideal para ambientes não-estacionários;
- Economia de armazenamento;
- Facilidade de implementação.

Como estamos trabalhando agora em um contexto "temporal", onde cada estimação é realizada com a chegada de uma nova amostra $\mathbf{x}_k$, faremos uma mudança da notação utilizada. Assim, as estimações em cada passo do algoritmo EM tornar-se-ão estimações instantâneas, ou seja, teremos a seguinte correspondência :

$$\begin{array}{rcl} \mathbf{x}_k & \Rightarrow & \mathbf{x}(n) \\ \alpha_i' & \Rightarrow & \alpha_i(n-1) \\ \mu_i' & \Rightarrow & \mu_i(n-1) \\ \Lambda_i' & \Rightarrow & \Lambda_i[n-1] \\ \phi_i' & \Rightarrow & \phi_i[n-1] \\ \Phi' & \Rightarrow & \Phi[n-1] \end{array}$$

Para a obtenção deste algoritmo, partimos das Eqs. (5.162), (5.163) e (5.164) e escrevemos uma forma recursiva para o valor do parâmetro no instante n, a partir do seu valor no instante $n-1$. Sendo assim, a probabilidade *a posteriori* de que a observação $\mathbf{x}(n)$, no instante n, tenha sido gerada na i-ésima Gaussiana, será dada por

$$P(i \mid \mathbf{x}(n)) = \frac{\alpha_i(n-1) \cdot p_i(\mathbf{x}(n) \mid \phi_i(n-1))}{p(\mathbf{x}(n) \mid \Phi(n-1))}, \tag{5.167}$$

onde

$$\begin{aligned} p_i(\mathbf{x}(n) \mid \phi_i(n-1)) &= \frac{1}{(2\pi)^{n/2} |\Lambda_i(n-1)|^{1/2}} \cdot \\ &\exp\left\{-\frac{1}{2}[\mathbf{x}(n) - \mu_i(n-1)]^T \Lambda_i^{-1}(n-1)[\mathbf{x}(n) - \mu_i(n-1)]\right\} \end{aligned} \tag{5.168}$$

e

$$p(\mathbf{x}(n) \mid \Phi(n-1)) = \sum_{i=1}^{M} \alpha_i(n-1) \cdot p_i(\mathbf{x}(n) \mid \phi_i(n-1)). \tag{5.169}$$

Nas Eqs. (5.162), (5.163) e (5.164) introduziremos um *fator de ponderação* ou *fator de esquecimento*, com o objetivo de permitir que os dados no passado distante sejam esquecidos e que as estimações realizadas acompanhem as variações estatísticas dos dados observados. O fator de esquecimento utilizado será da forma :

$$\beta(n,k) = \beta^{n-k}, \qquad k = 1, 2, \ldots, n \tag{5.170}$$

onde $0 < \beta < 1$.

No caso das probabilidades α_i das densidades componentes, teremos

$$\begin{aligned} \alpha_i(n) &= \frac{\sum_{k=1}^{n} \beta^{n-k} \frac{\alpha_i(k-1) \cdot p_i(\mathbf{x}(k)|\phi_i(k-1))}{p(\mathbf{x}(k)|\Phi(k-1))}}{\sum_{k=1}^{n} \beta^{n-k}} = \\ &= \frac{\sum_{k=1}^{n-1} \beta^{n-k} \frac{\alpha_i(k-1) \cdot p_i(\mathbf{x}(k)|\phi_i(k-1))}{p(\mathbf{x}(k)|\Phi(k-1))} + \frac{\alpha_i(n-1) \cdot p_i(\mathbf{x}(n)|\phi_i(n-1))}{p(\mathbf{x}(n)|\Phi(n-1))}}{\sum_{k=1}^{n} \beta^{n-k}} \\ &= \frac{\beta\left[\sum_{k=1}^{n-1} \beta^{n-k-1} \frac{\alpha_i(k-1) \cdot p_i(\mathbf{x}(k)|\phi_i(k-1))}{p(\mathbf{x}(k)|\Phi(k-1))}\right] + \frac{\alpha_i(n-1) \cdot p_i(\mathbf{x}(n)|\phi_i(n-1))}{p(\mathbf{x}(n)|\Phi(n-1))}}{\sum_{k=1}^{n} \beta^{n-k}}. \end{aligned} \tag{5.171}$$

Mas,

$$\sum_{k=1}^{n} \beta^{n-k} = \frac{1-\beta^n}{1-\beta}, \tag{5.172}$$

pois $0 < \beta < 1$. Logo,

$$\alpha_i(n) = \frac{\beta\alpha_i(n-1)\left(\frac{1-\beta^{n-1}}{1-\beta}\right) + \frac{\alpha_i(n-1)\cdot p_i(\mathbf{x}(n)|\phi_i(n-1))}{p(\mathbf{x}(n)|\Phi(n-1))}}{\left(\frac{1-\beta^n}{1-\beta}\right)}. \tag{5.173}$$

Temos então

$$\alpha_i(n) = \left(\frac{1-\beta^{n-1}}{1-\beta^n}\right)\beta\alpha_i(n) + \left(\frac{1-\beta}{1-\beta^n}\right)\frac{\alpha_i(n-1)\,p_i(\mathbf{x}(n) \mid \phi_i(n-1))}{p(\mathbf{x}(n) \mid \Phi(n-1))} \tag{5.174}$$

Para $n \to \infty$ (regime estacionário),

$$\alpha_i(n) = \beta\alpha_i(n-1) + (1-\beta)\frac{\alpha_i(n-1)\,p_i(\mathbf{x}(n) \mid \phi_i(n-1))}{p(\mathbf{x}(n) \mid \Phi(n-1))}, \tag{5.175}$$

ou

$$\alpha_i(n) = \beta\alpha_i(n-1) + (1-\beta)\,P(i \mid \mathbf{x}(n), \phi_i(n-1)). \tag{5.176}$$

Para as médias das densidades, teremos

$$\boldsymbol{\mu}_i(n) = \frac{\sum_{k=1}^{n}\beta^{n-k}\mathbf{x}(k)\cdot\left(\frac{\alpha_i(k-1)\cdot p_i(\mathbf{x}(k)|\phi_i(k-1))}{p(\mathbf{x}(k)|\Phi(k-1))}\right)}{\sum_{k=1}^{n}\beta^{n-k}\left(\frac{\alpha_i(k-1)\cdot p_i(\mathbf{x}(k)|\phi_i(k-1))}{p(\mathbf{x}(k)|\Phi(k-1))}\right)}. \tag{5.177}$$

Calculemos antes o denominador

$$d_i(n) = \sum_{k=1}^{n}\beta^{n-k}\frac{\alpha_i(k-1)\cdot p_i(\mathbf{x}(k) \mid \phi_i(k-1))}{p(\mathbf{x}(k) \mid \Phi(k-1))} \tag{5.178}$$

$$\begin{aligned} &= \sum_{k=1}^{n-1}\beta^{n-k}\frac{\alpha_i(k-1)\cdot p_i(\mathbf{x}(k) \mid \phi_i(k-1))}{p(\mathbf{x}(k) \mid \Phi(k-1))} + \frac{\alpha_i(n-1)\cdot p_i(\mathbf{x}(n) \mid \phi_i(n-1))}{p(\mathbf{x}(n) \mid \Phi(n-1))} \\ &= \beta\left[\sum_{k=1}^{n-1}\beta^{n-k-1}\left(\frac{\alpha_i(k-1)\cdot p_i(\mathbf{x}(k) \mid \phi_i(k-1))}{p(\mathbf{x}(k) \mid \Phi(k-1))}\right)\right] \\ &\quad + \frac{\alpha_i(n-1)\cdot p_i(\mathbf{x}(n) \mid \phi_i(n-1))}{p(\mathbf{x}(n) \mid \Phi(n-1))} \end{aligned} \tag{5.179}$$

$$= \beta d_i(n-1) + \frac{\alpha_i(n-1)\cdot p_i(\mathbf{x}(n) \mid \phi_i(n-1))}{p(\mathbf{x}(n) \mid \Phi(n-1))}.$$

Reescrevendo a expressão da média,

$$\begin{aligned}\boldsymbol{\mu}_i(n) &= \frac{\sum_{k=1}^{n}\beta^{n-k}\mathbf{x}(k)\left(\frac{\alpha_i(k-1)\cdot p_i(\mathbf{x}(k)|\phi_i(k-1))}{p(\mathbf{x}(k)|\Phi(k-1))}\right)}{d_i(n)} \\ &= \frac{\beta\left[\sum_{k=1}^{n-1}\beta^{n-k-1}\mathbf{x}(k)\left(\frac{\alpha_i(k-1)\cdot p_i(\mathbf{x}(k)|\phi_i(k-1))}{p(\mathbf{x}(k)|\Phi(k-1))}\right)\right]+\mathbf{x}(n)\left(\frac{\alpha_i(n-1)\cdot p_i(\mathbf{x}(n)|\phi_i(n-1))}{p(\mathbf{x}(n)|\Phi(n-1))}\right)}{d_i(n)} \\ &= \frac{\beta\boldsymbol{\mu}_i(n-1)\,d_i(n-1)}{d_i(n)}+\mathbf{x}(n)\left(\frac{\alpha_i(n-1)\cdot p_i(\mathbf{x}(n)\mid\phi_i(n-1))}{p(\mathbf{x}(n)\mid\Phi(n-1))}\right).\end{aligned} \tag{5.180}$$

Mas,

$$\frac{\beta d_i(n-1)}{d_i(n)} = 1-\frac{1}{d_i(n)}\left(\frac{\alpha_i(n-1)\cdot p_i(\mathbf{x}(n)\mid\phi_i(n-1))}{p(\mathbf{x}(n)\mid\Phi(n-1))}\right) \tag{5.181}$$

Substituindo na Eq. (5.180),

$$\boldsymbol{\mu}_i(n) = \boldsymbol{\mu}_i(n-1)+\frac{1}{d_i(n)}\left[\mathbf{x}(n)-\boldsymbol{\mu}_i(n-1)\right]\left(\frac{\alpha_i(n-1)\cdot p_i(\mathbf{x}(n)\mid\phi_i(n-1))}{p(\mathbf{x}(n)\mid\Phi(n-1))}\right). \tag{5.182}$$

Trabalhando da mesma forma com a matriz Λ_i, teremos

$$\begin{aligned}\Lambda_i(n) &= \frac{\sum_{k=1}^{n}\beta^{n-k}\left[[\mathbf{x}(k)-\boldsymbol{\mu}_i(k)]\cdot[\mathbf{x}(k)-\boldsymbol{\mu}_i(k)]^T P(i\mid\mathbf{x}(k))\right]+2\nu\mathbf{I}}{\sum_{k=1}^{n}\beta^{n-k}P(i\mid\mathbf{x}(n))} \\ &= \frac{\beta\left\{\sum_{k=1}^{n-1}\beta^{n-k-1}\left[[\mathbf{x}(k)-\boldsymbol{\mu}_i(k)]\cdot[\mathbf{x}(k)-\boldsymbol{\mu}_i(k)]^T P(i\mid\mathbf{x}(k))\right]\right\}}{d_i(n)} \\ &\quad \frac{+\left[[\mathbf{x}(n)-\boldsymbol{\mu}_i(n)]\cdot[\mathbf{x}(n)-\boldsymbol{\mu}_i(n)]^T P(i\mid\mathbf{x}(n))\right]+2\nu\mathbf{I}}{d_i(n)} \\ &= \frac{\beta\left[\Lambda_i(n-1)\,d_i(n-1)-2\nu\mathbf{I}\right]+\left[[\mathbf{x}(n)-\boldsymbol{\mu}_i(n)]\cdot[\mathbf{x}(n)-\boldsymbol{\mu}_i(n)]^T P(i\mid\mathbf{x}(n))\right]}{d_i(n)} \\ &\quad +\frac{2\nu\mathbf{I}}{d_i(n)} \\ &= \frac{\beta\Lambda_i(n-1)\,d_i(n-1)}{d_i(n)}+\frac{\left[[\mathbf{x}(n)-\boldsymbol{\mu}_i(n)]\cdot[\mathbf{x}(n)-\boldsymbol{\mu}_i(n)]^T P(i\mid\mathbf{x}(n))\right]}{d_i(n)} \\ &\quad +\frac{2\nu(1-\beta)\mathbf{I}}{d_i(n)} \\ &= \Lambda_i(n-1)\left[1-\frac{P(i\mid\mathbf{x}(n))}{d_i(n)}\right]+\frac{\left[[\mathbf{x}(n)-\boldsymbol{\mu}_i(n)]\cdot[\mathbf{x}(n)-\boldsymbol{\mu}_i(n)]^T P(i\mid\mathbf{x}(n))\right]}{d_i(n)} \\ &\quad \frac{+2\nu(1-\beta)\mathbf{I}}{d_i(n)}.\end{aligned}$$

Portanto,

$$\begin{aligned}\Lambda_i(n) &= \Lambda_i(n-1) + \frac{P(i \mid \mathbf{x}(n))}{d_i(n)} \Big[[\mathbf{x}(n) - \boldsymbol{\mu}_i(n)] [\mathbf{x}(n) - \boldsymbol{\mu}_i(n)]^T \\ &\quad - \Lambda_i(n-1)] + \frac{2\nu(1-\beta)}{d_i(n)} \mathbf{I}.\end{aligned}$$

Se fizermos

$$\eta_i(n) = \frac{1}{d_i(n)} \left(\frac{\alpha_i(n-1) \cdot p_i(\mathbf{x}(n) \mid \phi_i(n-1))}{p(\mathbf{x}(n) \mid \Phi(n-1))} \right), \tag{5.183}$$

nosso algoritmo se resume então a

$$\alpha_i(n) = \beta \alpha_i(n-1) + (1-\beta) P(i \mid \mathbf{x}(n)) \tag{5.184}$$

$$\boldsymbol{\mu}_i(n) = \boldsymbol{\mu}_i(n-1) + \eta_i(n) [x(n) - \boldsymbol{\mu}_i(n-1)] \tag{5.185}$$

$$\begin{aligned}\Lambda_i(n) &= \Lambda_i(n-1) + \eta_i(n) \Big\{ [\mathbf{x}(n) - \boldsymbol{\mu}_i(n-1)] [\mathbf{x}(n) - \boldsymbol{\mu}_i(n-1)]^T \\ &\quad - \Lambda_i(n-1)\} + \frac{2\nu(1-\beta)}{d_i(n)} \mathbf{I}\end{aligned} \tag{5.186}$$

No caso mais simples de uma matriz de covariância do tipo $\Lambda_i(n) = \sigma_i^2(n)\,\mathbf{I}$, a expressão para as variâncias $\sigma_i^2(n)$ fica da forma :

$$\sigma_i^2(n) = \sigma_i^2(n-1) + \eta_i(n) \left[\frac{1}{d} \|\mathbf{x}(n) - \boldsymbol{\mu}_i(n)\|^2 - \sigma_i^2(n-1) \right] + \frac{2\nu(1-\beta)}{d_i(n)} \tag{5.187}$$

É interessante observar neste algoritmo algumas similaridades com os algoritmos de gradiente estocástico que se utilizam de um sinal de "erro" entre uma resposta "desejada" e a saída obtida para sua atualização. No nosso caso, o "sinal de erro" é formado pela diferença entre um termo que contém a amostra atual $\mathbf{x}(n)$ e a estimação passada do parâmetro. Sendo assim, podemos afirmar que este é um algoritmo do tipo *auto-organizado* ("self-organized") ou *não-supervisionado*, no sentido que os parâmetros do modelo ajustam-se de acordo com as próprias amostras a que são apresentados, sem a necessidade de um segundo conjunto de amostras consideradas como *desejadas*.

Outro ponto importante a considerar é o fato de termos uma forma explícita para a "taxa de aprendizado" $\eta_i(n)$ do algoritmo, diferentemente da maioria dos algoritmos do tipo gradiente estocástico, onde essas taxas são dadas de alguma forma empírica ou simplesmente fixadas, em algum valor apropriado ditado pela prática.

Como se trata de um algoritmo recursivo, existe o inconveniente da possibilidade de ocorrência de instabilidades e/ou problemas de convergência, o que não foi estudado teoricamente.

Capítulo 6

Resultados

6.1 Introdução

Este Capítulo tem como objetivo apresentar os resultados obtidos para a predição de tráfego de redes LAN Ethernet através dos preditores estudados no Capítulo 5. Para isto, são utilizadas séries temporais obtidas a partir dos arquivos OctExt.TL e pOct.TL apresentados no Capítulo 4.

Antes de introduzirmos os resultados propriamente ditos, faremos uma breve introdução ao problema da predição de tráfego no contexto de redes ATM, focalizando a importância da predição para o problema de *transbordamento de "buffers"* na rede ATM.

Além das predições de tráfego Ethernet, apresentamos no final do Capítulo uma aplicação do *algoritmo EM adaptativo,* proposto neste trabalho, na estimação de parâmetros de misturas de densidades Gaussianas.

6.2 O Problema da Predição

Como afirmado anteriormente, um dos principais componentes das redes ATM é o **comutador de células ATM** [67, 9, 15]. Em geral, este componente engloba um ou mais **multiplexadores estatísticos** normalmente posicionados em sua saída. Em um comutador ATM típico, n enlaces de entrada I_i $(i = 1, 2, ..., n)$ são comutados para q enlaces de saída O_j $(j = 1, 2, ..., q)$. Cada enlace I_i suporta o fluxo das células ATM que foram agregadas no nó anterior da rede. A comutação de células de I_i para O_j é realizada por uma tabela ou matriz de comutação que é freqüentemente atualizada pelo sistema de controle de admissão da rede.

Considerando-se um certo enlace de entrada I_i qualquer, definimos por V_i o fluxo de células do enlace I_i que é destinado a um enlace de saída O_j. V_i representa, então, a quantidade de células de I_i para O_j por unidade de tempo. Dessa forma, V_1, V_2, ... , V_n, representam os fluxos de entrada, conforme mostrado no multiplexador estatístico da Figura 6.1. Nessa figura, V_o representa a vazão máxima **agregada** suportada por um enlace O_j, isto é, a quantidade máxima de células por intervalo de tempo ΔT (admitiremos ser esta a nossa unidade de tempo). Para facilitar a notação, vamos denominar por *vazão* tanto os *fluxos* V_i de entrada quanto a *vazão* V_o em um dado enlace de saída. Note que V_s é a soma (variável) de todas as vazões de entrada, ao passo que V_o é a vazão constante de saída.

Denominamos por X_s o processo de contagem de células à entrada da memória elástica da Figura 6.1. Isto é, a partir de um instante $t = t_0$, $X_s(t)$ representa o número total de células a

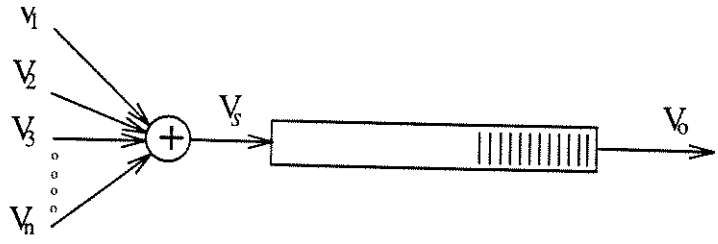

FIGURA 6.1: Diagrama esquemático de um multiplexador estatístico.

ser armazenado na memória no intervalo de tempo $t - t_0$. Neste mesmo intervalo, serão lidas ou retiradas da memória uma quantidade de células igual a $(t - t_0)V_o$.

Considerando-se que ΔT foi fixado, a vazão V_s é exatamente representada por

$$V_s(t) = \frac{X_s(t + \Delta T) - X_s(t)}{\Delta T}, \tag{6.1}$$

isto é, V_s é dada pelos incrementos normalizados de X_s.

Na Figura 6.2 é apresentada uma curva típica da variação da vazão total de entrada em função do tempo. Deste gráfico, podemos observar claramente a necessidade das memórias elásticas frente ao problema da eficiência de uso dos enlaces. Por exemplo, na hipótese de não utilização dessas memórias, haveria desperdício de recursos de transmissão (perda de faixa) nos instantes onde $V_s < V_o$, ao passo que quando $V_s > V_o$ haveria interrupção de transmissão de informação (perda de células). É justamente a capacidade das memórias elásticas de absorverem estas variações de vazão que tornam os multiplexadores estatísticos as *peças-chave* das redes com usuários de taxa de bit variável (VBR).

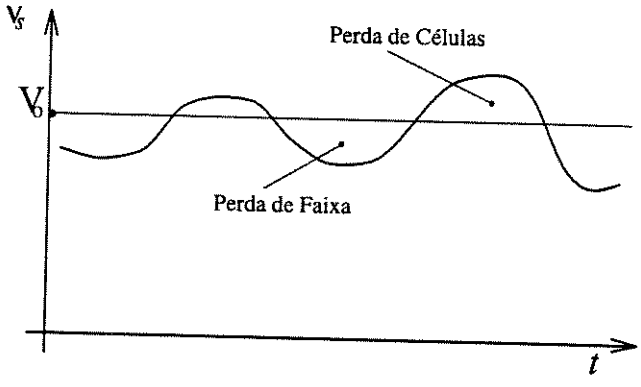

FIGURA 6.2: Curva típica da variação da vazão de entrada em função do tempo.

Até alguns anos atrás, admitia-se como hipótese realista que os processos estocásticos representativos das vazões (e conseqüentemente de V_s) eram processos de Poisson ou similares. Supunha-se que estes processos possuíam incrementos *independentes*. Neste caso, a multiplexação estatística estaria operando da forma mais favorável possível, pois eventuais picos de vazão ($V_s > V_o$) seriam rapidamente compensados pelos seus *vales* subseqüentes ($V_s < V_o$). Isto decorreria justamente da independência das amostras de V_s, que corresponderia a um processo estocástico *rápido*, à semelhança do ruído Gaussiano branco.

Um problema crucial no desempenho das Redes ATM é o *transbordamento de células* nas memórias elásticas. Quando isto ocorre, há perda de células com a conseqüente degradação

dos serviços de usuário. Pode-se mostrar que, sob a hipótese de *incrementos independentes*, a probabilidade de transbordamento (P_d) em um multiplexador com memória de tamanho L (medida em células, por exemplo), decresce *exponencialmente* com L, desde que $E\{Vs\}$, o valor médio de V_s, seja estritamente menor do que V_o. Além disso, o atraso médio de fila tende a um valor constante à medida que L tende a infinito. Esta hipótese garantiria, na prática, probabilidades de transbordamento desprezíveis com o uso de memórias relativamente pequenas.

Infelizmente, constatou-se recentemente que os principais sinais que farão uso da rede ATM possuem *incrementos dependentes*, consistindo em processos estocásticos lentos. Curiosamente, verificou-se também que esses sinais possuem fortes características *auto-similares* (*fractais*) .

Deve-se enfatizar que, quando o processo V_s é lento, o transbordamento da memória pode tornar-se muito freqüente, principalmente quando se deseja operar com $E\{Vs\}$ próximo de V_o (alta eficiência). Já foi demonstrado que, para processos *auto-similares*, a probabilidade de transbordamento decai apenas *algebricamente* com L. Além disso, o atraso médio de fila cresce *indefinidamente* com L [10].

A fim de analisarmos a aplicação da predição no problema de transbordamento de células nos multiplexadores estatísticos, devemos analisar, de antemão, a sua dinâmica de ocupação. Como definido anteriormente, seja V_s a vazão (ou fluxo) de entrada no multiplexador estatístico, V_o a vazão máxima suportada na saída do mesmo, $L_{\max}$ a ocupação máxima permitida no multiplexador e L_1 a sua ocupação no instante t_1. No intervalo de tempo $t - t_1$ ($t > t_1$), chega ao multiplexador um número de células igual a

$$\Delta X_s(t) = \int_{t_1}^{t} V_s(t')\, dt'. \tag{6.2}$$

Neste mesmo intervalo de tempo, é retirada da memória uma quantidade de células igual a

$$S(t) = (t - t_1)V_o. \tag{6.3}$$

Portanto, no instante $t > t_1$ a ocupação do multiplexador será dada por [67, 68]

$$L(t) = L_1 + \int_{t_1}^{t} V_s(t')\, dt' - (t - t_1)V_o \tag{6.4}$$

Haverá transbordamento de células se $L(t) > L_{\max}$. Da Eq. (6.4), notamos que o único termo do segundo membro não conhecido previamente é a quantidade de células $\Delta X_s(t) = \int_{t_1}^{t} V_s(t')\, dt'$ a ser recebida na entrada do multiplexador no intervalo de tempo $(t - t_1)$.

Uma maneira de se tentar evitar o transbordamento de células nos multiplexadores estatísticos é realizando-se uma predição desta quantidade. Portanto, se for possível a predição, o controle de congestionamento da rede pode tomar as devidas precauções para evitar o transbordamento de células. Não parece haver ainda um consenso de como isto deve ser feito, mas presume-se que um sistema de controle reativo possa efetuar esta tarefa de forma eficiente. Obviamente, este sinal de predição pode ser utilizado também para controle rápido de admissão, policiamento de usuário, entre outros.

Consideremos então um intervalo de tempo fixo, o qual chamaremos de intervalo básico ΔT. Para este intervalo, utilizamos o processo de incrementos de X_s, dado por $\{X_s(t) - X_s(t - \Delta T)\}$. Se $t = n\Delta T$, podemos eliminar ΔT das expressões (uma vez que ele está subentendido) e escrevê-las na forma discreta $\{X_s(n) - X_s(n-1)\}$, representado-as por $\{x(n)\}$, ou seja, $x(n) = X_s(n) - X_s(n-1)$. Estaremos interessados na predição do processo $\{x(n)\}$ a partir de um conjunto finito de amostras de tempo passado deste mesmo processo. Ou seja, desejamos estimar a amostra $x(n+1)$, no instante n, por

$$\widehat{x}(n+1) = F[x(n), x(n-1), \cdots, x(n-M)] \tag{6.5}$$

onde $F[\cdot]$ é alguma função das amostras passadas $x(n), x(n-1), \cdots, x(n-M)$.e M, o número de amostras passadas.

Uma importante interpretação pode ser dada à função $F[\cdot]$ quando otimizamos os nossos preditores segundo o critério de minimização do erro quadrático médio. É possível se mostrar [63] que, neste caso, a saída do preditor será uma estimação da *esperança condicional* do valor da amostra no instante $n+1$, dadas as amostras passadas $x(n), x(n-1), \cdots, x(n-M)$, ou seja,

$$\widehat{x}(n+1) = E[x(n+1) \mid x(n), x(n-1), \cdots, x(n-M)]. \tag{6.6}$$

6.3 As Séries Temporais

Como nosso objetivo é avaliar o desempenho dos preditores na predição de tráfego auto-similar *real*, extraímos as nossas séries temporais a partir dos arquivos OctExt.TL e pOct.TL. Estes arquivos, como já descritos anteriormente, constituem-se de registros de medições de alta resolução dos instantes de chegada dos pacotes Ethernet e de seus respectivos tamanhos em bytes.

Na predição, é interessante que trabalhemos com uma unidade de tempo bem definida, de forma que o "passo" da predição esteja bem caracterizado. Portanto, nas simulações que se seguem, utilizamos séries temporais obtidas a partir dos arquivos originais de acordo com uma escala de tempo bem definida.

Para o arquivo OctExt.TL a escala de tempo escolhida foi de 1 (um) minuto e resultou em uma série de 2046 pontos, o que corresponde a aproximadamente 34h10min de tráfego. No caso do arquivo pOct.TL a escala escolhida foi de 1 (um) segundo e resultou em uma série de 1759 pontos, o que corresponde a aproximadamente 29min31s de tráfego. As Figuras. ?? e ?? exibem as séries temporais originadas, já normalizadas para efeito de treinamento.

Nas Figuras ?? e ?? podemos observar uma clara diferença no comportamento das duas séries. Na série da Figura ?? percebemos um comportamento mais uniforme do tráfego, à exceção dos altos picos ocorridos, manifestando a presença de uma maior correlação entre as amostras da série. Na série da Figura ?? observamos um comportamento mais próximo do ruído branco, sugerindo uma maior descorrelação entre as amostras. Essa diferença observada entre as séries é comprovada pelos diferentes valores de H obtidos para cada uma delas no Capítulo 4.

Seguindo o procedimento padrão adotado no processo de treinamento de redes neurais, subdividimos cada uma das séries em outras três séries. Como de costume, selecionamos uma série para *treinamento*, outra para *validação cruzada* e outra para *teste*.

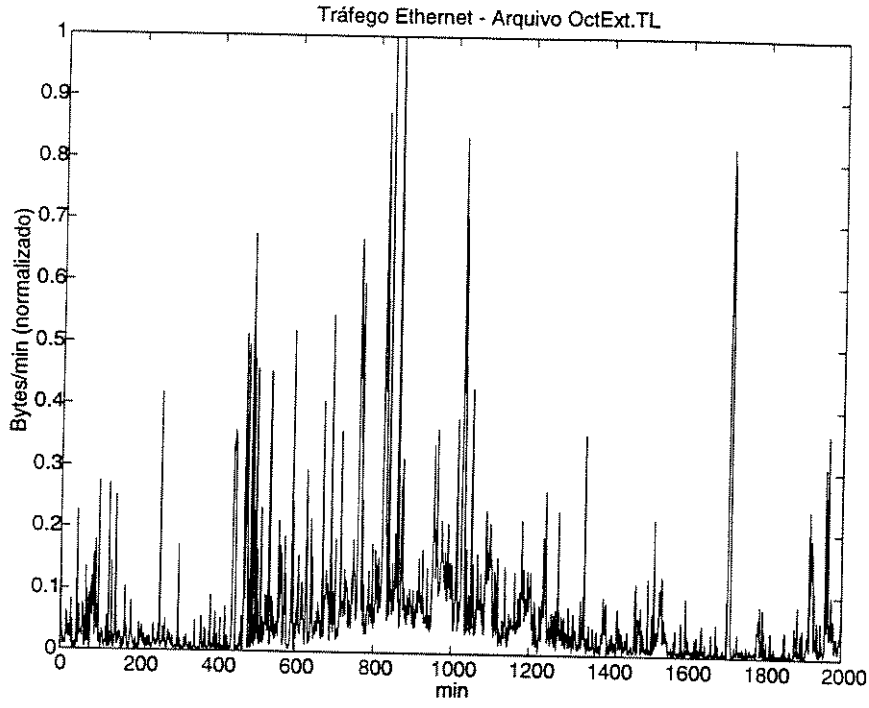

FIGURA 6.3: Série temporal normalizada obtida a partir do arquivo OctExt.TL agregando-se os pontos em intervalos de 1 (um) minuto.

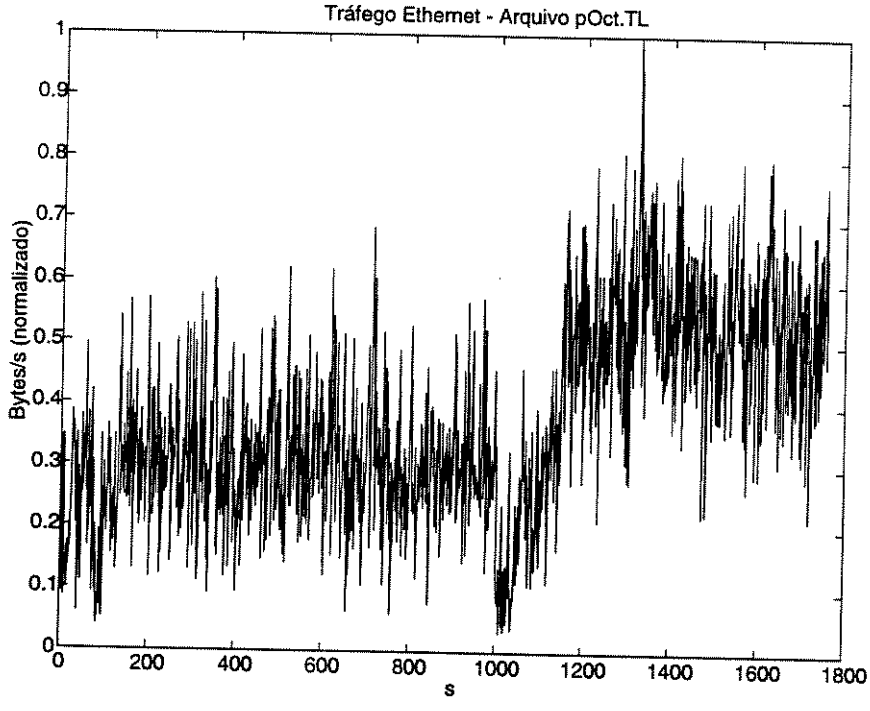

FIGURA 6.4: Série temporal normalizada obtida a partir do arquivo pOct.TL agregando-se os pontos em intervalos de 1 (um) segundo.

A série de *treinamento*, como o próprio nome indica, é utilizada na fase de ajuste dos pesos sinápticos (ou coeficientes do filtro). Nesta fase, a série é apresentada diversas vezes à rede, enquanto esta atualiza seus pesos sinápticos.

A série de *validação* corresponde também a uma série que é utilizada durante o processo de treinamento. No entanto, esta é apresentada à rede sem que ocorra atualização dos pesos. Ou seja, apenas é verificada a saída da rede quando estimulada por essas entradas, para que seja possível a avaliação da capacidade de generalização da rede neural ao longo do treinamento. Tipicamente, durante o treinamento, a curva de erro quadrático médio calculado sobre esta série apresenta um ponto mínimo correspondente ao ponto em que a rede apresenta sua melhor capacidade de generalização. É justamente neste ponto que devemos parar o treinamento.

Por fim, a série de *teste* é utilizada após o treinamento da rede, para que se possa avaliar o seu desempenho quando apresentada a entradas não utilizadas na fase de treinamento, verificando-se assim sua capacidade de generalização.

No caso da série OctExt.TL, temos o seguinte conjunto de séries, mostradas nas Figuras. 6.5, 6.6, 6.7 :

- Série de treino : formada pelos elementos 200 a 699;
- Série de validação : formada pelos elementos 700 a 999;
- Série de teste : formada pelos elementos 1000 a 1399.

No caso da série pOct.TL, tomamos o seguinte conjunto de séries, mostradas nas Figuras. 6.8, 6.9 e 6.10 :

- Série de treino : formada pelos elementos 800 a 1299 (a fim de incluir a variação abrupta em torno de 1100);
- Série de validação : formada pelos elementos 300 a 799;
- Série de teste : formada pelos elementos 1300 a 1699.

A fim de avaliarmos a qualidade das predições, adotaremos duas medidas de desempenho baseadas no erro quadrático médio. A primeira delas corresponde ao erro quadrático médio *normalizado* pela variância, da forma

$$\mathrm{EQMN}_1 = \frac{1}{\sigma^2 N} \sum_{k=1}^{N} [y(k) - \widehat{y}(k)]^2 , \tag{6.7}$$

onde $y(k)$ é o valor verdadeiro da seqüência, $\widehat{y}(k)$ é a predição e σ^2 é a variância amostral da seqüência verdadeira sobre o intervalo de duração da predição. Um valor de $\mathrm{EQMN}_1 = 1$ corresponde, simplesmente, a predizer o valor médio da seqüência.

A segunda medida de desempenho é dada por

$$\mathrm{EQMN}_2 = \frac{\sum_{k=1}^{N} [y(k) - \widehat{y}(k)]^2}{\sum_{k=1}^{N} [y(k) - y(k-1)]^2} . \tag{6.8}$$

Nesta medida, comparamos nosso preditor com um mais simples, cuja predição é o último valor observado da série. Um valor de EQMN_2 maior do que 1 (um) corresponde a uma predição que é pior do que estimar pelo preditor ótimo do "passeio aleatório", cujos incrementos correspondem a uma seqüência de variáveis aleatórias i.i.d., como por exemplo, o ruído branco Gaussiano de variância σ^2 (a variância amostral da série verdadeira sobre o intervalo de duração da predição).

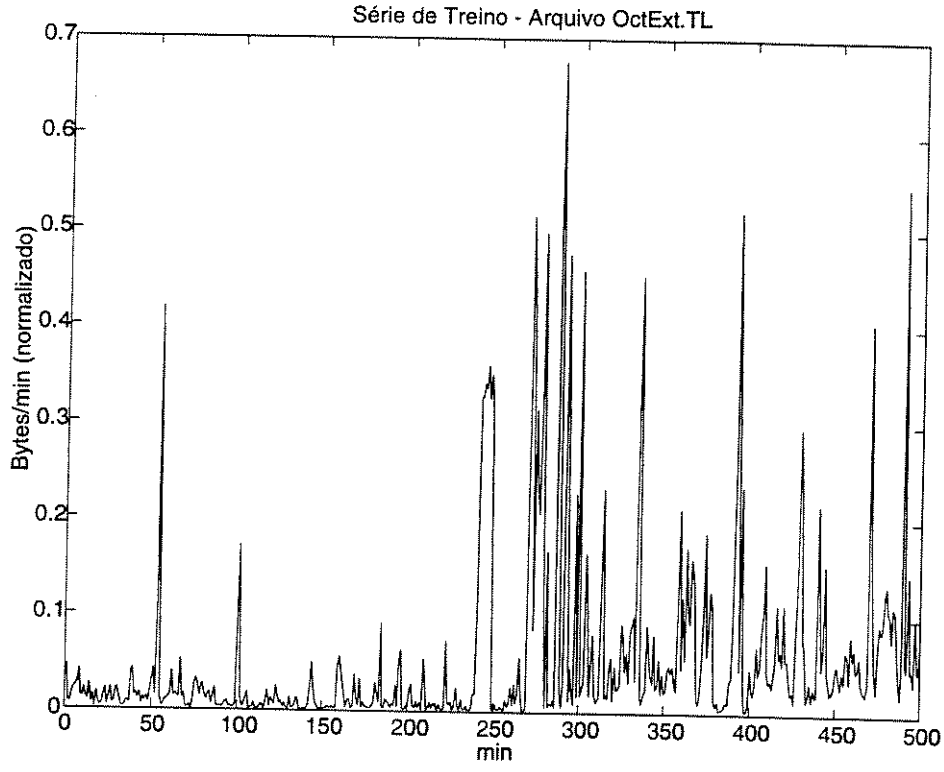

FIGURA 6.5: Série temporal de treinamento obtida a partir do arquivo OctExt.TL.

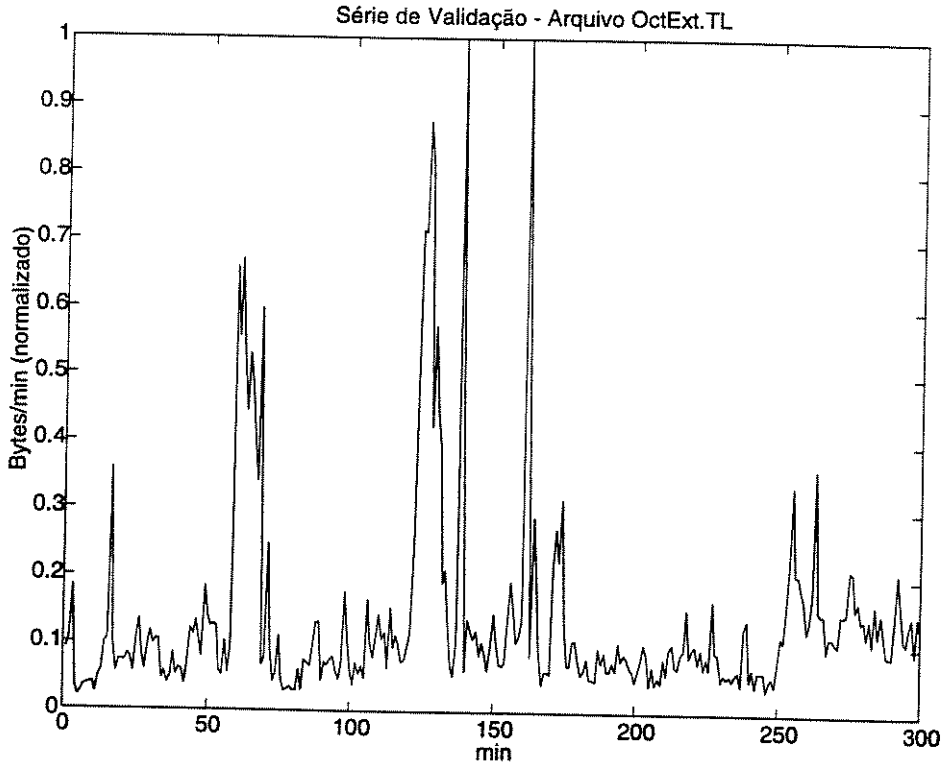

FIGURA 6.6: Série temporal de validação obtida a partir do arquivo OctExt.TL.

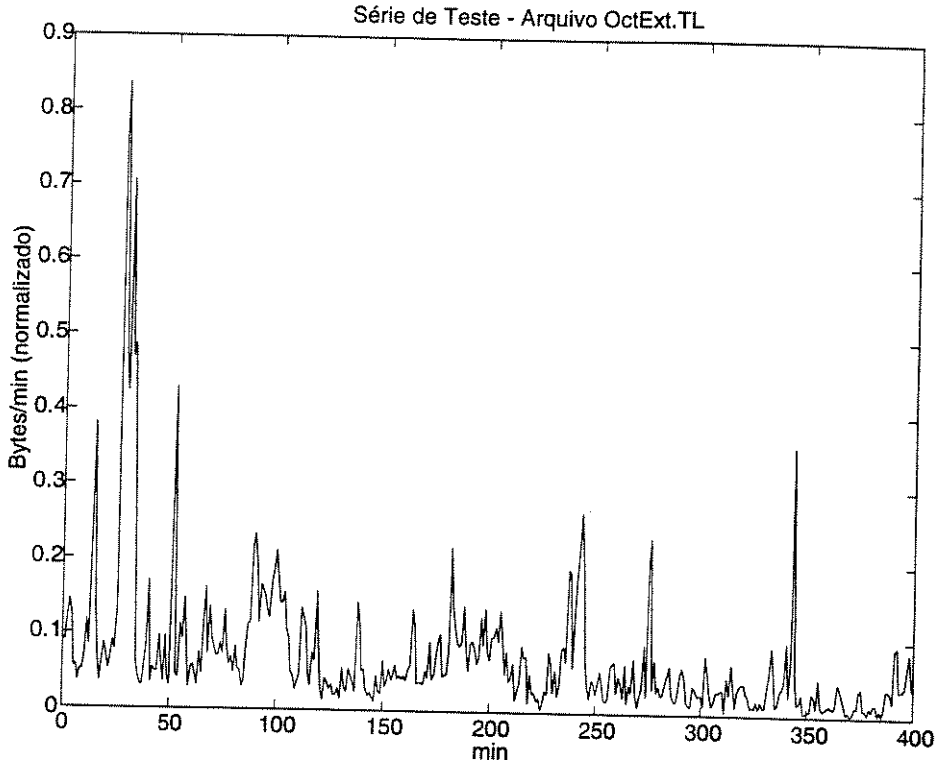

FIGURA 6.7: Série temporal de teste obtida a partir do arquivo OctExt.TL.

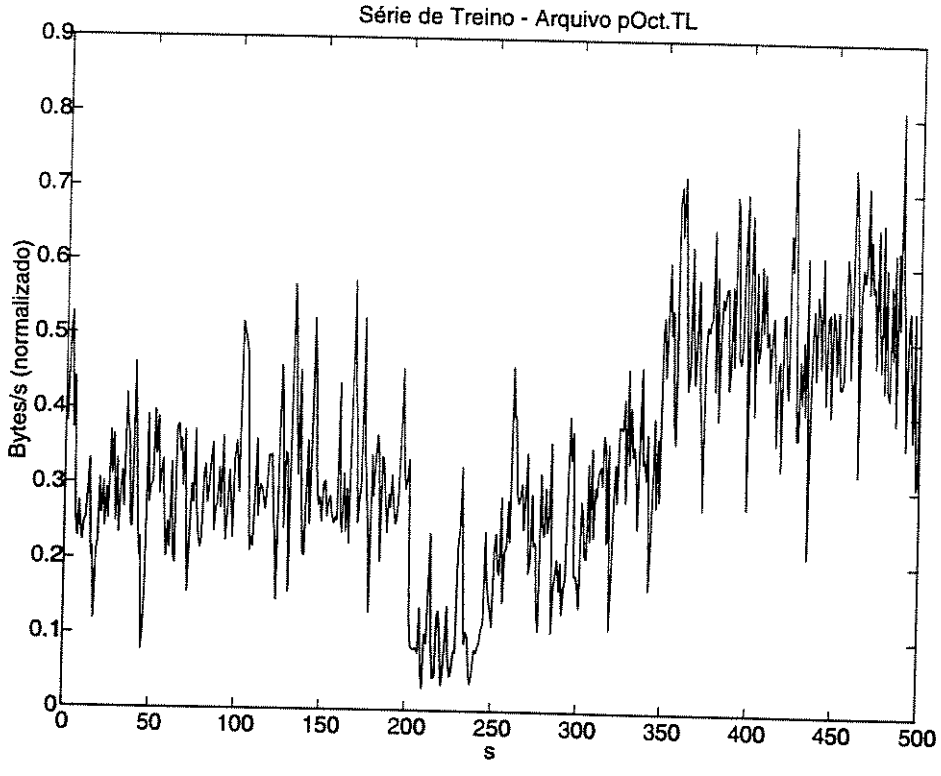

FIGURA 6.8: Série temporal de treinamento obtida a partir do arquivo pOct.TL

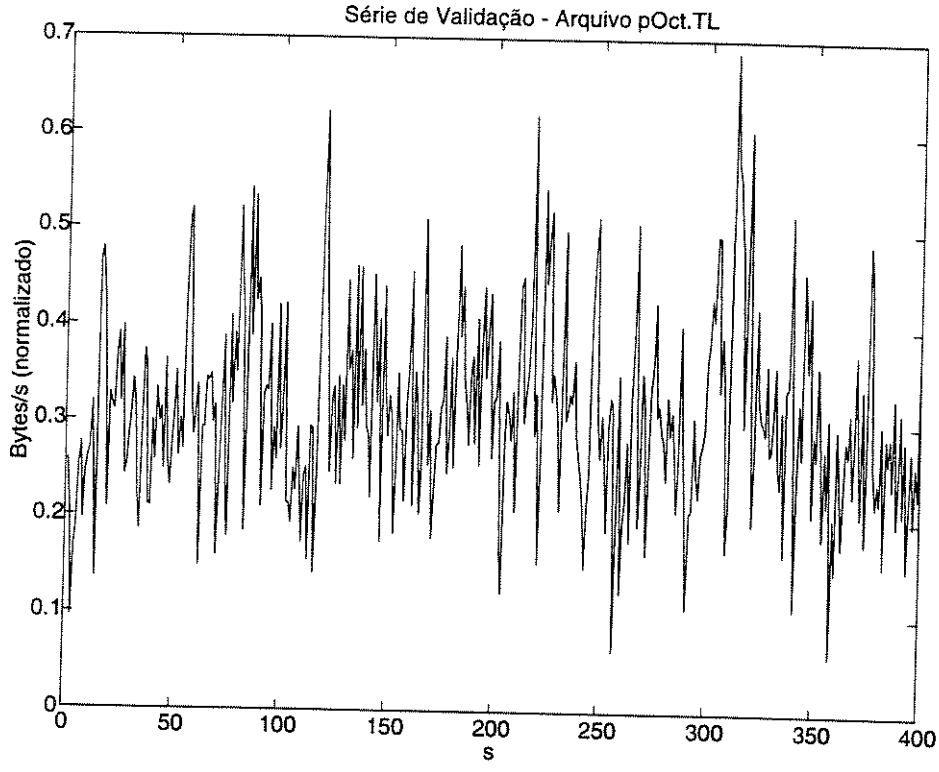

FIGURA 6.9: Série temporal de validação obtida a partir do arquivo pOct.TL.

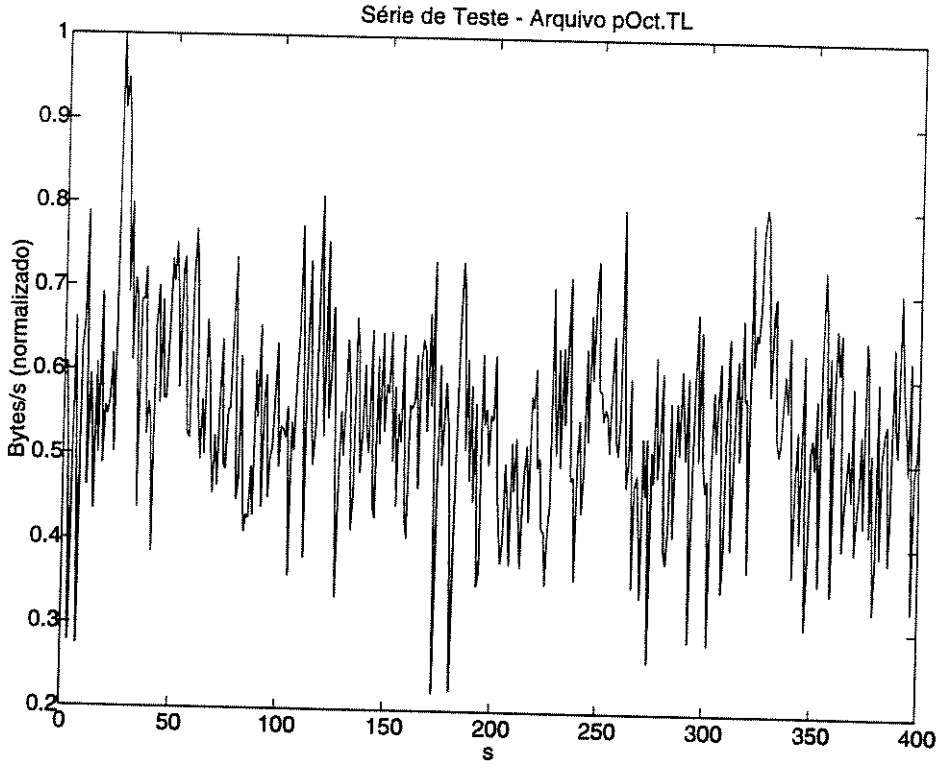

FIGURA 6.10: Série temporal de teste obtida a partir do arquivo pOct.TL.

6.4 Predições

6.4.1 Predição Linear

A predição através de um filtro linear foi obtida aplicando-se o algoritmo LMS para ajuste adaptativo dos pesos do filtro. Para fins de comparação com as redes neurais, calculamos o erro quadrático médio normalizado apenas sobre a série de teste. A seguir, apresentamos os resultados obtidos para cada série.

Desempenho da Predição - Série OctExt.TL

Na série OctExt.TL, a ordem do filtro que apresentou o melhor resultado foi de $M = 20$. Os pesos foram atualizados segundo uma taxa de $\mu = 0,01$. Na Fig. 6.11 temos uma comparação entre a série original e a série predita. Neste caso, os erros quadráticos médios normalizados obtidos foram

$$\text{EQMN}_1 = 0,4769 \quad \text{e} \quad \text{EQMN}_2 = 0,9182\,.$$

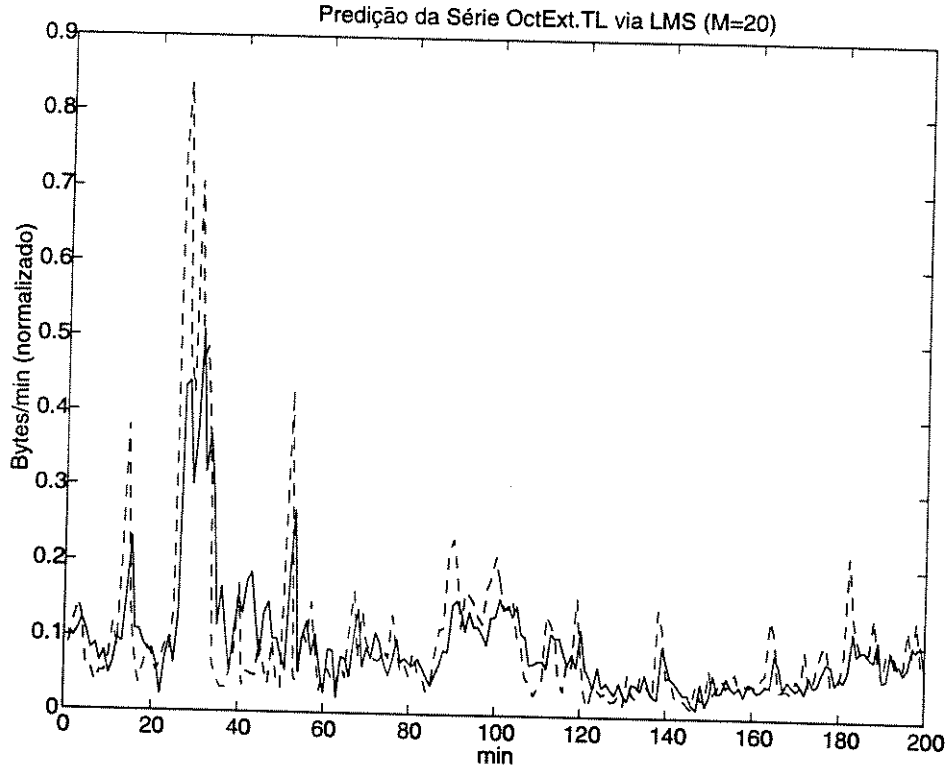

FIGURA 6.11: Predição da série de teste relativa ao arquivo OctExt.TL via LMS ($M = 20$). Série real (linha tracejada), série predita (linha cheia).

Desempenho da Predição - Série pOct.TL

Para este caso, a melhor ordem utilizada correspondeu a $M = 5$. Os pesos foram também atualizados segundo uma taxa de 0,01. Na Fig. ?? podemos visualizar a predição realizada

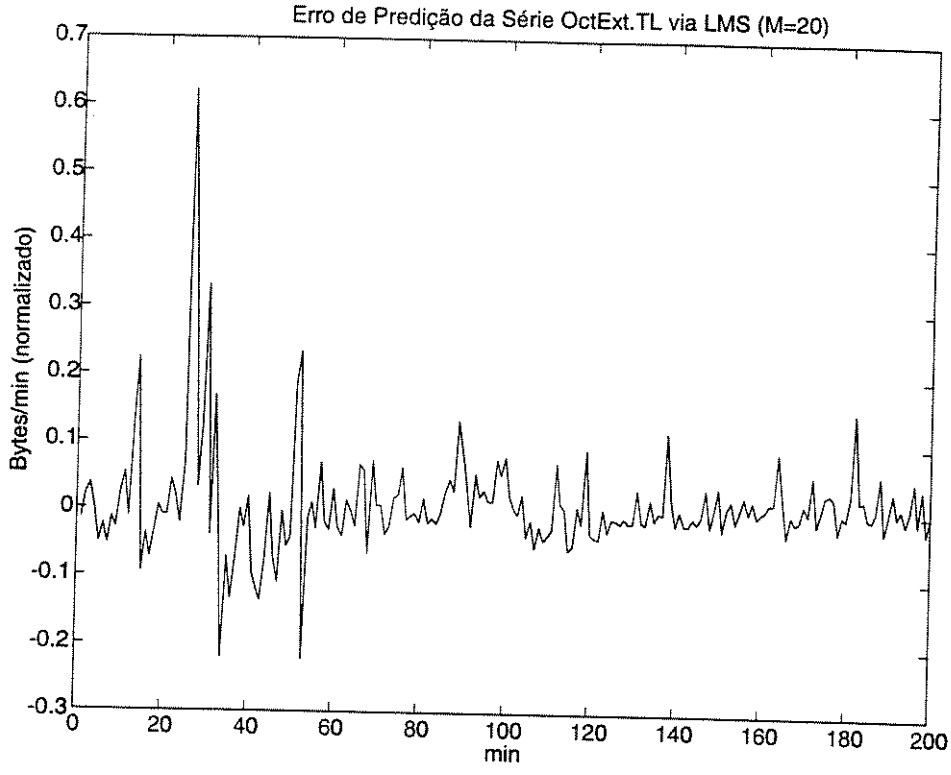

FIGURA 6.12: Sinal de erro de predição via LMS relativo à série OctExt.TL.

comparada com a série real. Os erros quadráticos médios normalizados, obtidos neste caso, foram

$$\text{EQMN}_1 = 0,93368 \qquad \text{e} \qquad \text{EQMN}_2 = 0,720081\,.$$

6.4.2 Perceptron Multicamadas FIR

A predição através do perceptron multicamadas FIR apresentou um pequeno ganho com relação à predição linear. A arquitetura utilizada foi a mesma em ambos os casos (pOct.TL e OctExt.TL), consistindo de 3 neurônios na primeira camada, 2 neurônios na camada escondida (intermediária) e 1 neurônio na saída. Foram testadas outras arquiteturas, mas esta se destacou por apresentar um bom desempenho a um custo computacional relativamente baixo. Em geral, verificou-se que, à medida que aumentamos o número de neurônios da rede, obtemos um melhor desempenho. No entanto, maior será o custo computacional envolvido para a realização do aprendizado através do algoritmo de retropropagação temporal.

Desempenho da Predição - Série OctExt.TL

Para o caso da série OctExt.TL os parâmetros para as quais obtivemos os melhores resultados foram $M_1 = 20$ para os filtros da primeira camada e $M_2 = 5$ para os neurônios da segunda camada. As taxas de aprendizado utilizadas foram as mesmas para as duas camadas e iguais a $\eta = 0,01$. Nessas condições, obtivemos o seguinte desempenho (em relação à série de teste) :

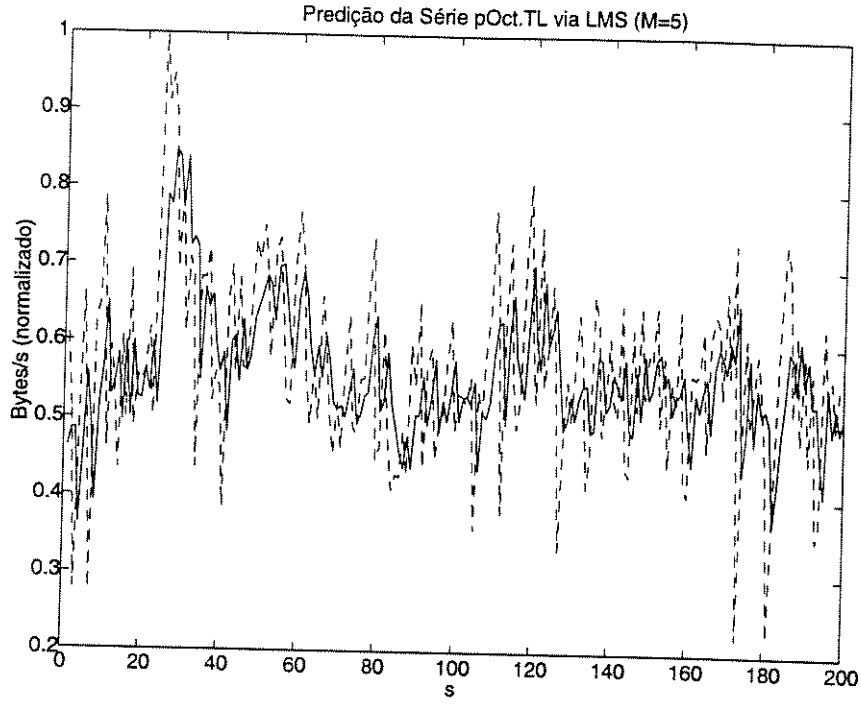

FIGURA 6.13: Predição da série de teste relativa ao arquivo pOct.TL via LMS (M=5). Série real (linha tracejada) e série predita (linha cheia).

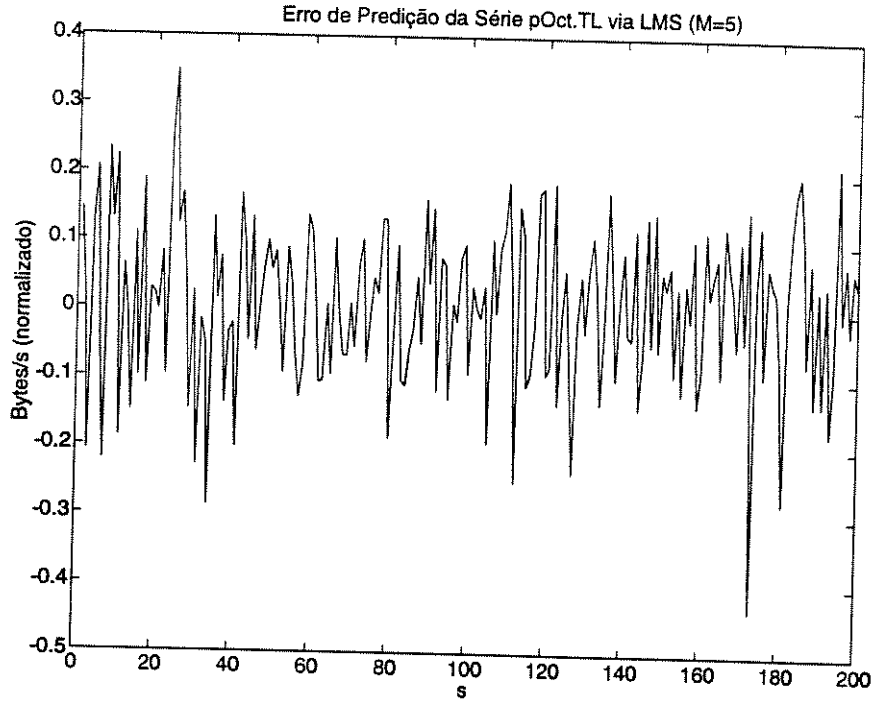

FIGURA 6.14: Sinal de erro de predição via LMS relativo ao arquivo pOct.TL .

$$\text{EQMN}_1 = 0,4736 \qquad \text{e} \qquad \text{EQMN}_2 = 0,9096\,.$$

A predição das primeiras 200 amostras da série de teste está mostrada na Fig. 6.15.

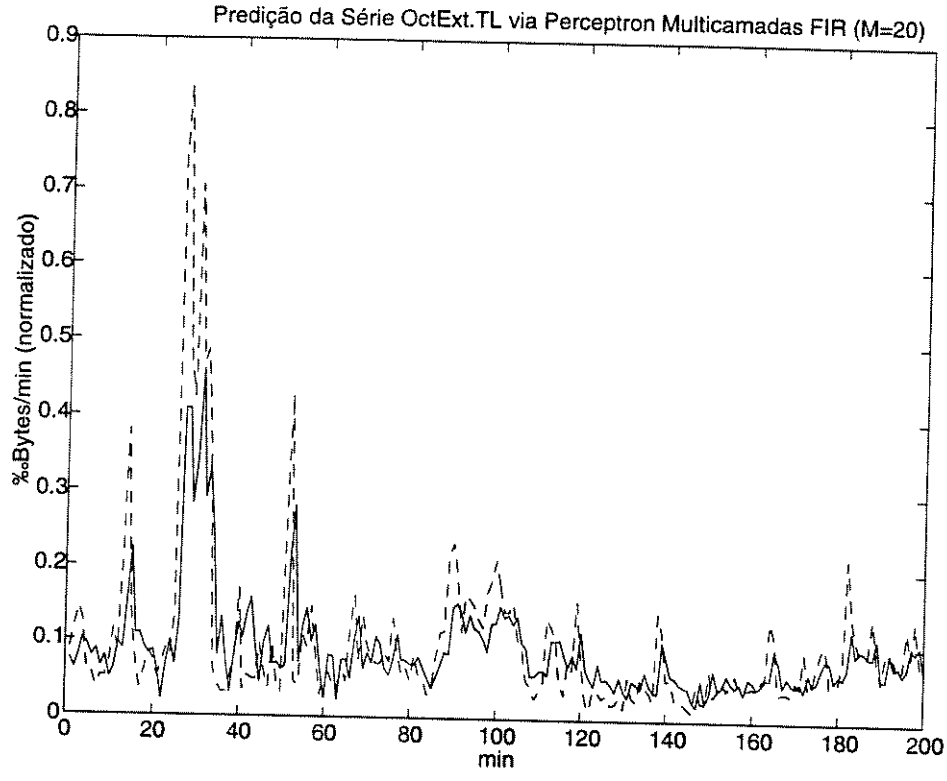

FIGURA 6.15: Predição da série de teste relativa ao arquivo OctExt.TL via redes neurais FIR. Série real (linha tracejada) e predita (linha cheia).

Desempenho da Predição - Série pOct.TL

No caso da predição da série pOct.TL, verificamos que tivemos de utilizar ordens de predição mais baixas, de forma que o melhor desempenho obtido correspondeu a filtros com ordens $M_1 = 5$ e $M_2 = 3$. As taxas utilizadas foram também iguais a $\eta = 0,01$. Nessas condições, obtivemos o seguinte desempenho :

$$\text{EQMN}_1 = 0,9031 \qquad \text{e} \qquad \text{EQMN}_2 = 0,6965$$

A predição das primeiras 200 amostras da série de teste está mostrada na Fig. 6.17.

6.4.3 Rede RBF Dinâmica

A predição através da rede RBF dinâmica apresentou um ganho em relação ao outros preditores nas medidas de desempenho. No entanto, tivemos de trabalhar com ordens maiores de predição.

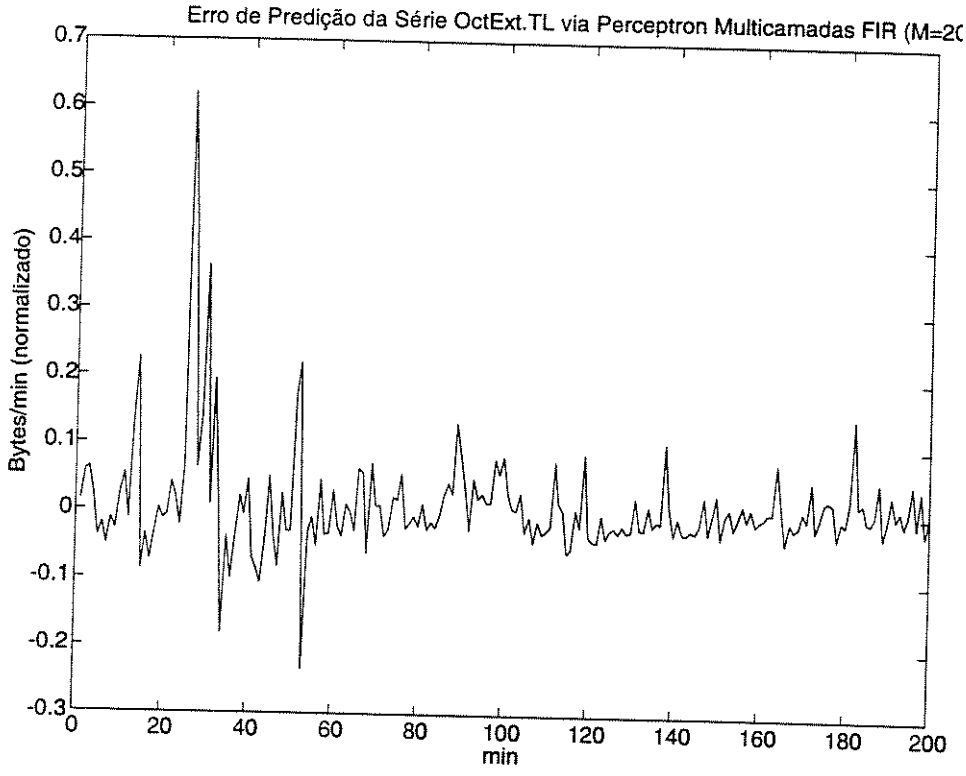

FIGURA 6.16: Sinal de erro de predição via perceptron multicamadas FIR relativo à série OctExt.TL.

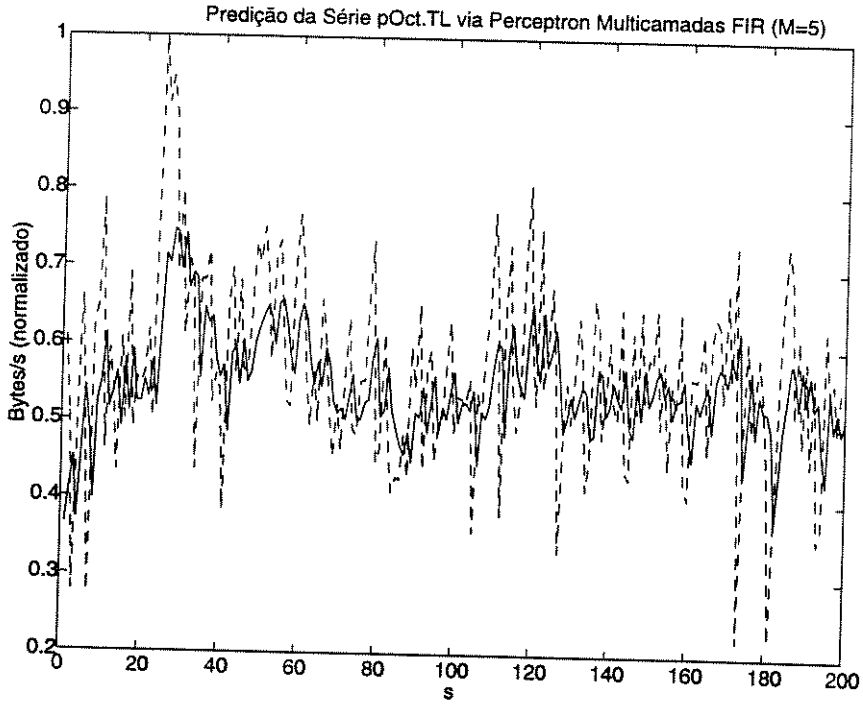

FIGURA 6.17: Predição da série de teste relativa ao arquivo pOct.TL via redes neurais FIR. Série real (linha tracejada), série predita (linha cheia).

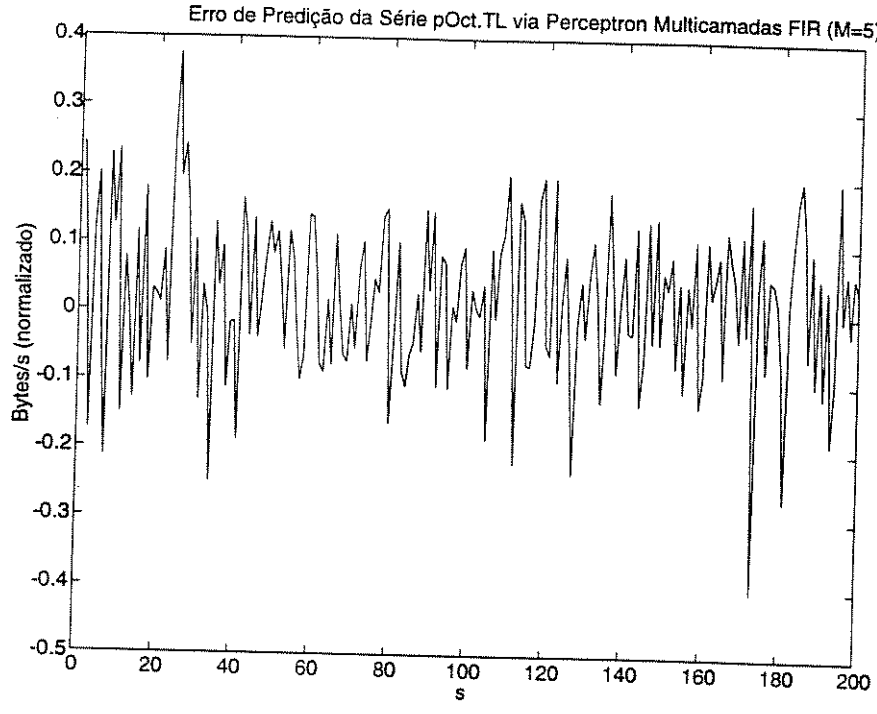

FIGURA 6.18: Sinal de erro de predição via perceptron multicamadas FIR relativo à série pOct.TL.

Desempenho da Predição - Série OctExt.TL

Para o caso da série OctExt.TL, os parâmetros que nos forneceram os melhores resultados foram $K = 20$, $M = 300$, $\lambda = 10^{-8}$ e $\alpha = 400$. Nessas condições, obtivemos o seguinte desempenho :

$$\text{EQMN}_1 = 0,4582 \qquad \text{e} \qquad \text{EQMN}_2 = 0,8823\,.$$

As primeiras 200 amostras preditas da série de teste estão mostradas na Fig. 6.19.

Desempenho da Predição - Série pOct.TL

No caso da série pOct.TL, os parâmetros utilizados foram $K = 30$, $M = 300, \alpha = 30$ e $\lambda = 10^{-8}$. Nessas condições, obtivemos o seguinte desempenho :

$$\text{EQMN}_1 = 0,8983 \qquad \text{e} \qquad \text{EQMN}_2 = 0,6946\,.$$

As primeiras 200 amostras preditas da série de teste estão mostradas na Fig. 6.21.

6.5 Algoritmo EM Adaptativo

Nesta seção, avaliamos o desempenho do *algoritmo EM adaptativo*, proposto neste trabalho, comparado ao *algoritmo EM padrão* modificado por uma função de penalidade. Para fins de estimação, utilizamos uma seqüência de pontos no espaço bidimensional referente a

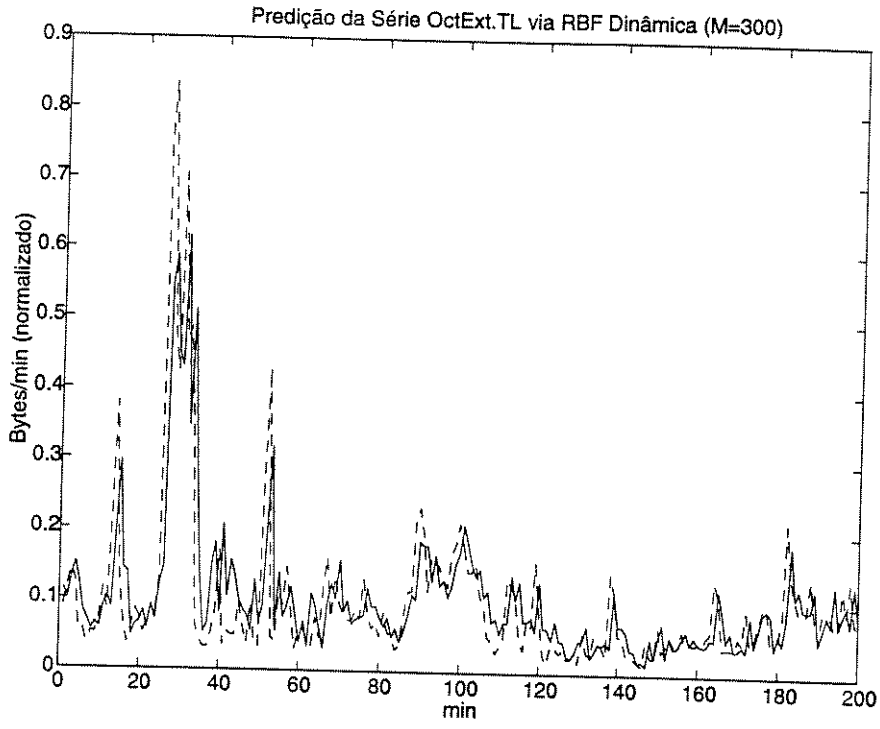

FIGURA 6.19: Predição da série de teste relativa ao arquivo OctExt.TL via RBF dinâmica. Série real (linha tracejada) e predita (linha cheia).

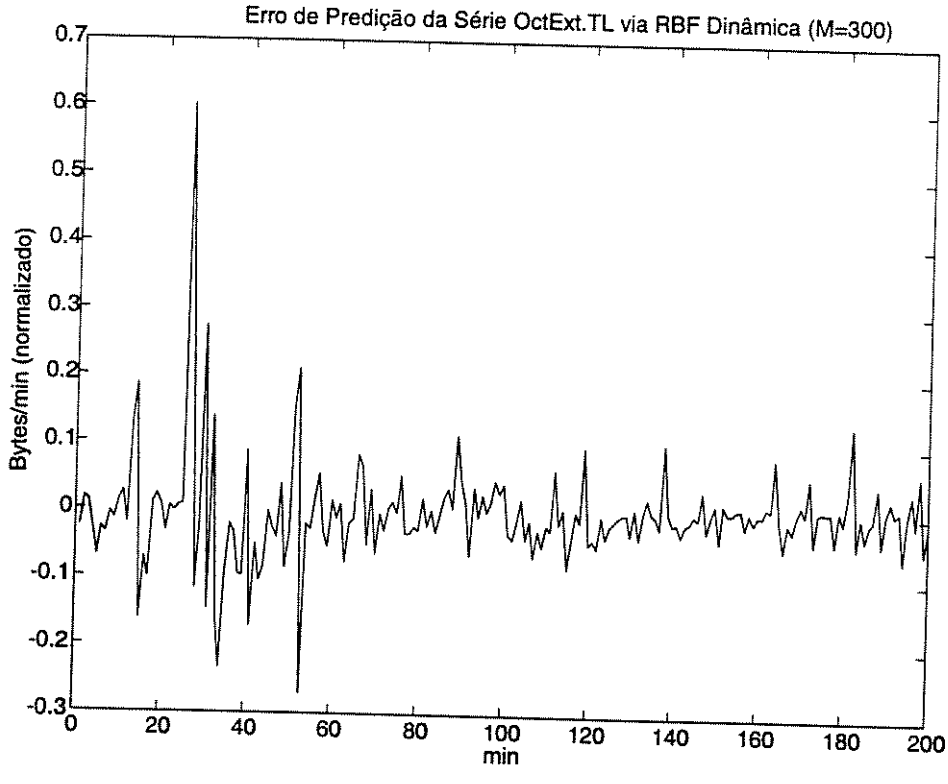

FIGURA 6.20: Sinal de erro de predição via RBF Dinâmica relativo à série OctExt.TL.

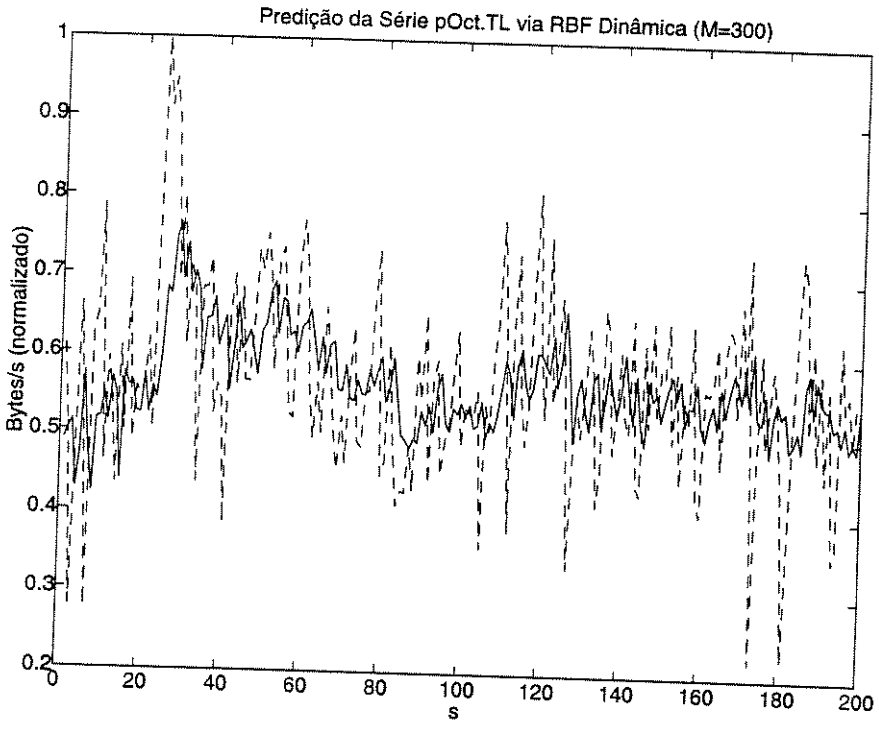

FIGURA 6.21: Predição da série de teste relativa ao arquivo pOct.TL via RBF dinâmica. Série real (linha tracejada) e série predita (linha cheia).

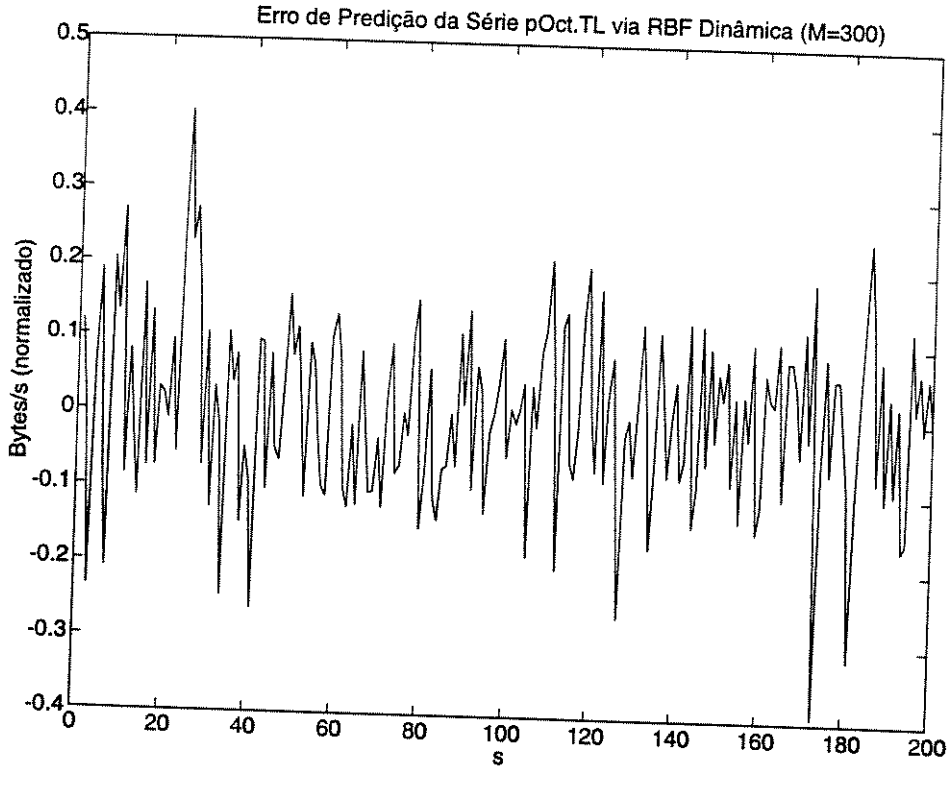

FIGURA 6.22: Sinal de erro de predição via RBF Dinâmica relativo à série pOct.TL.

uma mistura de quatro Gaussianas, todas com a mesma probabilidade ($p = 0,25$) e mesmas matrizes de covariância (identidade). As médias escolhidas para as quatro Gaussianas foram

$$\mu_1 = (4,4)\,, \quad \mu_2 = (-4,4)\,, \quad \mu_3 = (4,-4) \quad \text{e} \quad \mu_4 = (-4,-4)\,.$$

Foram gerados 2000 pontos, mostrados graficamente na Figura 6.23.

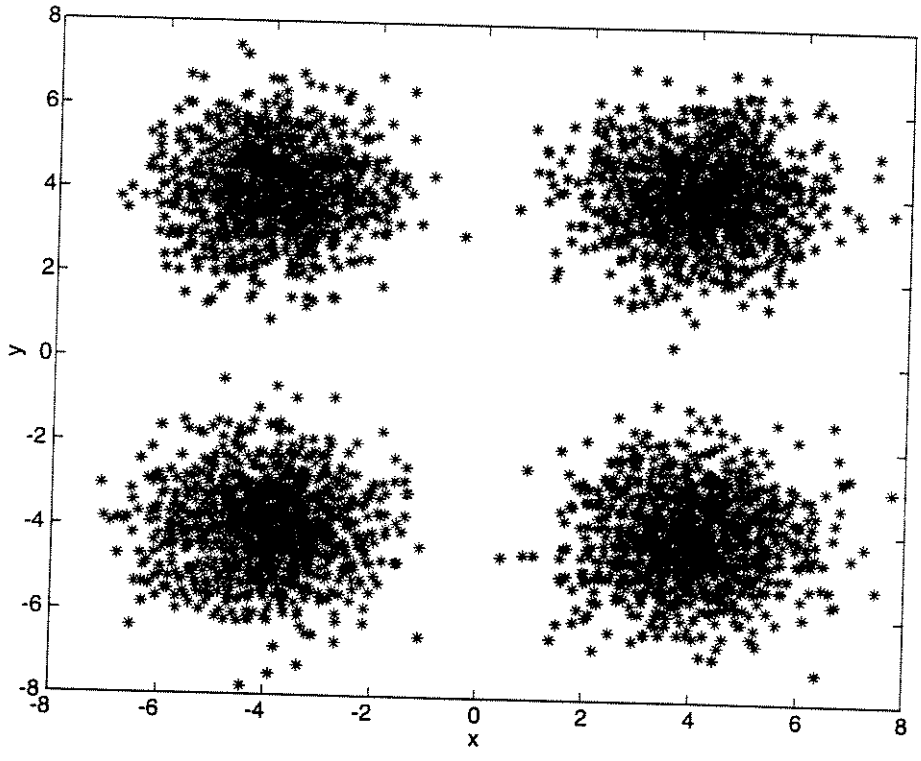

FIGURA 6.23: Representação gráfica de 2000 pontos gerados segundo uma mistura de quatro Gaussianas.

Realizamos então três experiências, cada uma das quais correspondendo a diferentes escolhas das condições iniciais fornecidas aos algoritmos. Em cada uma dessas experiências, comparamos os resultados obtidos para os dois algoritmos. É importante salientar que os resultados que seguem constituem uma amostra típica dos resultados obtidos ao se executar os algoritmos nas condições iniciais fornecidas, de modo que, eventualmente, estes resultados podem vir a ser totalemente diferentes. As condições das experiências, bem como seus resultados, estão descritos a seguir.

6.5.1 Experiência 1

Nesta experiência, as médias iniciais fornecidas a ambos os algoritmos foram pontos escolhidos voluntariamente, próximos aos centros verdadeiros das distribuições Gaussianas. As médias iniciais escolhidas foram

$$\mu_1^0 = (3,8,\ 3,2)\,, \quad \mu_2^0 = (-3,1,\ 4,7)\,, \quad \mu_3^0 = (4,8,\ -3,7) \quad \text{e} \quad \mu_4^0 = (-4,5,\ -3,5)\,.$$

Quanto às condições iniciais das matrizes de covariância e das probabilidades das Gaussianas, utilizamos matrizes identidades e probabilidades iguais a 0,25. Aplicando-se o *algoritmo EM padrão* com função de penalidade obtivemos o seguinte resultado (após 50 iterações) :

- **Médias :**

$$\mu_1^{padrão} = (3,9871,\ 4,0269)\,, \quad \mu_2^{padrão} = (-3,9299,\ 4,0202)\,,$$
$$\mu_3^{padrão} = (4,0328,\ -4,0651) \quad \mu_4^{padrão} = (-3,9609,\ -4,0206)\,.$$

- **Matrizes de Covariância :**

$$\Lambda_1^{padrão} = \begin{pmatrix} 1,0836 & 0,0322 \\ 0,0322 & 0,9996 \end{pmatrix}, \Lambda_2^{padrão} = \begin{pmatrix} 0,9445 & 0,0143 \\ 0,0144 & 1,0149 \end{pmatrix},$$
$$\Lambda_3^{padrão} = \begin{pmatrix} 0,9538 & 0,0281 \\ 0,0281 & 1,0381 \end{pmatrix} \text{e}\, \Lambda_4^{padrão} = \begin{pmatrix} 0,9986 & 0,0314 \\ 0,0314 & 0,9301 \end{pmatrix}.$$

- **Probabilidades das Gaussianas :**

$$\alpha_1^{padrão} = 0,2480, \quad \alpha_2^{padrão} = 0,2555, \quad \alpha_3^{padrão} = 0,2445, \quad \alpha_4^{padrão} = 0,2519.$$

Aplicando-se o *algoritmo EM adaptativo*, obtivemos o seguinte resultado :

- **Médias :**

$$\mu_1^{adapt.} = (4,0789,\ 4,0987)\,, \quad \mu_2^{adapt.} = (-3,9693,\ 3,9451)\,,$$
$$\mu_3^{adapt.} = (4,2038,\ -3,9290) \quad \mu_4^{adapt.} = (-3,9494,\ -3,9909)\,.$$

- **Matrizes de Covariância :**

$$\Lambda_1^{adapt.} = \begin{pmatrix} 0,7783 & 0,0474 \\ 0,0474 & 0,8488 \end{pmatrix}, \Lambda_2^{adapt.} = \begin{pmatrix} 1,0919 & 0,0439 \\ 0,0439 & 0,6461 \end{pmatrix},$$
$$\Lambda_3^{adapt.} = \begin{pmatrix} 1,0914 & 0,0710 \\ 0,0710 & 1,3550 \end{pmatrix} \text{e}\, \Lambda_4^{adapt.} = \begin{pmatrix} 1,0414 & -0,1287 \\ -0,1287 & 1,0059 \end{pmatrix}.$$

- **Probabilidades das Misturas :**

$$\alpha_1^{adapt.} = 0,21469, \quad \alpha_2^{adapt.} = 0,2800, \quad \alpha_3^{adapt.} = 0,2326, \quad \alpha_4^{adapt.} = 0,2727$$

Os valores das médias das Gaussianas obtidos através dos dois algoritmos podem ser visualizados na Figura 6.24.

Pelo o que foi obtido, podemos constatar que os dois algoritmos convergem para valores muito próximos dos valores reais dos parâmetros da mistura. Neste caso, a vantagem de se utilizar o algoritmo EM adaptativo é justamente o baixo custo computacional, uma vez que necessitamos apenas de apresentar uma única vez os dados da série, o que não acontece no caso do algoritmo EM padrão, o qual precisa ser aplicado a todo o conjunto de dados um certo número de vezes (50 iterações, no caso).

6.5.2 Experiência 2

Nesta segunda experiência, as médias iniciais foram escolhidas aleatoriamente sobre o conjunto de dados disponível. Quanto às matrizes de covariância e às probabilidades iniciais, repetimos as condições anteriores, ou seja, matrizes identidades e probabilidades iguais a $0,25$. Aplicando-se o *algoritmo EM padrão*, com função de penalidade, obtivemos o seguinte resultado (após 50 iterações) :

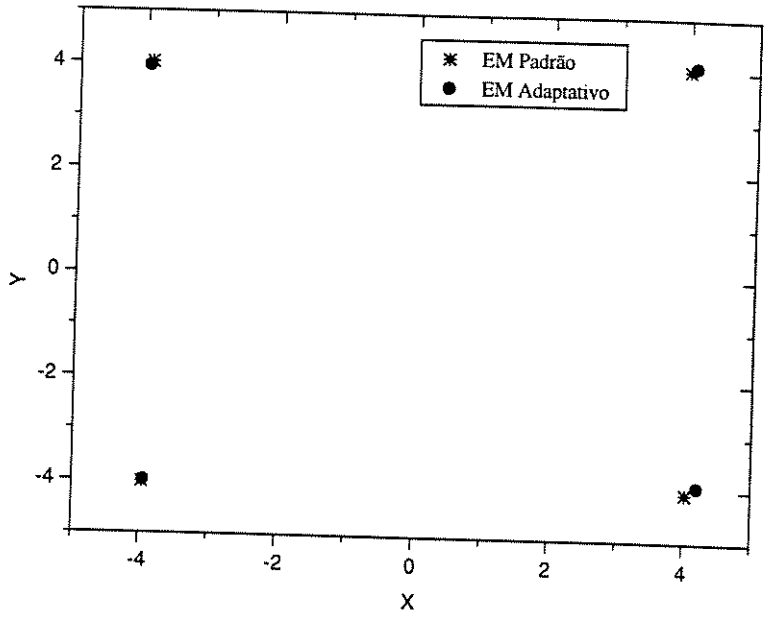

FIGURA 6.24: Gráfico das estimações obtidas para as médias das Gaussianas através dos algoritmos EM padrão e EM adaptativo (Experiência 1).

- **Médias** :

$$\mu_1^{padrão} = (4,0242,\ 4,0375)\,, \quad \mu_2^{padrão} = (-3,9687,\ 3,9930)\,,$$
$$\mu_3^{padrão} = (-3,9936,\ -4,0116) \quad \mu_4^{padrão} = (4,0069,\ -4,0782)\,.$$

- **Matrizes de Covariância** :

$$\Lambda_1^{padrão} = \begin{pmatrix} 1,0607 & 0,0130 \\ 0,0130 & 0,9740 \end{pmatrix}, \Lambda_2^{padrão} = \begin{pmatrix} 0,9665 & 0,0125 \\ 0,0125 & 0,9664 \end{pmatrix},$$
$$\Lambda_3^{padrão} = \begin{pmatrix} 0,9926 & 0,0115 \\ 0,0115 & 0,9898 \end{pmatrix} \text{ e } \Lambda_4^{padrão} = \begin{pmatrix} 1,0498 & 0,0166 \\ 0,0166 & 1,0092 \end{pmatrix}.$$

- **Probabilidades das Gaussianas** :

$$\alpha_1^{padrão} = 0,2462, \quad \alpha_2^{padrão} = 0,2474, \quad \alpha_3^{padrão} = 0,2556, \quad \alpha_4^{padrão} = 0,2508.$$

Aplicando-se o algoritmo EM adaptativo, obtivemos :

- **Médias** :

$$\mu_1^{adapt.} = (5,0119,\ -4,0409)\,, \quad \mu_2^{adapt.} = (-3,8887,\ 4,4419)\,,$$
$$\mu_3^{adapt.} = (4,3182,\ 3,9320) \quad \mu_4^{adapt.} = (-0,0393,\ -0,0429)\,.$$

- **Matrizes de Covariância :**

$$\Lambda_1^{adapt.} = \begin{pmatrix} 0,0023 & -0,00004 \\ -0,00004 & 0,00235 \end{pmatrix}, \Lambda_2^{adapt.} = \begin{pmatrix} 0,00424 & 0,00002 \\ 0,00002 & 0,00374 \end{pmatrix},$$
$$\Lambda_3^{adapt.} = \begin{pmatrix} 0,00034 & 2,055 \times 10^{-7} \\ 2,055 \times 10^{-7} & 0,00034 \end{pmatrix} \text{ e } \Lambda_4^{adapt.} = \begin{pmatrix} 17,2290 & 0,6446 \\ 0,6446 & 18,3261 \end{pmatrix}.$$

- **Probabilidades das Gaussianas :**

$$\alpha_1^{adapt.} = 0,0001, \quad \alpha_2^{adapt.} = 0,0001, \quad \alpha_3^{adapt.} = 0,0006, \quad \alpha_4^{adapt.} = 0,9992$$

Os valores obtidos para as médias das Gaussianas pode ser visualizado na Figura 6.25.

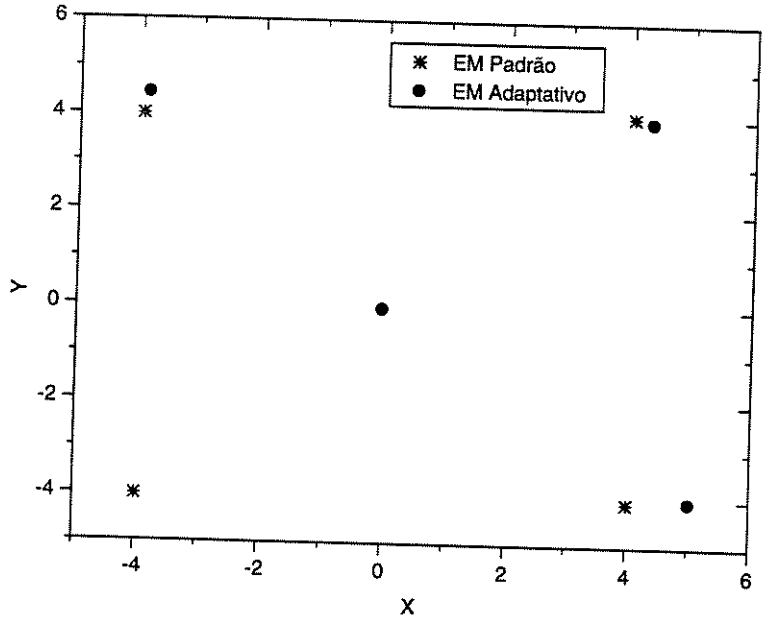

FIGURA 6.25: Gráfico das estimações obtidas para as médias das Gaussianas através dos algoritmos EM padrão e EM adaptativo (Experiência 2).

Nesta experiência, o algoritmo EM adaptativo apresentou um desempenho inferior ao algoritmo EM padrão com função de penalidade. Apesar de estimar, aproximadamente, três dos quatro centros da mistura, o algoritmo EM falhou ao estimar as matrizes de covariância e as probabilidades. Neste caso, percebemos que o algoritmo EM adaptativo fixou um centro entre as quatro Gaussianas, de forma a representar o conjunto de dados como se fossem provenientes de uma única Gaussiana. Esta experiência mostra bem a sensibilidade do algoritmo EM adaptativo às condições iniciais. Vale ressaltar, novamente, que essa estimação é baseada em uma única leitura (seqüencial) dos dados e que, mesmo assim, o algoritmo EM adaptativo pôde dar uma estimação, se não verdadeira, coerente com uma distribuição simétrica de pontos no espaço.

6.5.3 Experiência 3

Nesta experiência escolhemos valores diferentes para as probabilidades iniciais das Gaussianas. Além disso, as médias iniciais foram obtidas aleatoriamente, segundo uma distribuição uniforme, sobre o conjunto de dados disponível. A única restrição feita foi que as médias iniciais não assumissem valores iguais. Os valores iniciais utilizados para as probabilidades foram

$$\alpha_1^0 = 0,1, \quad \alpha_2^0 = 0,3, \quad \alpha_3^0 = 0,4, \quad \alpha_4^0 = 0,3\,.$$

Nesta experiência, as matrizes de covariância iniciais foram também matrizes identidades. Os resultados obtidos para o *algoritmo EM padrão,* após 50 iterações, foram os seguintes :

- **Médias** :

$$\mu_1^{padrão} = (4.2909,\ -3.6571)\,, \quad \mu_2^{padrão} = (-3,9454,\ 0.0275)\,,$$
$$\mu_3^{padrão} = (3.9316,\ -4,2249) \quad \mu_4^{padrão} = (3.9871,\ 4.0269)\,.$$

- **Matrizes de Covariância** :

$$\Lambda_1^{padrão} = \begin{pmatrix} 1,2512 & -0.3569 \\ -0.3569 & 0,8953 \end{pmatrix}, \Lambda_2^{padrão} = \begin{pmatrix} 0,9716 & 0,0850 \\ 0,0144 & 17,1358 \end{pmatrix},$$
$$\Lambda_3^{padrão} = \begin{pmatrix} 0,8011 & 0,1214 \\ 0,1214 & 1,0033 \end{pmatrix} \text{ e } \Lambda_4^{padrão} = \begin{pmatrix} 1.0837 & 0,0322 \\ 0,0322 & 0,9996 \end{pmatrix}.$$

- **Probabilidades das Gaussianas** :

$$\alpha_1^{padrão} = 0,0688, \quad \alpha_2^{padrão} = 0,5075, \quad \alpha_3^{padrão} = 0,1757, \quad \alpha_4^{padrão} = 0,2480.$$

Para o caso do algoritmo EM adaptativo, os resultados obtidos foram :

- **Médias** :

$$\mu_1^{adapt.} = (4,0770,\ 4,1422)\,, \quad \mu_2^{adapt.} = (-3,2475,\ 4,1003)\,,$$
$$\mu_3^{adapt.} = (-0.3352,\ -0.0178) \quad \mu_4^{adapt.} = (4.0453,\ -4.0931)\,.$$

- **Matrizes de Covariância** :

$$\Lambda_1^{adapt.} = \begin{pmatrix} 0,0099 & -0,0029 \\ -0.0029 & 0,0081 \end{pmatrix}, \Lambda_2^{adapt.} = \begin{pmatrix} 0.0028 & -0.0003 \\ -0.0003 & 0,0031 \end{pmatrix},$$
$$\Lambda_3^{adapt.} = \begin{pmatrix} 17,1151 & -0.3356 \\ -0.3356 & 16.7227 \end{pmatrix} \text{ e } \Lambda_4^{adapt.} = \begin{pmatrix} 0.0050 & 0.0003 \\ 0.0003 & 0.0061 \end{pmatrix}.$$

- **Probabilidades das Gaussianas** :

$$\alpha_1^{adapt.} = 0,000042, \quad \alpha_2^{adapt.} = 0,000075, \quad \alpha_3^{adapt.} = 0,999840, \quad \alpha_4^{adapt.} = 0,000042$$

Os valores das médias das Gaussianas obtidos através dois algoritmos podem ser visualizados na Figura 6.26.

Esta experiência demonstra claramente o problema de ocorrência de mínimos locais comum aos dois algoritmos. Tanto o algoritmo EM padrão como o algoritmo EM adaptativo não foram capazes de realizar estimações corretas. No caso do algoritmo EM padrão, podemos

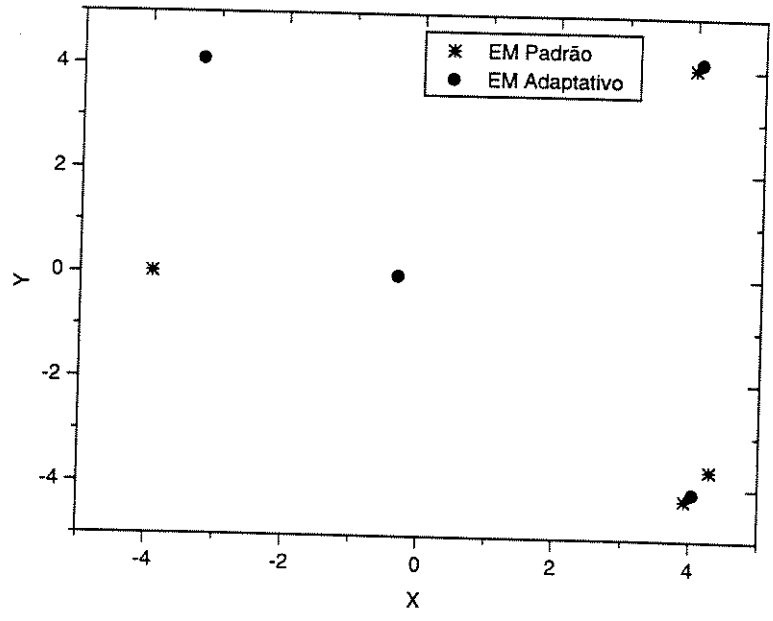

FIGURA 6.26: Gráfico das estimações obtidas para as médias das Gaussianas através dos algoritmos EM padrão e EM adaptativo (Experiência 3).

perceber que este fixou dois centros praticamente iguais, deixando de estimar um dos centros. Além disso, as matrizes de covariância e as probabilidades das misturas não correspondem exatamente aos valores desejados. No caso do algoritmo EM adaptativo, percebemos que este generalizou novamente a distribuição dos pontos no espaço, fixando uma Gaussiana no centro. É importante lembrarmos também que o algoritmo EM adaptativo "esquece" os valores mais distantes da série e realiza estimações baseadas nos valores mais atuais. Daí, caso ocorra a predominância de pontos representativos de uma dada Gaussiana, o algoritmo tende a adaptar-se aos valores dos parâmetros desta Gaussiana em particular, o que, eventualmente, pode vir a prejudicar a estimação correta de todos os parâmetros da mistura.

Capítulo 7

Conclusões

Dos resultados obtidos, podemos extrair algumas conclusões interessantes. Como uma primeira observação, podemos perceber que tanto a predição linear quanto a predição não-linear resultaram em desempenhos relativamente parecidos no que se refere à ordem de grandeza das medidas de desempenho $EQMN_1$e $EQMN_2$. Salvo alguns ganhos relativos obtidos, não foi possível obter grandes diferenças nas predições realizadas a partir desses preditores. No entanto, pelas medidas obtidas, todos forneceram um ganho de predição, ou seja, todos se mostraram eficientes quanto aos critérios utilizados para avaliação do desempenho de predição. Se os parâmetros forem ajustados apropriadamente, é possível obter predições com erro quadrático médio normalizado inferior à unidade.

Com relação aos preditores, podemos tecer alguns comentários. O *perceptron multicamadas FIR*, apesar de ser reconhecido na literatura como uma excelente opção para predição de séries caóticas [60], apresentou um desempenho muito próximo ao desempenho obtido com um único filtro linear, de mesma ordem de predição. Embora tenhamos obtido um pequeno ganho na predição, com relação ao filtro linear, esta rede demanda um custo computacional muito mais elevado, e talvez não se justifique o esforço para o ganho obtido.

Já a *rede RBF dinâmica* mostrou-se ligeiramente superior aos outros dois preditores quanto às medidas de desempenho, possuindo a vantagem, se compararmos com o perceptron multicamadas FIR e com as redes neurais em geral, de realizar um aprendizado em *tempo real*, à medida que são apresentadas novas amostras da série. No entanto, podemos perceber que o seu desempenho depende de uma boa escolha dos parâmetros utilizados e de uma ordem de predição mais alta, diferentemente do que acontece com os outros dois preditores. A influência da escolha apropriada dos parâmetros pode ser percebida a partir dos valores de α (fator de escalonamento das variâncias), onde $\alpha = 400$ na série OctExt.TL (maior parâmetro H) e $\alpha = 30$, na série pOct.TL (menor parâmetro H). Em suma, numa situação real, onde o valor de H pode oscilar bastante, o desempenho da rede RBF pode ser comprometido se o valor de α permanecer constante. Mesmo assim, esta parece ser uma boa alternativa para a predição, dada a sua aparente robustez.

Quanto às ordens de predição obtidas para o filtro linear e o perceptron multicamadas FIR, percebemos que necessitamos de ordens maiores (no caso, $M = 20$) para valores de H maiores (como é o caso da série OctExt.TL). Para valores de H menores (como é o caso da série pOct.TL), necessitamos de ordens de predição menores ($M = 5$). Este fato faz bastante sentido se percebermos que quando o valor de H aproxima-se de $H = 0,5$, o processo tende a um ruído branco, cujo preditor ótimo é simplesmente a média amostral. Para valores maiores de H, a série se apresenta mais correlacionada, o que significa que necessitamos de mais

informação no passado para podermos aprimorar a predição.

Uma observação interessante pode ser feita com relação aos valores assumidos pelas medidas de desempenho em cada uma das séries. Podemos perceber que há uma "inversão" da ordem de grandeza dos valores quando alternamos de uma série para outra. Esta alternância pode ser explicada, talvez, pelo fato da série OctExt.TL possuir picos bastante elevados ("Efeito de Noah"), contrastantes com o comportamento geral da série, que se concentra em valores baixos, próximos a 0,2. Neste caso, o erro normalizado pela variância é bem mais baixo do que o erro normalizado pela amostra passada. No caso da série pOct.TL, a situação se inverte. A série parece apresentar uma maior variância, o que contribui para um aumento na ordem de grandeza do erro normalizado pela variância. Enquanto isso, a ausência de grandes picos tende a favorecer a atuação dos preditores, ocasionando erros normalizados pela amostra passada, um pouco mais baixos.

Com relação ao *algoritmo EM adaptativo* proposto neste trabalho, podemos perceber, a partir das simulações realizadas, que este algoritmo apresenta um desempenho bastante razoável se comparado com o algoritmo EM *padrão*. Inclusive, o que poderia ser apontado como o seu principal problema, trata-se também do principal problema encontrado no algoritmo EM *padrão* : a **convergência para mínimos locais**. A favor deste algoritmo, temos o seu caráter *adaptativo*, podendo ser implementado em situações que exijam estimações em tempo real de parâmetros de mistura de Gaussianas.

Uma possível abordagem para um melhor desempenho do algoritmo EM adaptativo, seria a realização de um "aprendizado" *a priori* sobre um conjunto de amostras, a partir do algoritmo EM *padrão*. Os valores obtidos nesta etapa de "treinamento", podem então ser utilizados como valores iniciais para o algoritmo EM *adaptativo*, o qual ficaria encarregado de acompanhar as pequenas variações ocorridas nas amostras ao longo do tempo.

Finalmente, podemos enumerar as seguintes contribuições deste trabalho :

- Estudo dos principais conceitos da rede ATM ;
- Revisão dos conceitos e modelos de processos auto-similares e o impacto desses processos na rede ATM ;
- Estudo e análise da auto-similaridade do tráfego Ethernet ;
- Descrição e implementação dos modelos e análise comparativa dos preditores lineares e não-lineares ;
- Proposta de uma modificação do algoritmo EM padrão para mistura de densidades Gaussianas, com a introdução de uma função de penalidade ;
- Proposta de um algoritmo EM adaptativo para estimação de parâmetros de mistura de densidades Gaussianas.

Apêndice A

Dimensão Fractal

Um conjunto fractal ideal $\mathcal{F}$ deve possuir as seguintes propriedades :

1. $\mathcal{F}$ possui uma estrutura fina, com detalhes em todos os níveis de resolução;
2. $\mathcal{F}$ é muito irregular para ser descrito na linguagem geométrica tradicional, tanto localmente quanto globalmente;
3. $\mathcal{F}$ possui alguma forma de auto-similaridade, quer seja determinística ou estatística;
4. Freqüentemente, $\mathcal{F}$ é definido de uma forma muito simples (quase sempre recursiva);
5. A dimensão fractal de $\mathcal{F}$ é maior do que a sua dimensão topológica.

Esta última propriedade deriva da definição formal utilizada por Mandelbrot [24, 5, 34] para caracterizar rigorosamente um conjunto fractal. Note que a dimensão topológica de um conjunto é sempre um número inteiro não-negativo. Ela é definida de forma recursiva, partindo do postulado de que a dimensão topológica de um ponto, ou de um conjunto de pontos desconectados, é igual a zero. Por outro lado, a dimensão fractal pode ser definida de modos diferentes, porém de certa forma equivalentes para todos os efeitos práticos. As definições mais conhecidas são a *dimensão de caixa* ("box dimension" ou Entropia de Kolmogorov) e a *dimensão de Hausdorff*. A dimensão fractal é, em geral, um número fracionário.

Na maioria dos casos práticos, utiliza-se como definição de dimensão fractal uma forma mais simples e mais intuitiva que as duas mencionadas acima. Para entendê-la, tomemos o caso simples de um conjunto $\mathcal{F}_1$ correspondente a uma linha (curva ou reta) de tamanho L. Para medirmos o tamanho da linha, tomemos uma régua rígida de tamanho r e estimemos o menor número $N(r)$ de réguas que cobrem o tamanho L. Portanto, para $r \to 0$ temos que $L \approx N(r) \cdot r$. No limite,

$$L = \lim_{r \to 0} N(r) \cdot r.$$

Note porém que, para qualquer $\delta > 0$,

$$\lim_{r \to 0} N(r) \cdot r^{1-\delta} = \infty$$

e

$$\lim_{r \to 0} N(r) \cdot r^{1+\delta} = 0.$$

Portanto, $\delta = 0$ corresponde ao expoente $D_H = 1$de r^{D_H},para o qual o limite de $N(r) \cdot r^{D_H}$ converge para um número maior que zero e menor que infinito. Neste caso, $D_H = 1$

corresponde à dimensão topológica e à dimensão fractal do conjunto $\mathcal{F}_1$. Como estas dimensões são idênticas, $\mathcal{F}_1$ não é um fractal.

Analogamente, se tomarmos $\mathcal{F}_2$como uma superfície de uma folha de área S,teremos

$$S = \lim_{r \to 0} N(r) \cdot r^2$$

e, para $\delta > 0$,

$$\lim_{r \to 0} N(r) \cdot r^{2-\delta} = \infty$$

e

$$\lim_{r \to 0} N(r) \cdot r^{2+\delta} = 0.$$

Neste caso, $N(r)$ é o menor número de quadrados de lado r que cobrem a área S.Aqui as dimensões topológicas e fractais são ambas iguais a dois. O mesmo raciocínio pode ser aplicado a um conjunto $\mathcal{F}_3$ correspondente a um sólido com dimensão topológica (e fractal) igual a três.

A discussão acima sugere a seguinte definição de dimensão fractal :

Definição A.1 *A dimensão fractal D_H de um conjunto $\mathcal{F}$ é o número real não-negativo para o qual o limite de*

$$N(r) \cdot r^{D_H}$$

quando $r \to 0$ converge para um valor finito e maior que zero.

Vários autores utilizam esta definição como sendo a da dimensão de Hausdorff[24, ?]. Deve-se enfatizar, todavia, que a dimensão de Hausdorff é definida de forma mais rigorosa e, em muitos casos, fornece valores diferentes daqueles obtidos com a dimensão fractal acima.

Note que, se $N(r) \cdot r^{D_H} \cong c \ (0 < c < \infty)$,então

$$\log c \cong \log N(r) + D_H \log r$$

e

$$D_H = \frac{\log N(r)}{-\log(r)} + \frac{\log c}{\log r}.$$

Tomando o limite $(r \to 0)$, o segundo termo tende a zero e, então,

$$D_H = \lim_{r \to 0} \frac{\log N(r)}{\log(1/r)}.$$

Esta expressão é muito útil para aplicações práticas, pois D_H pode ser estimado como o gradiente do gráfico log-log traçado em uma região conveniente de r.Mandelbrot define então como fractal todo conjunto que possui uma dimensão de Hausdorff D_H estritamente maior que sua dimensão topológica.

Bibliografia

[1] W. E. Leland, M. Taqqu, W. Millinger, and D. V. Wilson, "On the self-similar nature of ethernet traffic (extended version)," *IEEE/ACM Transactions on Networking*, vol. 2, no. 1, pp. 1–14, 1994.

[2] G. Gripenberg and I. Norros, "On the prediction of fractional brownian motion," *J. Appl. Prob.*, vol. 33, pp. 400–410, 1996.

[3] J. Roberts, U. Mocci, and J. Virtamo, eds., *Broadband Network Teletraffic : Performance Evaluation and Design of Broadband Multiservice Networks ; Final Report of Action COST 242.* Berlin : Springer -Verlag, 1996.

[4] W. Willinger, M. S. Taqqu, and A. Erramilli, *Stochastic Networks : Theory and Applications*, ch. A Bibliographical Guide to Self-Similar Traffic and Performance Modeling for Modern High-Speed Networks, pp. 339–366. Oxford : Claredon Press, 1996.

[5] B. B. Mandelbrot, *The Fractal Geometry of Nature.* San Francisco, CA : Freeman, 1982.

[6] B. B. Mandelbrot and J. V. Ness, "Fractional brownian motions, fractional noises and applications," *SIAM Review*, vol. 10, pp. 422–437, Outubro 1968.

[7] V. Paxson and S. Floyd, "Wide-area traffic : The failure of poisson modeling," *IEEE/ACM Transactions on Networking*, vol. 3, pp. 226–244, 1995.

[8] J. Beran, R. Sherman, M. S. Taqqu, and W. Willinger, "Long-range dependence in variable-bit-rate video traffic," *IEEE Transactions on Communication*, vol. 43, no. 2-4, pp. 1566–1579, 1995.

[9] A. G. D. de Oliveira, M. M. de Carvalho, and D. S. Arantes, "Modelos fractais para tráfego de taxa de bit variável em redes atm," Tech. Rep. Pub. FEEC 027/96, Faculdade de Engenharia Elétrica e de Computação da Universidade Estadual de Campinas, 1996.

[10] B. Tsybakov and N. D. Georganas, "On self-similar traffic in atm queues : Definitions, overflow probability bound, and cell delay variation," *IEEE/ACM Transactions on Networking*, vol. 5, no. 3, pp. 397–409, 1997.

[11] I. Norros, "On the use of fractional brownian motion in the theory of connectionless networks," *IEEE Journal on Selected Areas In Communications*, vol. 13, no. 6, pp. 953–996, 1995.

[12] B. B. Mandelbrot, "Long-run linearity, locally gaussian processes, h-spectra and infinite variances," *International Economic Review*, vol. 10, pp. 82–113, 1969.

[13] S. Haykin, A. H. Sayed, J. R. Zeidler, P. Yee, and P. Wei, "Tracking of linear time-variant systems," in *MILCOM 95 - San Diego CA*, 1995.

[14] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum-likelihood from incomplete data via the EM algorithm," *J. Royal Statistic Soc. Ser. B (methodological)*, vol. 39, pp. 1–38, 1977.

[15] M. de Prycker, *Asynchronous Transfer Mode.* Prentice Hall International, 3a. ed., 1995.

[16] R. O. Onvural, *Asynchronous Transfer Mode Networks : Performance Issues.* Artech House, second ed., 1995.

[17] D. Boettle, "Switching of 140 mbit/s signals in broadband communication systems," *Electrical Communication*, vol. 58, no. 4, 1984.

[18] J. P. Leduc, *Digital Moving Pictures - Coding and Transmission on ATM Networks.* Elsevier, 1995.

[19] H. Scheffè, *The Analysis of Variance.* New York, NY : Wiley, 1959.

[20] J. Beran, "A goodness of fit tests for time series with long-range dependence," *J. Roy. Statist. Soc. Ser. B*, vol. 54, pp. 749–760, 1992.

[21] G. Wornell, *Signal Processing with Fractals : A Wavelet Based Approach.* Prentice Hall, 1996.

[22] H. E. Hurst, "Long-term storage capacity of reservoirs," *Transactions of the American Society of Civil Engineers*, no. 116, pp. 770–779, 1951.

[23] I. Adelman, "Long cycles : Fact or artefact ?," *American Economic Review*, vol. LV, pp. 444–463, Junho 1965.

[24] H.-O. Peitgen and D. Saupe, eds., *The Science of Fractal Images.* New York : Springer -Verlag, 1988.

[25] A. N. Kolmogorov, "Wienersche spiralen und einige andere interessante kurven im hilbertschen raum," *C. R. (Doklady) Acad. Sci. URSS (N.S.)*, no. 26, pp. 115–118, 1940.

[26] B. B. Mandelbrot, "Self-similar error-clusters in communication systems and the concept of conditional stationarity," *IEEE Transactions on Communication Technology*, vol. COM-13, pp. 71–90, Março 1965.

[27] B. B. Mandelbrot, "Some noises with 1/f spectrum, a bridge between direct current and white noise," *IEEE Transactions on Information Theory*, vol. IT-13, pp. 289–298, Abril 1967.

[28] R. J. Barton and H. V. Poor, "Signal detection in fractional gaussian noise," *IEEE Transactions on Information Theory*, vol. 34, pp. 943–959, Setembro 1988.

[29] H. Weyl, "Bemerkungen zum begriff der differential-quotenten gebrochener ordnung," *Vierteljschr. Naturforsch. Ges. Zurich*, vol. 62, pp. 296–302, 1967.

[30] P. Flandrin, "On the spectrum of fractional brownian motions," *IEEE Transactions on Information Theory*, vol. 35, pp. 197–199, Janeiro 1989.

[31] W. Martin and P. Flandrin, "Wigner-ville spectral analysis of nonstationary processes," *IEEE Transactions on Acoustic, Speech and Signal Processing Proceedings*, vol. ASSP-33, no. 6, pp. 1461–1470, 1985.

[32] I. Daubechies, "The wavelet transform : Time-frequency localization and signal analysis." Preprint. AT&T Bell Labs., 1987.

[33] A. Grossmann and J. Morlet, "Decomposition of hardy functions into square integrable wavelets of constant shape," *SIAM J. Math. Anal.*, vol. 15, no. 4, pp. 723–736, 1984.

[34] K. Falconer, *Fractal Geometry : Mathematical Foundations and Applications.* New York, NY : John Wiley and Sons, 1990.

[35] I. M. Gelfand and N. Y. Vilenkin, *Generalized Functions*, vol. 4. New York : Academic Press, 1964.

[36] J. R. M. Hosking, "Fractional differencing," *Biometrika*, vol. 68, no. 1, pp. 165–176, 1981.

[37] G. E. P. Box and G. M. Jenkins, *Time Series Analysis : Forecasting and Control*. Oakland, CA : Holden-Day, 1976.

[38] M. S. Taqqu and J. B. Levy, "Using renewal processes to generate long-range dependence and high variability," *Dependence in Probability and Statistics*, vol. 11, pp. 73–89, 1986.

[39] W. E. Leland and D. V. Wilson, "High time-resolution measurement and analysis of lan traffic : Implications for lan interconnection," in *Proc. IEEE INFOCOM'91*, (Bal Harbour, FL), 1991.

[40] H. Heffes and D. M. Lucantoni, "A markov modulated characterization of packetized voice and data traffic and related statistical multiplexer performance," *IEEE Journal of Selected Areas on Communication*, vol. SAC-4, pp. 856–868, 1986.

[41] R. Jain and S. A. Routhier, "Packet trains : Measurements and a new model for computer network traffics," *IEEE Journal of Selected Areas on Communication*, vol. SAC-4, pp. 986–995, 1986.

[42] D. Anick, D. Mitra, and M. M. Sondhi, "Stochastic theory of a data-handling system with multiple sources," *Bell System Technical Journal*, vol. 61, pp. 1871–1894, 1982.

[43] J. Beran, *Statistics for Long-Memory Processes*. New York : Chapman and Hall, 1994.

[44] M. E. Crovella and A. Bestavros, "Self-similarity in world wide web traffic : Evidence and possible causes," in *Proceedings of the 1996 ACM SIGMETRICS. International Conference on Measurement and Modeling of Computer Systems*, 1996.

[45] V. Paxson and S. Floyd, "Wide-area traffic : The failure of poisson modeling," in *Proceedings of the 1996 ACM/SIGCOMM'94*, pp. 257–268, 1994.

[46] M. S. Taqqu, V. Teverovsky, and W. Willinger, "Estimators for long-range dependence : An empirical study," *Fractals*, vol. 3, no. 4, pp. 785–788, 1995.

[47] B. B. Mandelbrot and J. R. Wallis, "Computer experiments with fractional gaussian noises, parts 1,2,3," *Water Resources Research*, vol. 10, pp. 228–267, 1969.

[48] B. B. Mandelbrot and M. S. Taqqu, "Robust r/s analysis of long-run serial correlation," in *Proceedings of the 42nd Session of the International Statistical Institute*, vol. 48 of *Book 2*, (Manila), pp. 69–104, Bulletin of the International Statistical Institute, 1979.

[49] T. Higuchi, "Approach to an irregular time series on the basis of the fractal theory," *Physica D*, vol. 31, pp. 277–283, 1988.

[50] L. Kosten, "Stochastic theory of a multi-entry buffer," *Delft Progress Report*, vol. 1, pp. 10–18, 1974.

[51] L. C. G. Rogers, "Arbitrage with fractional brownian motion," *Mathematical Finance*, 1995.

[52] W. Willinger, M. S. Taqqu, R. Sherman, and D. V. Wilson, "Self-similarity through high-variability : Statistical analysis of ethernet lan traffic at the source level," *IEEE/ACM Transactions on Networking*, vol. 5, no. 1, pp. 71–86, 1997.

[53] R. A. Redner and H. F. Walker, "Mixture densities, maximum likelihood and the EM algorithm," *SIAM Review*, vol. 26, pp. 195–239, Abril 1984.

[54] B. Widrow and S. D. Stearns, *Adaptive Signal Processing.* Prentice-Hall, Englewood Cliffs, N.J., 1985.

[55] S. Haykyn, *Adaptive Filter Theory.* Prentice Hall International, 2a. ed., 1991.

[56] A. Lapedes and R. Farber, "Nonlinear signal processing using neural networks : Prediction and modeling," Tech. Rep. LA-UR87-2662, Los Alamos National Laboratory, Los Alamos, New Mexico, 1987.

[57] S. Haykyn, *Neural Networks : A Comprehensive Foundation.* Prentice Hall International, 1994.

[58] B. J. A. Krose and P. P. van der Samgt, "An introduction to neural networks," University of Amsterdam, The Netherlands, 1993.

[59] D. O. Hebb, *The Organization of Behaviour.* Wiley, New York, 1949.

[60] N. A. Gershenfeld and A. S. Weigend, eds., *Times Series Prediction : Forecasting The Future and Understanding The Past.* Addison-Wesley, 1993.

[61] D. S. Broomhead and D. Lowe, "Multivariable functional interpolation and adaptive networks," *Complex Systems*, vol. 2, pp. 321–355, 1988.

[62] F. Girosi and T. Poggio, "Networks and the best approximation property," *Biological Cybernetics*, vol. 63, pp. 169–176, 1990.

[63] C. M. Bishop, *Neural Networks for Pattern Recognition.* Clarendon Press - Oxford, 1996.

[64] I. J. Good and R. A. Gaskins, "Nonparametric roughness penalties for probability densities," *Biometrika*, vol. 58, no. 2, pp. 255–277, 1971.

[65] D. S. Arantes, *Estimation of Sparse Contingency Tables and of Normal Mixtures.* PhD thesis, Cornell University, 1977.

[66] H. G. C. Traven, "A neural network approach to statistical pattern classification by semiparametric estimation of probability density functions," *IEEE Transactions on Neural Networks*, vol. 2, no. 3, pp. 366–377, 1991.

[67] M. M. de Carvalho and D. S. Arantes, "Predição de tráfego auto-similar em redes atm," in *XV Simpósio Brasileiro de Telecomunicações*, (Recife), Setembro 1997.

[68] M. M. de Carvalho and D. S. Arantes, "Self-similar traffic prediction with fir neural networks," in *IV Simpósio Brasileiro de Redes Neurais*, (Goiânia), pp. 106–108, Dezembro 1997.